

GC38-0282-1
File No. S370-30

**Advanced Communications
Function for VTAM
(ACF/VTAM)**

Program Product

Concepts and Planning

**Program Numbers 5746-RC3 (DOS/VS)
5735-RC2 (OS/VS)**

IBM

GC38-0282-1
File No. S370-30

Advanced Communications Function for VTAM (ACF/VTAM)

Program Product

Concepts and Planning

**Program Numbers 5746-RC3 (DOS/VS)
5735-RC2 (OS/VS)**



Second Edition (August 1977)

This edition is a major revision of the previous edition, GC38-0282-0, which is now obsolete. Refer to the Summary of Amendments for a description of changes made in this edition.

This edition applies to the initial DOS/VS version of ACF/VTAM (Program Number 5746-RC3) and to the initial OS/VS version of ACF/VTAM for OS/VS1, OS/VS2 SVS, and OS/VS2 MVS (Program Number 5735-RC2). Information about the optional Multisystem Networking Facility of ACF/VTAM is included. The information in this publication should be used for planning purposes until ACF/VTAM becomes available for your operating system.

The program product described in this manual, and all licensed materials available for it, are provided by IBM under terms of the Agreement for IBM Licensed Programs. Your branch office can advise you on the ordering procedures.

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. Comments become the property of IBM.

Summary of Amendments (August 1977) to GC38-0282-0 by Revision GC38-0282-1 ACF/VTAM for DOS/VS

New Program Features

DOS/VS: ACF/VTAM is now available as a program product for use with DOS/VS Release 34.

New 3270 Devices: ACF/VTAM supports the following new 3270 devices:

- 3274 Control Unit, Models 1A,1B, and 1C
- 3276 Control Unit Display Station,
Models 1, 2, 3, 4, 11, 12, 13, and 14
- 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 Printer, Model 2
- 3289 Printer, Models 1 and 2

These devices are included in Appendix A, "Supported Terminals."

Changed Documentation

Scope of Changes: Changes, corrections, and improvements have been made throughout the text, and the manual should be reviewed in its entirety.

References to Non-SNA Terminals: The text has been changed so that local 3270, BSC, and start-stop devices are now referred to collectively as *non-SNA terminals*.

References to ACF/VTAM: The Advanced Communications Function for the Virtual Telecommunications Access Method is now referred to throughout the book by its abbreviation, *ACF/VTAM*.

References to Device-Type Logical Unit: At some points in the text, a distinction is made between two types of logical units: an application program (which is one kind of a logical unit), and a device-type logical unit (which is any logical unit other than an application program).

Rewritten Topics: The following topics have been completely rewritten:

- "How ACF/VTAM Operates" in Chapter 2
- "Defining Cross-Domain Path Tables" in Chapter 3
- "Defining Field-Formatted Logons" in Chapter 3
- "Defining Logon Modes" in Chapter 3
- "Holding the Physical Unit Connection" in Chapter 3
- "Defining ACF/VTAM Buffering" in Chapter 3
- "Halting ACF/VTAM" in Chapter 4
- "Monitoring ACF/VTAM Status with the Display Command" in Chapter 4
- "Activating and Deactivating Local Non-SNA Terminals" in Chapter 4
- "Activating and Deactivating a Remote NCP" in Chapter 4
- "Activating and Deactivating Cross-Domain Resource Managers" in Chapter 4
- "Activating and Deactivating Path Tables" in Chapter 4
- "Establishing Sets of Session Parameters" in Chapter 5
- "Storage Management" in Chapter 6
- "Switched Network Backup" in Chapter 6
- "Resource Takeover" in Chapter 6
- "Dial-In Operation" in Chapter 7
- "Resources That Can Be Shared" in Chapter 7

Change in Appendix A Format: Appendix A has been changed from a list format to a table format, and new 3270 devices have been added to the table.

Glossary: The Glossary has been revised.

Preface

This publication provides an overview of Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM). It is directed primarily to data processing managers and system programmers who may install or maintain a data communication system that uses ACF/VTAM.

The installer of a system that uses ACF/VTAM should use this book to get a general understanding of ACF/VTAM concepts and of the factors that must be considered in planning and installing an ACF/VTAM system. Then, to perform specific steps, such as estimating main storage requirements and writing ACF/VTAM definition statements, the installer should use the appropriate *ACF/VTAM System Programmer's Guide*, the appropriate *ACF/VTAM Installation Guide*, the *NCP Generation* publication (see "Related Publications" below), relevant operating system publications (see the Bibliography that precedes the Index), and publications that may be required to install the terminals and terminal systems that will communicate with ACF/VTAM application programs. The programmer who writes ACF/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 ACF/VTAM is provided in *ACF/VTAM General Information*, GC38-0254.

All statements in this manual that pertain to multiple-host configurations and cross-domain operations are valid only for ACF/VTAM systems that include the Multisystem Networking Facility.

ACF/VTAM can operate either with the ACF version of the network control program (ACF/NCP/VS) or with the latest current level of NCP/VS. However, some of the new functions available in ACF/VTAM (such as cross-domain communication) cannot be utilized unless ACF/NCP/VS is installed. The *ACF/VTAM Program Product Design Objectives*, GC38-0253, lists the functions that require ACF/NCP/VS.

This publication uses the term *NCP* to refer to either version of the network control program. Where a unique reference to the ACF/NCP/VS version is required for clarity, the term *ACF/NCP* is used. Both versions of the network control program are described in *Introduction to the 3704 and 3705 Communications Controllers*, GA27-3051.

How This Book Is Organized

Chapters 1 and 2 introduce ACF/VTAM and describe its major concepts and facilities.

Chapters 3, 4, and 5 describe the primary interfaces to ACF/VTAM. Chapter 3 describes how to define an ACF/VTAM data communication system. Chapter 4 describes how to control an ACF/VTAM system. Chapter 5 describes in general how to write ACF/VTAM application programs.

Chapter 6 describes ACF/VTAM's reliability, availability, and serviceability features.

Chapter 7 describes hardware and software requirements for ACF/VTAM and discusses planning considerations for functions such as data communication security.

Chapter 8 describes support for local non-SNA 3270, BSC, and start-stop terminals (which are referred to in this manual as *non-SNA terminals*).

Related Publications (Other Than ACF/VTAM)

The reader should be familiar with the basic concepts of data communications. These can be found in the publication *Introduction to Data Communications Systems*, SR20-4461.

Depending on the terminal product with which ACF/VTAM is used, the reader may need 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, and to *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112.

References are made in this publication to the *NCP Generation* publication. *NCP Generation* is an abbreviation for one of the following books depending on which version of the network control program is being used:

For ACF/NCP/VS users:

IBM 3705 Advanced Communications Function for Network Control Program/VS Generation and Utilities Reference Manual, SC30-3116

For NCP/VS users:

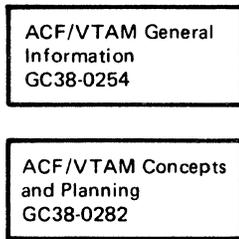
IBM 3704 and 3705 Communications Controllers: Network Control Program/VS Generation and Utilities: Guide and Reference Manual, GC30-3008

Publications other than ACF/VTAM publications that are referred to in this publication or are related to this publication are listed in the Bibliography at the end of this book.

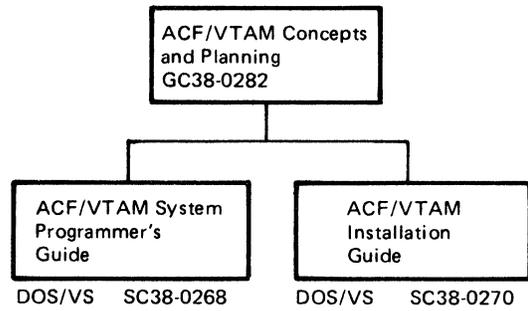
Related ACF/VTAM Publications

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

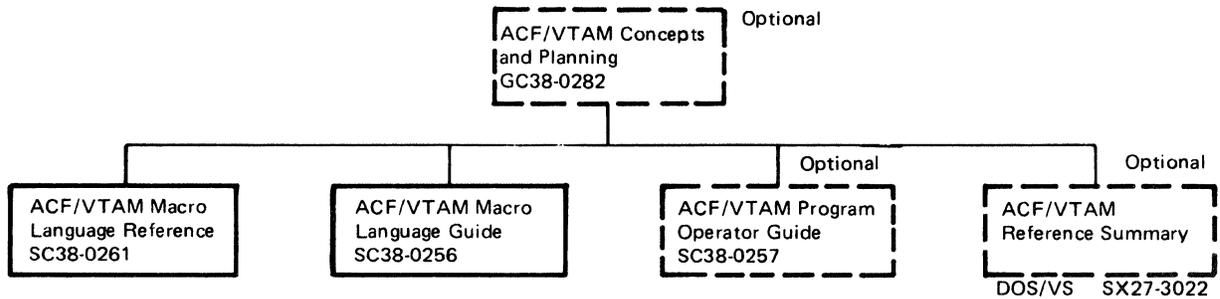
General Information



Planning and Installing ACF/VTAM



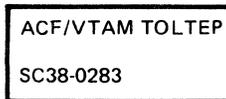
Writing ACF/VTAM Application Programs



Operating ACF/VTAM



Using the Teleprocessing Online Test Executive Program (TOLTEP)



Analyzing/Maintaining ACF/VTAM

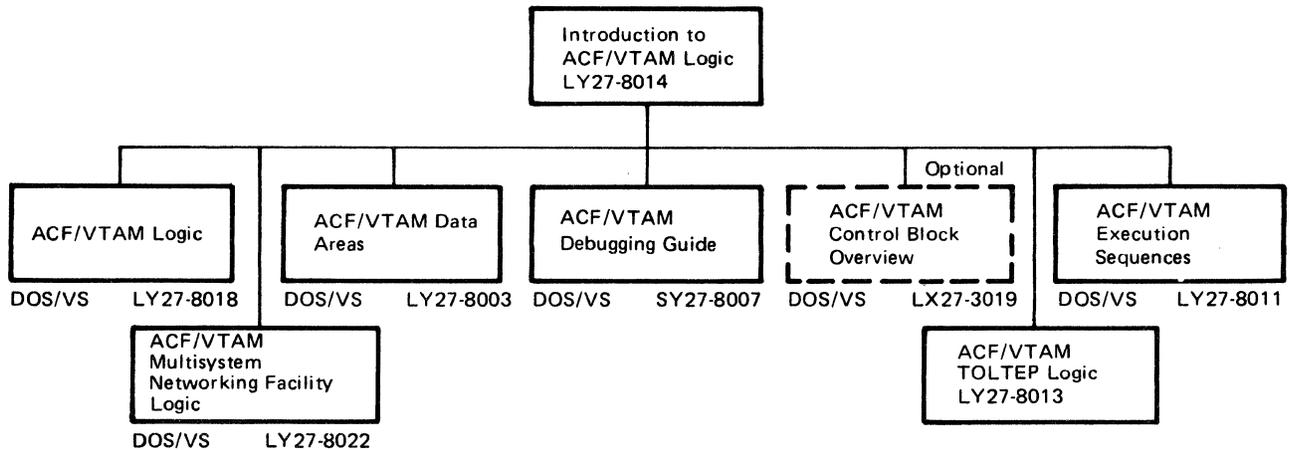


Figure P-1. ACF/VTAM Publications

Contents

Preface i

Chapter 1. Introduction to ACF/VTAM 1

ACF/VTAM and the Multisystem Networking Facility	1
Systems Network Architecture (SNA) Concepts in ACF/VTAM	4
The SNA Concept of Network Addressable Units	4
The SNA Concept of Primary and Secondary Logical Units	6
The SNA Concept of Sessions	6
The SSCP-PU Session	7
The SSCP-LU Session	8
Initiate Command or Logon	8
Logon Exit Routine	8
Opening the LU-LU Session (OPNDST Macro Instruction)	8
The Bind Command	9
Completing the LU-LU Session	9
The SNA Concept of Domains as Implemented by ACF/VTAM	9
Major Facilities and Characteristics of ACF/VTAM	9
Allocation of Resources	11
Data Flow	11
Sharing Resources	11
Distributed Function	13
Elements of an ACF/VTAM System	13
ACF/VTAM Application Programs	15
Communications Controllers	15
Terminals	16
SNA Terminals	16
Non-SNA Terminals	17

Chapter 2. An ACF/VTAM System 19

Views of an ACF/VTAM System	19
A Single-Domain System	19
A Multiple-Domain System	28
Installing an ACF/VTAM System	29
Creating the ACF/VTAM System	29
Controlling the ACF/VTAM System	29
Designing ACF/VTAM Application Programs	30
Establishing Procedures for Using the System	30
How ACF/VTAM Operates	30

Chapter 3. Creating an ACF/VTAM System 33

Defining ACF/VTAM and Logical Devices to the Operating System	33
Generating a Network Control Program (NCP)	36
Defining the Network to ACF/VTAM	36
The Major and Minor Node Structure	37
Naming Major and Minor Nodes	39
Defining Application Program Major Nodes	40
Defining Local Non-SNA Major Nodes	41
Defining Local SNA Major Nodes	41
Defining NCP Major Nodes	41
Defining Switched SNA Major Nodes	43
Defining Other Domains to ACF/VTAM	43
Defining Cross-Domain Resource Manager Major Nodes	43
Defining Cross-Domain Resource Major Nodes	43
Defining Cross-Domain Path Tables	44
Defining Connection Procedures	44
Defining Terminal-Initiated Connection	46
Defining Secondary Application Program Connection	46
Types of Logons	46
Defining Field-Formatted Logons	46
Defining Character-Coded Logons	47

Using the IBM-Supplied Character-Coded (USS) Definition Table	47
Defining a Character-Coded (USS) Definition Table	49
Defining and Using an Interpret Table	52
Defining ACF/VTAM-Initiated Connection (Automatic Logon)	52
Defining ACF/VTAM Application-Program-Initiated Connection	52
Acquiring Connection (OPNDST with OPTCD=ACQUIRE)	53
Simulating a Logon (SIMLOGON)	53
Passing Connection to Another Program (CLSDST with OPTCD=PASS)	53
Defining Network-Operator-Initiated Connection	53
Defining Logon Modes (Session Parameters)	54
Tables That Contain Session Parameters	54
Processing of Session Parameters	55
Defining Disconnection Procedures	56
Disconnection Requested by the Terminal	56
Conditional or Unconditional Disconnection	57
Holding the Physical Unit's Physical Connection	57
Using the Request Shutdown Protocol	57
Disconnection Requested by a Primary Application Program	58
Disconnection Requested by a Secondary Application Program	59
Disconnection Requested by the Network Operator	59
Disconnection Requested by ACF/VTAM	59
Effect of Disconnection on Automatic Logon	60
Authorization, Accounting, and Logon-Interpret Exit Routines	60
Authorization Exit Routine	60
Accounting Exit Routine	61
Logon-Interpret Routines	62
Defining ACF/VTAM Start Options	62
Defining ACF/VTAM Buffering	64
Chapter 4. Controlling an ACF/VTAM System 67	
Levels of Control	67
ACF/VTAM Network Operator Commands	67
Starting ACF/VTAM	68
Starting ACF/VTAM in DOS/VS	69
Starting ACF/VTAM in OS/VS	69
Halting ACF/VTAM	69
Orderly Shutdown	69
Quick Shutdown	69
Cancel Shutdown (OS/VS Only)	70
Monitoring ACF/VTAM Status with the DISPLAY Command	70
Activating and Deactivating Nodes	74
Starting and Stopping Application Programs	76
Activating and Deactivating Local Non-SNA Terminals	76
Activating and Deactivating Local SNA Major Nodes	77
Activating and Deactivating a Nonshared Local NCP	77
Activating and Deactivating a Shared Local NCP	78
Activating and Deactivating a Remote NCP	78
Activating and Deactivating Remote Attachments	78
Activating and Deactivating Switched SNA Major Nodes	79
Activating and Deactivating Cross-Domain Resource Managers	80
Activating and Deactivating Cross-Domain Resources	80
Activating Path Tables	80
Special Considerations for Activation	81

Acquiring and Releasing Resources	82	Chapter 6. Reliability, Availability, and Serviceability	125
Initiating Requests for Connection	82	ACF/VTAM's RAS Strategy	125
Starting and Stopping ACF/VTAM Facilities	83	ACF/VTAM's RAS Facilities	126
Suppressing Network Operator Messages	83	Serviceability Aids	126
Changing Line-Scheduling Specifications	83	Error Recording	127
Changing NCP Configurations	84	Traces	127
Network Operator Control from an Authorized Application Program	84	Dumps	129
Considerations for Network Operator Control	86	Teleprocessing Online Test Executive Program (TOLTEP)	130
Chapter 5. ACF/VTAM Application Programs	87	Reliability and Availability Support	131
The ACF/VTAM Language	87	Error Detection and Feedback	132
ACF/VTAM Macro Instructions	87	3704/3705 Initial Test	132
Connection Macro Instructions	88	Storage Management	132
Communication Macro Instructions	88	Error Recovery	133
Network Control Macro Instructions	89	NCP Slowdown	134
Control-Block Macro Instructions	89	Switched Network Backup	134
Support Macro Instructions	90	Resource Takeover	134
Relating ACF/VTAM Control Blocks and Executable Macro Instructions	91	Configuration Restart	140
Opening and Closing an Application Program	91	Transferring a Remote Communications Controller as a Remote Controller	141
Primary Application Program Connection	92	Lost Subarea Notification	142
Primary Application Program Disconnection	93	Chapter 7. ACF/VTAM Planning Considerations and Requirements	143
Secondary Application Program Connection	93	Machine Requirements	143
Secondary Application Program Disconnection	93	CPU Support	143
Communicating in Basic Mode or Record Mode	93	Local 3270s	143
Handling Control Blocks, I/O Areas, and Work Areas	94	Local 3790s	143
Organization of the Application Program	94	Requirements for Communications Controllers	143
Overlapping ACF/VTAM Requests with Other Processing	95	Remote Terminals	145
EXLST Exit Routines	96	Storage Requirements	145
Error Notification	98	Operating System Requirements	145
Other Considerations for Program Organization	100	Upward Compatibility	145
Application Program Concepts and Facilities	100	Requirements for DOS/VS	146
Opening an Application Program	100	Requirements for OS/VS	147
Primary Application Program Connection	100	Network Control Program Requirements	148
Acceptance	100	NCP Functions Required by ACF/VTAM	148
Acquisition	101	Defining the NCP to ACF/VTAM	148
Queuing Connection Requests	102	Pacing	149
Establishing Sets of Session Parameters	104	Support for SNA Switched Networks	149
Secondary Application Program Connection	105	Defining a Switched SNA Major Node	150
Primary Application Program Disconnection	105	Generating Switched Line Groups in the NCP	150
Secondary Application Program Disconnection	105	ID Verification for Switched SNA Major Nodes	151
Record-Mode Communication	106	Dial-In Operation	151
Messages and Responses	106	Dial-Out Operation	152
Normal and Expedited Message Flow	107	Physical Disconnection of Physical Units and Logical Units in a Switched Major Node	154
Session Control and Data Flow Control	107	Partitioned Emulation Programming (PEP)	154
Requesting a Response	107	ACF/VTAM Data Sets	155
Positive and Negative Responses	110	Data Sets for ACF/VTAM under OS/VS	155
Definite Response Types 1 and 2	110	Data Requirements for ACF/VTAM under DOS/VS	155
Special Handling of a Response	110	Sharing Resources	157
Sequencing and Chaining	110	Resources That Can Be Shared	157
Scheduled and Responded Output	111	Sharing and Managing Resources	158
Receiving Input	112	Managing Resources through ACF/VTAM Definition	158
Quiesce, Change-Direction, and Bracket Protocols	115	Managing Resources through NCP Generation	159
Request and Response Modes	120	Managing Resources through Application Programs	160
Additional Protocols	120	ACF/VTAM Security	160
Basic-Mode and Record-Mode Communication	121	SNA Terminal Identification Verification on Switched Lines	161
Identifying Application Programs, Logical Units, and Non-SNA Terminals	121	Host ID Verification for Physical Units	161
Specific-Mode and Any-Mode	121	BSC and TWX Identification Verification on Switched Lines	161
Continue-Any and Continue-Specific Modes	123	Controlling Connections	163
Handling Excess Input Data	123		

Authorization Exit Routine	163	Basic-Mode Macro Instructions	184
Acquiring and Passing Connections	163	The LDO Control Block	184
Controlling Logons	164	Communicating with Non-SNA Terminals	185
Symbolic Names	166	Solicitation	185
Controlling Access to ACF/VTAM	166	Special I/O Operations	187
Controlling the Use of ACF/VTAM Facilities	167	Special Processing Options	187
Protecting Confidential Data	167	Communicating with BSC 3270 or Local Non-SNA 3270	
Other Telecommunication Access Methods	167	Terminals in Record Mode	188
Other Telecommunication Access Methods in DOS/VS	168	Appendix A. Supported Terminals	189
Other Telecommunication Access Methods in OS/VS	169	SNA Terminals	189
Other Telecommunication Access Methods in a Multiple-Domain Network	170	Non-SNA Terminals	189
Chapter 8. Support for Non-SNA Terminals	171	Start-Stop Terminals	189
Using BTAM	171	Binary Synchronous Communications (BSC) Terminals	190
Using ACF/VTAM	171	Local Non-SNA 3270 Terminals	190
Topics Not Applicable to Non-SNA Terminals	175	Terminal Support List	190
Creating an ACF/VTAM System That Includes Non-SNA Terminals	175	Appendix B. Remote Station Versus Remote Controller	193
The Network Solicitor	176	Appendix C. Summary of Message Control Information	197
How the Network Solicitor Works	176	Appendix D. Considerations in Moving from VTAM Level 2 to ACF/VTAM	199
Modifying the Network Solicitor	177	Use of the System Management Program to Install ACF/VTAM	199
Replacing the Network Solicitor	178	Configuration Considerations	199
Specifying Interpret Tables	178	Definition Considerations	200
Defining Terminal-Initiated Logons for Non-SNA Terminals	180	Operator Considerations	201
Defining a Switched Network for Start-Stop and BSC Terminals	181	Application Program Considerations	202
Call-In Terminals	181	Glossary	205
Call-Out Terminals	182	Bibliography	213
Call-In/Call-Out Terminals	182	Index	215
Operating an ACF/VTAM System with Non-SNA Terminals	183		
Writing an ACF/VTAM Application Program	183		
Basic-Mode Concepts	184		

Figures

- Figure P-1. ACF/VTAM Publications i
- Figure 1-1. A Single-Domain ACF/VTAM Network 2
- Figure 1-2. A Multiple-Domain ACF/VTAM Network 3
- Figure 1-3. The SSCP, Physical Units, and Logical Units in a Network 5
- Figure 1-4. Establishing an LU-LU Session 7
- Figure 1-5. A Data Communication Network with Two Domains 10
- Figure 1-6. Data Flow through a Multiple-Domain ACF/VTAM Network 12
- Figure 1-7. Data Flow through an ACF/VTAM System 13
- Figure 1-8. The Major Elements in a Single-Domain ACF/VTAM Network 14
- Figure 2-1. Four Views of an ACF/VTAM Data Communication System 20
- Figure 2-2. Physical Configuration of a Multiple-Domain ACF/VTAM Network 22
- Figure 2-3. The Network As Viewed by the Host Operating Systems 23
- Figure 2-4. The Domains As Viewed by ACF/VTAM 24
- Figure 2-5. The Network As Viewed by ACF/VTAM 1 25
- Figure 2-6. The Network As Viewed by ACF/VTAM 2 26
- Figure 2-7. The Data Communication System As Viewed by Application Programs 27
- Figure 3-1. The Steps in Creating an ACF/VTAM Domain 34
- Figure 3-2. Major and Minor Nodes in a Single-Domain ACF/VTAM System 38
- Figure 3-3. Additional Major and Minor Nodes in a Multiple-Domain ACF/VTAM System 39
- Figure 3-4. Generating NCP Support in an ACF/VTAM System 43
- Figure 3-5. ACF/VTAM Path Tables for Cross-Domain Paths 45
- Figure 3-6. A Field-Formatted Logon 48
- Figure 3-7. A Character-Coded Logon 48
- Figure 3-8. Defining a USS Definition Table 50
- Figure 3-9. Example of ACF/VTAM's Handling of a Character-Coded (USS) Logon 51
- Figure 3-10. Defining a Logon Mode Table 55
- Figure 3-11. How ACF/VTAM Uses Input and Output Buffers 65
- Figure 4-1. Network Operator Control of the ACF/VTAM System 85
- Figure 5-1. Processing Pattern for a Synchronous Request 95
- Figure 5-2. Processing Pattern When an ECB is Used with an Asynchronous Request 96
- Figure 5-3. Processing Pattern When an RPL Exit Routine is Used with an Asynchronous Request 97
- Figure 5-4. Processing Pattern for Reporting Errors during an Asynchronous Operation 98
- Figure 5-5. Queued and Nonqueued Connection Requests 103
- Figure 5-6. An Example of Start Data Traffic and Clear Commands 108
- Figure 5-7. An Example of a Logical Unit Requesting (A) a Definite Response and (B) Only an Exception Response 109
- Figure 5-8. An Example of Message Chaining 112
- Figure 5-9. Scheduled Output 113
- Figure 5-10. Responded Output 114
- Figure 5-11. Types of Information Exchanged between an Application Program and Terminals 115
- Figure 5-12. An Example of Quiesce Protocol 116
- Figure 5-13. An Example of Change-Direction Protocol 117
- Figure 5-14. An Example of Bracket Protocol 118
- Figure 5-14. Commands and Indicators Used to Direct the Flow of Data 119
- Figure 5-16. Using a Combination of Any-Mode and Specific-Mode 122
- Figure 5-17. Using the Continue-Any and Continue-Specific Modes to Handle Concurrent Inquiries 124
- Figure 6-1. Resource Takeover (Local Controller Taken Over as Local Controller) 137
- Figure 6-2. Resource Takeover (Local Controller Taken Over as Remote Controller) 139
- Figure 7-1. Dial-In Operation 151
- Figure 7-2. A Path for a Dial-Out Operation 152
- Figure 7-3. An Alternate Path for a Dial-Out Operation with an Automatic Calling Unit 153
- Figure 7-4. Data Sets Used by ACF/VTAM under OS/VS 156
- Figure 7-5. Files Used by ACF/VTAM under DOS/VS 157
- Figure 7-6. ACF/VTAM's Use of Libraries under DOS/VS 158
- Figure 7-7. Other Telecommunication Access Methods in DOS/VS 168
- Figure 7-8. Other Telecommunication Access Methods in OS/VS 169
- Figure 7-9. Other Telecommunication Access Methods in a Multiple-Domain Network 170
- Figure 8-1. Major Similarities and Differences between ACF/VTAM and BTAM Application Programs 172
- Figure 8-2. Using BTAM and ACF/VTAM to Communicate with Non-SNA Terminals 174
- Figure 8-3. Processing a Terminal-Initiated Logon with the Network Solicitor 177
- Figure 8-4. Filing Interpret Tables 179
- Figure 8-5. Providing Control Information for Processing Logons from Non-SNA Terminals 181
- Figure 8-6. Implicit and Explicit Solicitation Using Basic Mode 186
- Figure A-1. Devices Supported by ACF/VTAM 191
- Figure B-1. A Remote Communications Controller 193
- Figure B-2. Communications Controllers Attached as Part of a Remote Station 194

Chapter 1. Introduction to ACF/VTAM

The IBM program product, Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM), controls the transmission of data between elements in a data communication network. With ACF/VTAM, application programs and terminals can communicate without concern for intermediate connections such as communications controllers and communication lines. To support communication within the network, ACF/VTAM does these things:

- Controls the allocation of resources

- Establishes, controls, and terminates connections between resources

- Transfers data between points in the network

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

- Allows communication controllers and terminals to perform some network functions

- Permits the operation of the network to be monitored by the network operator

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

- Attempts to detect and correct problems in the operation of the network

The portion of the data communication system that ACF/VTAM controls is called its *domain*. A domain is that portion of a total network that is controlled by a particular ACF/VTAM or by another telecommunications access method (such as ACF/TCAM) that supports networking. Figure 1-1 shows a possible single-domain ACF/VTAM configuration. In a data communication system with more than one host computer or in a system in which one NCP is shared between separate hosts, there may be more than one ACF/VTAM domain. Figure 1-2 shows a possible multiple-domain ACF/VTAM network.

ACF/VTAM and the Multisystem Networking Facility

ACF/VTAM can be used alone as a telecommunication access method, or it can be used with the Multisystem Networking Facility (an additional program product feature of ACF/VTAM).

When used *without* the Multisystem Networking Facility, ACF/VTAM does the things listed above for communications within the single-domain network under its control. When used *with* the Multisystem Networking Facility, ACF/VTAM in concert with ACF/VTAMs and ACF/TCAMs in other domains provides the services listed above and, in addition, the following cross-domain services:

- Permits application programs in one host computer to communicate with application programs and terminals in other host computers

- Permits terminals attached to one host computer to communicate with application programs in other host computers

- Establishes, controls, and terminates access to application programs and SNA terminals in other parts of the multiple-host network

- Allows a communications controller of one host computer to be transferred to an adjacent host computer

- Permits the terminals of one 3705-II Communications Controller to be shared by up to four host computers

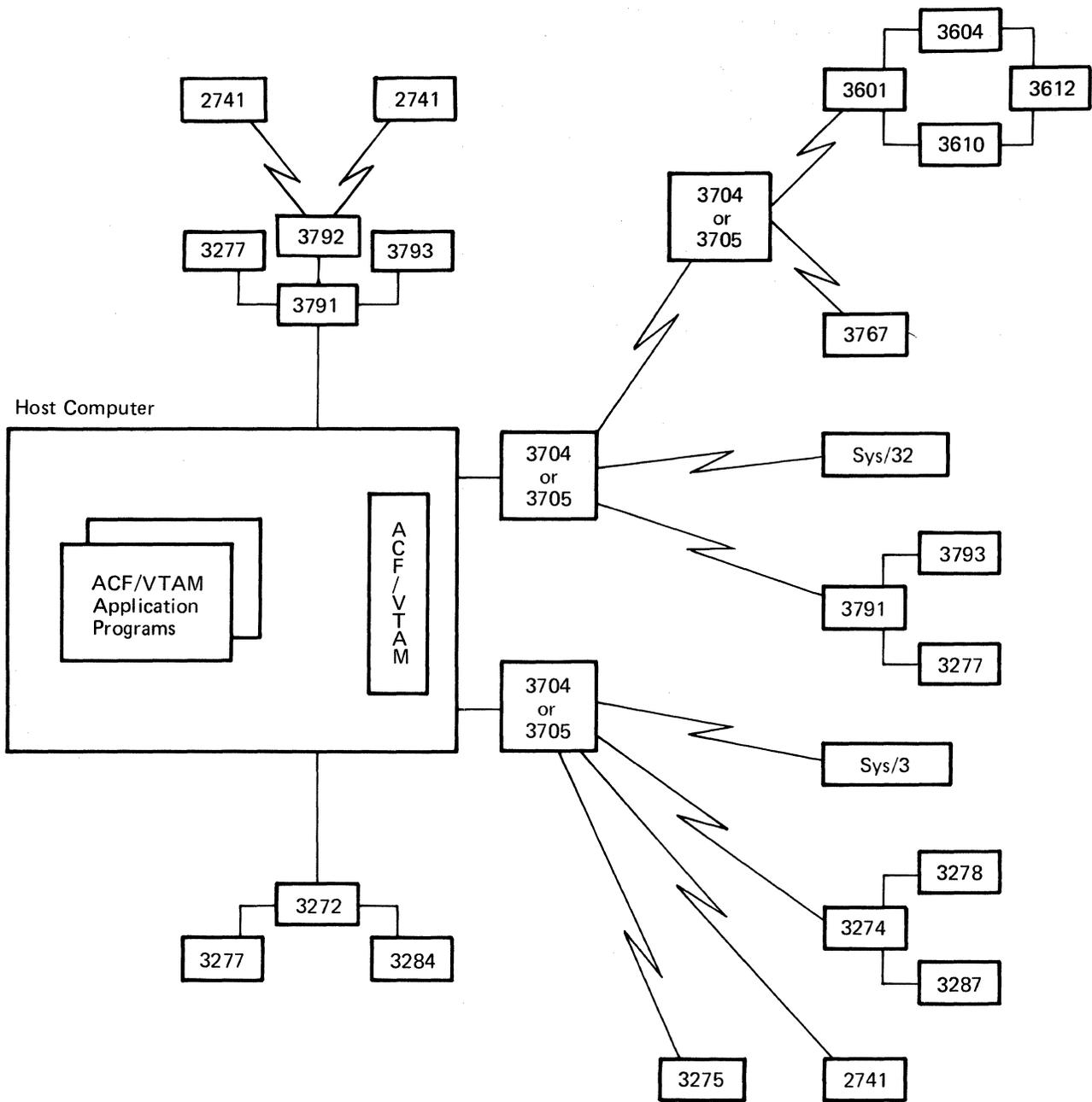


Figure 1-1. A Single-Domain ACF/VTAM Network

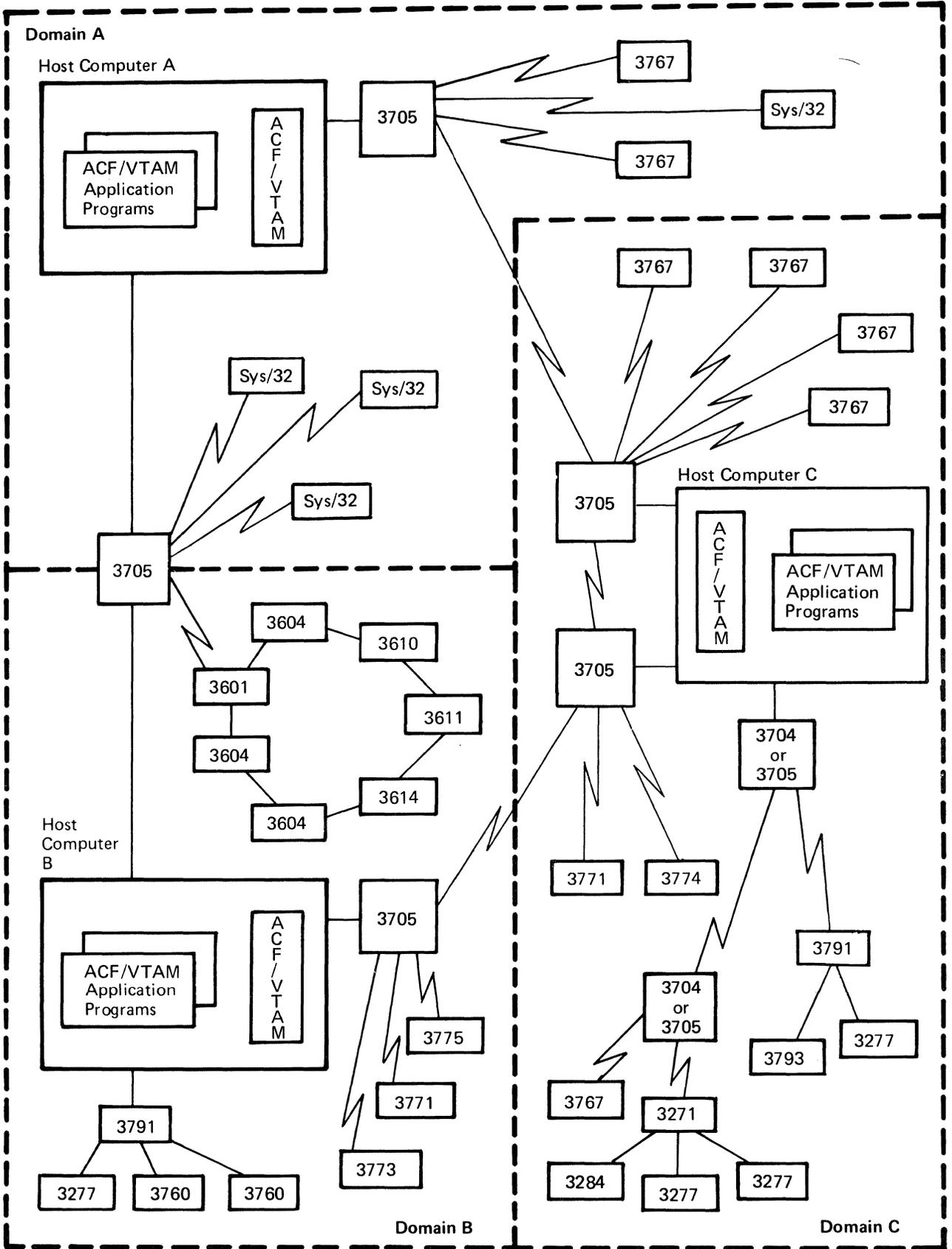


Figure 1-2. A Multiple-Domain ACF/VTAM Network

For further information on functions of ACF/VTAM alone and with the Multisystem Networking Facility, see *Program Product Specifications for Advanced Communications Function for VTAM (ACF/VTAM)* for the operating system being used at the installation.

Systems Network Architecture (SNA) Concepts in ACF/VTAM

ACF/VTAM follows SNA concepts and uses SNA protocols to connect and communicate with elements in a data communication network.

Several SNA concepts provide helpful background information for a person who wants to understand and use an ACF/VTAM system. Those concepts are:

- The concept of network addressable units
- The concept of primary and secondary logical units
- The concept of sessions
- The concept of domains

Those concepts are described in the following sections.

The SNA Concept of Network Addressable Units

Each element in a network to which a data or control message can be sent is assigned a network address by ACF/VTAM. Each element with such an address is known as a *network addressable unit (NAU)*. The network address uniquely identifies the element, regardless of whether the element is a device (such as a terminal or terminal control unit), a program (such as an application program in a cluster controller or terminal), or a portion of ACF/VTAM. For ACF/VTAM and other elements in the data communication network, the network address contains the information necessary to route a message to its destination.

Three types of network addressable units are defined by SNA and recognized by ACF/VTAM: (1) system services control point (SSCP), (2) physical units (PUs), and (3) logical units (LUs). Figure 1-3 shows the locations of these types of network addressable units in a simplified network.

The *system services control point* is a unit of coding in ACF/VTAM that manages the network and has primary control over communications. The SSCP performs functions such as bringing up the network and shutting it down, establishing and disestablishing connections (sessions) between units, and reacting to network problems (such as failure of a link unit). To perform these functions, the SSCP must be able to communicate with physical units and logical units in the network under its control.

A *physical unit* is not literally a physical device in the network. Rather, a physical unit is a portion of a device (usually programming or circuitry, or both) that performs control functions for the device in which it is located and, in some cases, for other devices that are attached to the PU-containing device. For the devices under its control, the physical unit takes action during activation and deactivation, during error recovery and resynchronization, during testing, and during gathering of statistics on operation of the device. Each device in the network is associated with a physical unit.

The physical unit may exist either within the device or within an attached controlling device. The physical unit exists within a host computer, a communications controller, and a cluster control unit. For a terminal, however, the physical unit may be within the terminal or it may be within the terminal control unit, cluster control unit, or communications controller to which the terminal is attached.

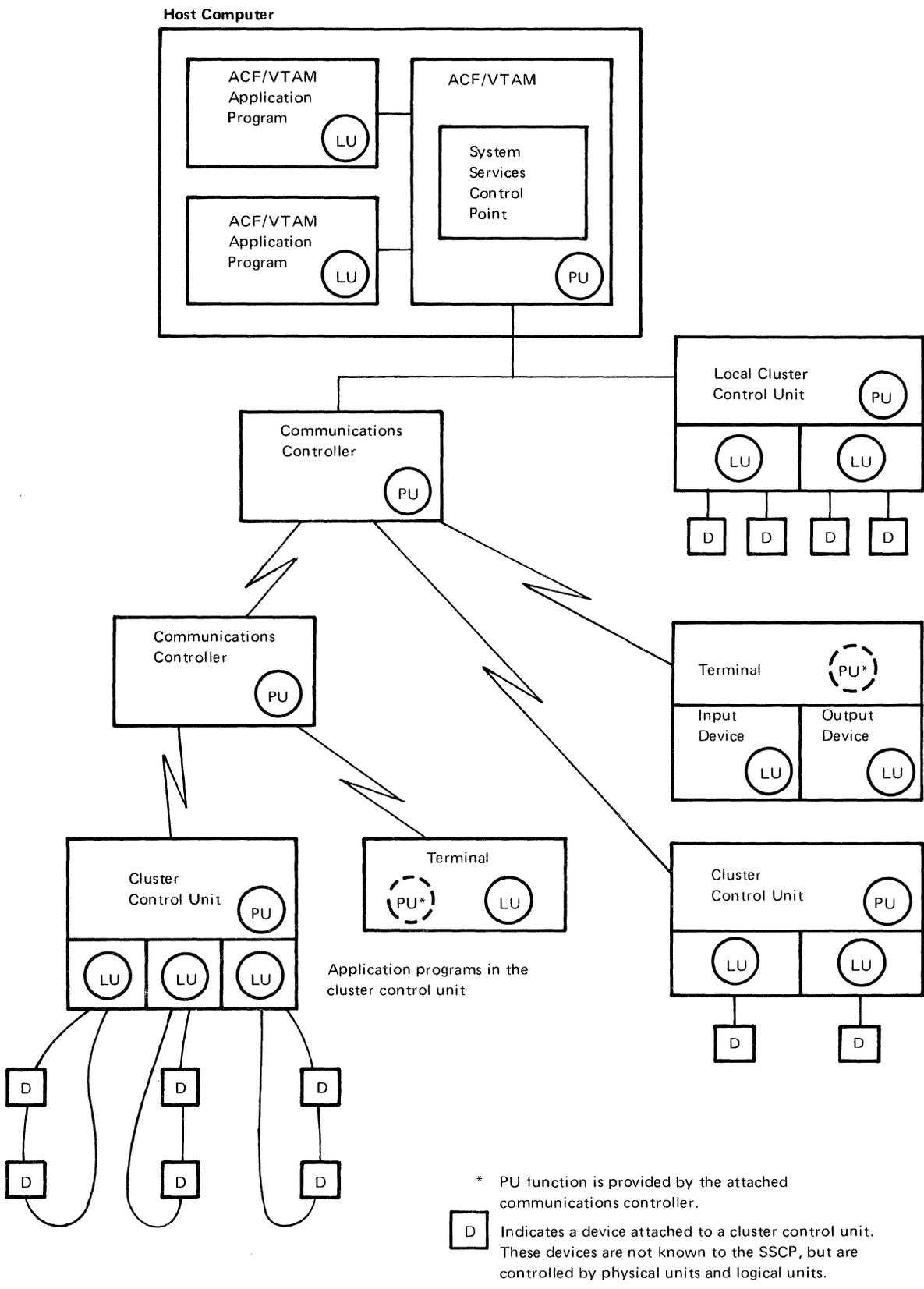


Figure 1-3. The SSCP, Physical Units, and Logical Units in a Network

A *logical unit* is a device or program by which an *end user* (a terminal operator or an input/output mechanism) gains access to the data communication network. To the network, a logical unit is the source of a message coming into the network. But the logical unit may or may not be the original source. The contents of the message or the information on which the message is based may have originated at a device controlled by the logical unit. (For example, in a 3601 cluster control unit, the logical unit is a program that handles input and output for one or several finance terminals attached to the controller. Input actually originates at one of the terminals, but it is the logical unit [the program] in the 3601 that uses the input to create a message and begin transmission of the message.) Similarly, the network sees a logical unit as the destination of a message, but the logical unit may actually pass the message on to a device for recording, printing, or displaying to a terminal operator. (For example, a message received by a logical unit [a program] in a 3601 may be passed on to a finance terminal to be displayed on the screen of that terminal.) In some cases, however, the logical unit is an intrinsic part of the device at which the message is displayed (for example, a 3767 terminal contains the logical unit and is the input/output device).

An ACF/VTAM application program is also a logical unit. ACF/VTAM sees it as an originator of and destination for messages. But there can be other programs in the host computer that interface with an ACF/VTAM application program and to which the contents of messages can be passed or from which the contents of messages can be received. Thus, although the ACF/VTAM application program is the logical unit, the messages it handles may be used by another program. In this case, the other program is the end user.

The SNA Concept of Primary and Secondary Logical Units

In communication between two logical units, one logical unit acts as the primary end of the session (by using primary protocols), and the other end acts as the secondary end of the session (using secondary protocols).

In communication between an ACF/VTAM application program and a device-type logical unit (a device-type logical unit is a logical unit other than an ACF/VTAM application program), the ACF/VTAM application program acts as the primary end of the session and the device-type logical unit acts as the secondary end.

In communication between two application programs, one program acts as the primary end, and the other program acts as the secondary end. The manner in which the programs become connected determines which is the primary and which is the secondary end of the session. The program that functions as the primary end of the session is called the *primary application program*. The program that functions as the secondary end of the session is called the *secondary application program*.

When an ACF/VTAM application program has concurrent sessions with device-type logical units and with other application programs, it functions as the primary end of the sessions with the device-type logical units and can function as the primary end or secondary end of each session with another application program. Thus, under these circumstances, an ACF/VTAM application program can be both primary and secondary ends of sessions at the same time.

The SNA Concept of Sessions

Before two units in a network can communicate with each other, the units must be tied together in what is known as a *session*. In an SNA network, several different types of sessions are established, including SSCP-SSCP sessions, SSCP-PU sessions, SSCP-LU sessions, and LU-LU sessions.

When a network includes more than one host computer and therefore more than one ACF/VTAM (or ACF/VTAM in one or more hosts and ACF/TCAM in one or more other hosts), a session called an *SSCP-SSCP session* must be established between the SSCP in one ACF/VTAM and each other SSCP with which the first SSCP will communicate.

Within the machine configuration controlled by each SSCP, different kinds of sessions are established in stages. The SSCP must first establish an *SSCP-PU session* with each physical unit that is active in the configuration. Then, for each active logical unit associated with a physical unit, the SSCP must establish an *SSCP-LU session*. And finally, when a pair of logical units indicate that they want to communicate with each other, the SSCP must establish an *LU-LU session* between them. The paragraphs that follow describe the steps in establishing these sessions, with the circled numbers beside the headings serving as keys to the circled numbers in Figure 1-4.

The SSCP-PU Session

①

To get ready for an SSCP-LU session, the SSCP must first establish a session with the physical unit that controls the logical unit. This type of session is called an *SSCP-PU session*. This session is used to exchange messages and commands that pertain to startup and shutdown of the machine configuration or the individual physical unit and to the recovery of operations after a device or link failure. After the SSCP-PU session has been established, the SSCP can attempt to establish sessions with logical units associated with that physical unit. The SSCP-PU session is established on a nonswitched line as soon as the physical unit is activated. On a switched line, the session is established following a dial-in or dial-out operation. For local SNA devices, the session is established when physical connection is established. In ACF/VTAM, the SSCP-PU session is established by the SSCP; and ACF/VTAM application program does not itself take any direct action to establish that session.

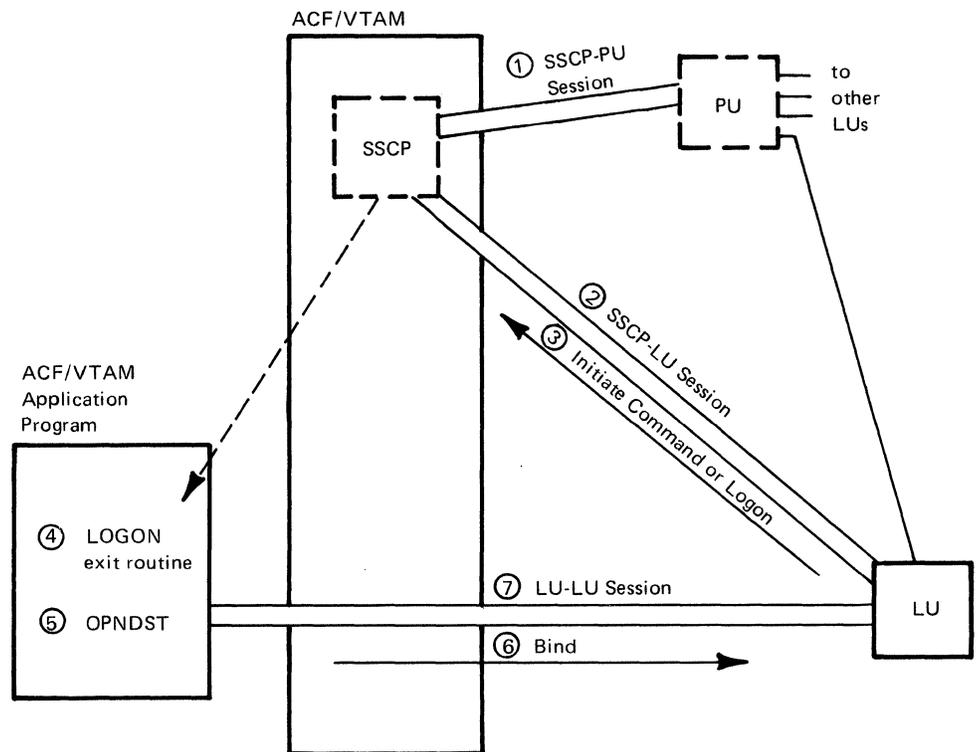


Figure 1-4. Establishing an LU-LU Session

The SSCP-LU Session ②

Once a session has been established between the SSCP and a physical unit, the SSCP can issue commands to establish a session between itself and any active logical unit associated with the physical unit. This type of session is called an *SSCP-LU session*. The establishment of this session allows SNA commands to flow back and forth between the logical unit and the SSCP. These commands pertain mainly to connection and disconnection. In ACF/VTAM, the SSCP-LU session is established by the SSCP; an ACF/VTAM application program does not itself take any direct action to establish the session.

Initiate Command or Logon ③

After the SSCP has established a session with a logical unit, the logical unit can attempt to initiate a session with an ACF/VTAM application program. Action by the logical unit to start such a session is usually initiated when a terminal operator communicates with the logical unit and indicates that he or she wants to work with an application program in the host computer. The logical unit either uses the logon information entered by the terminal operator to create an Initiate command to be sent to the SSCP, or the logical unit passes the logon information from the terminal operator on to the SSCP in the form in which it was received from the operator.

In addition, after the application-program-to-ACF/VTAM interface has been established, a secondary ACF/VTAM application program can request that a session be established with a primary ACF/VTAM application program. In this case, the secondary program issues a macro instruction (the REQSESS macro instruction), which causes an Initiate command (containing logon information) to be sent to the SSCP.

In either case, when the logon information reaches the SSCP, the SSCP notifies the primary application program that the logon has been received and should be processed. The logon information includes the logon mode name and, optionally, a user logon message. The user logon message is particular data that the terminal operator or logical unit wants to be passed to the application program. When the primary application program is notified that the logon has been received, session parameters and the user logon message are made available for inspection by the program. Session parameters are a set of codes that indicate the communication rules that the logging-on unit wants to use for the session that is about to be established. These parameters are defined by the logon mode name associated with the secondary end of the session. The parameters specify such things as whether chained or unchained messages will be sent, what kinds of responses will be requested, which logical unit will start and end brackets, and so on.

LOGON Exit Routine ④

In ACF/VTAM, the SSCP notifies the application program that the logon has been received by scheduling execution of the program's LOGON exit routine. The LOGON exit routine then either accepts or rejects the logon. During processing of the logon, the application program determines whether the session parameters suggested by the logical unit are the right ones for the session or whether a different set of session parameters should be used.

Opening the LU-LU Session (OPNDST Macro Instruction) ⑤

If the application program decides to go into session with the logical unit, it issues an OPNDST macro instruction. As a result of the OPNDST macro instruction, ACF/VTAM builds a Bind command and sends it to the logical unit. If the application program decides to reject the request for a session, it issues a CLSDST macro instruction, and the session is not established.

The Bind Command

⑥

The Bind command is the key item in establishing the LU-LU session. Besides indicating the application program's willingness to go into session, the Bind command contains the session parameters that the program decided should be used for the session (the parameters may be the same or different from those suggested in the logical unit's logon information). If the logical unit agrees with the session parameters included in the Bind command, the logical unit sends a positive response to the Bind. If the logical unit does not agree with the session parameters, it sends a negative response and the LU-LU session is not completed.

Completing the LU-LU Session

⑦

When ACF/VTAM receives a positive response to the Bind command, ACF/VTAM completes the LU-LU session and the logical units are ready to communicate. In some cases, the exchange of messages and responses cannot begin until a Start Data Traffic command is sent by the primary end of the session to the logical unit. The need for the Start Data Traffic command is determined by the transmission services profile in the session parameters.

The SNA Concept of Domains as Implemented by ACF/VTAM

A data communication network is divided into domains when the network contains more than one host computer and each host computer contains ACF/VTAM (or another telecommunication access method that supports cross-domain communication). A domain is that portion of a total network that is controlled by a particular ACF/VTAM (or another telecommunication access method). Figure 1-5 shows a network with two domains. The elements in domain A are controlled by the ACF/VTAM in host computer 1; the elements in domain B are controlled by the ACF/VTAM in host computer 2.

The domains are joined either (1) by a cross-domain link between two local communications controllers (as shown in Figure 1-5) or (2) by a communications controller that is channel-attached to more than one host computer or that services more than one telecommunication access method in the same host computer.

When a network contains more than one domain, an ACF/VTAM application program in one domain can communicate with logical units in its own domain and in other domains. The application program can communicate across domain boundaries with most SNA logical units, with ACF/VTAM or ACF/TCAM application programs, and with specially defined BSC 3270 terminals. The application program cannot communicate with local, start-stop, or BSC terminals (except specially defined BSC 3270 terminals) in other domains.

In cross-domain communications, neither end of the session need be aware that the other end of the session is in another domain. By using a symbolic name to refer to the resource in another domain, each end of the session provides enough information for the resource to be identified and located. All addressing and routing of messages between the domains is handled automatically by ACF/VTAM (and other networking access methods) and by the network control programs (NCPs) in the local communications controllers.

Major Facilities and Characteristics of ACF/VTAM

Information on the major facilities and characteristics of ACF/VTAM is provided in the following paragraphs.

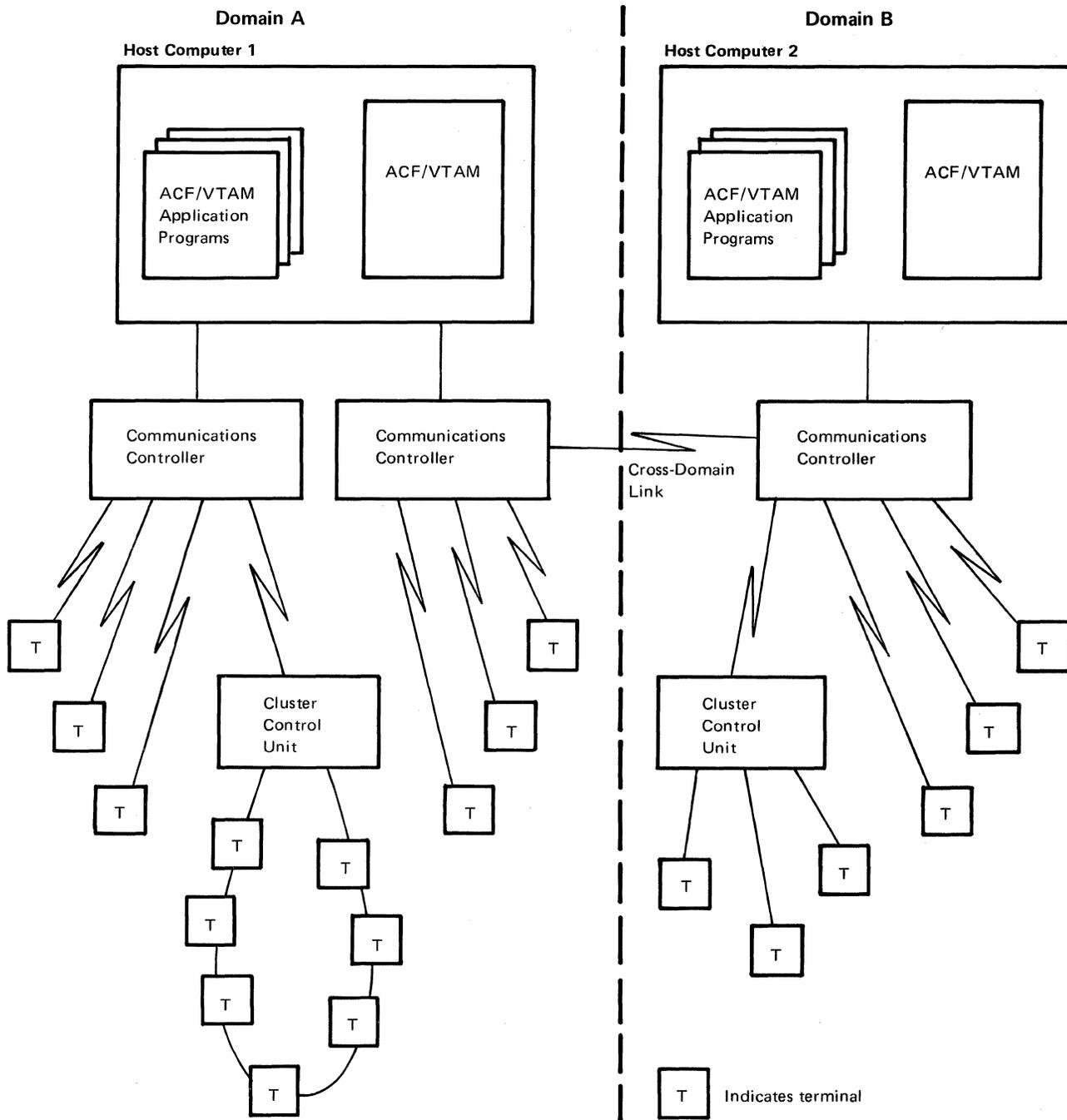


Figure 1-5. A Data Communication Network with Two Domains

Allocation of Resources

ACF/VTAM controls the allocation of all resources within its domain; that is, ACF/VTAM decides who can use which resources (such as lines and terminals) and when they can use them. The only aspect of resource allocation of direct concern to an application program is the issuing of requests to have other resources connected to it. ACF/VTAM allocates intermediate elements (lines and devices between the ends of a communication session) only for the time needed to satisfy a specific transmission request.

Data Flow

ACF/VTAM manages the flow of data between itself and application programs, and between the host computer (using the I/O facilities of the operating system) and 3704/3705 Communications Controllers (hereinafter referred to as *communications controllers*). Although ACF/VTAM does not actually transfer data throughout the network, it provides the information needed for intermediate elements (such as communications controllers) to route the data.

In a multiple-domain network, data flows through only those elements that are necessary to complete a path between two end points. For example, in the network in Figure 1-6, data moving from an application program in host computer A to a 3767 Communication Terminal attached to host computer C does not have to pass through host computer C. The 3705 Communications Controller routes the data to the appropriate terminal. Note that data for the 3771 or 3774 terminal attached to host computer C passes through both 3705 Communications Controllers rather than through the host computer. A host computer that lies in the path between two ends of a conversation can also route data to other parts of the network. In Figure 1-6, host computer B is part of the path for communication between an application program in domain A and the 3775 Communication Terminal in domain B.

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

In a data communication system with SNA terminal systems or System/370s used as remote stations, some communication occurs beyond ACF/VTAM's control. Figure 1-7 shows the data flow for an ACF/VTAM system that has an SNA terminal system. Note that ACF/VTAM manages the data transfer only between the application program and the logical unit in the control unit. The program in the control unit is responsible for the transfer of the data between itself and the work station (which is the final destination of the message).

ACF/VTAM application programs are responsible for transferring data to and from auxiliary storage devices (tape drives, disk storage units, and other external storage devices attached to the host computer). The application program can use an access method such as IBM's Virtual Storage Access Method (VSAM) to perform these input/output operations.

Sharing Resources

ACF/VTAM allows the users of a network to share the resources of that network. A user can define the network so that any application program can communicate with terminals or application programs in its domain or in other domains without regard for the physical location of that resource.

An application program can communicate simultaneously with several terminals (including other application programs), but a device-type terminal can communicate with only one application program at a time.

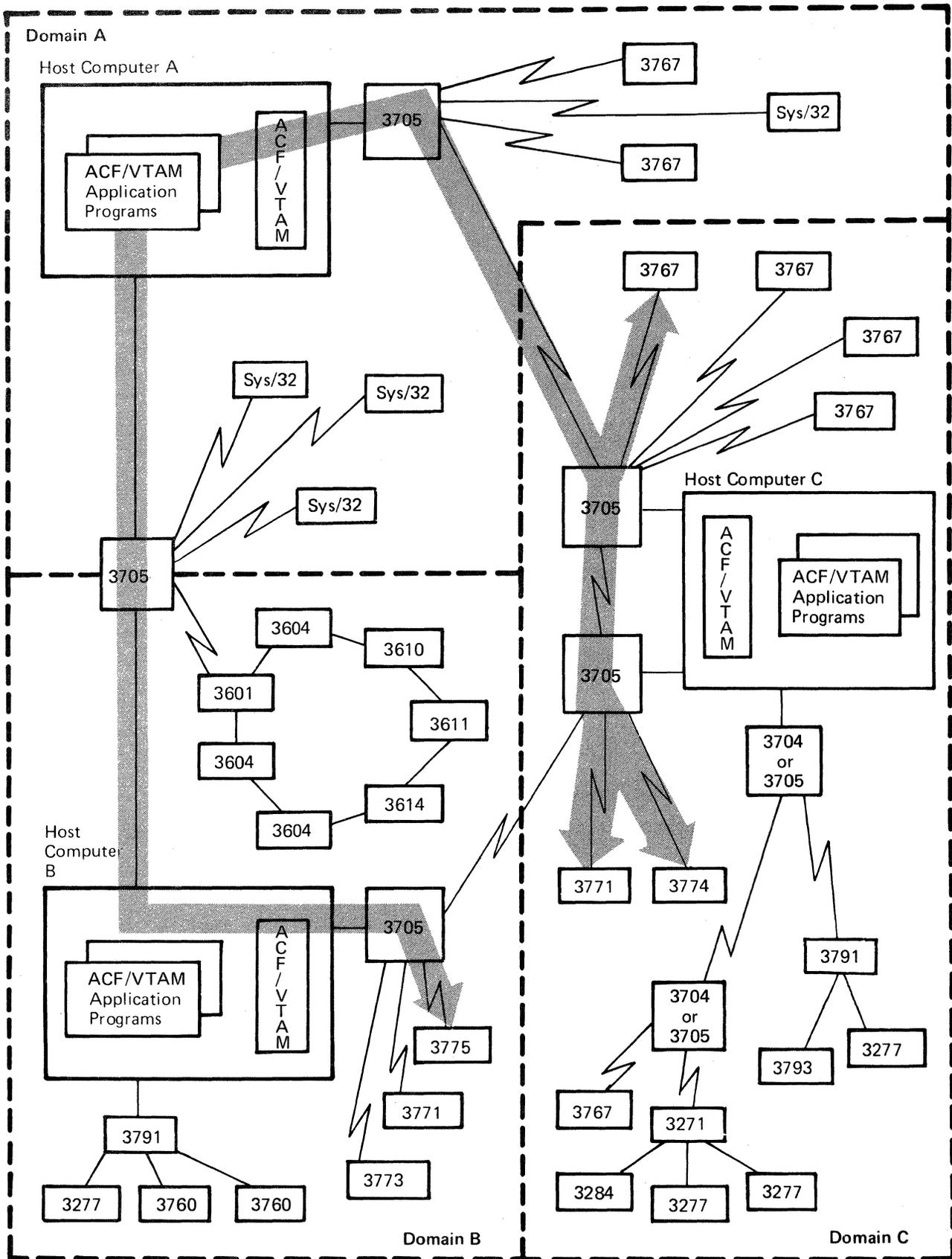


Figure 1-6. Data Flow through a Multiple-Domain ACF/VTAM Network

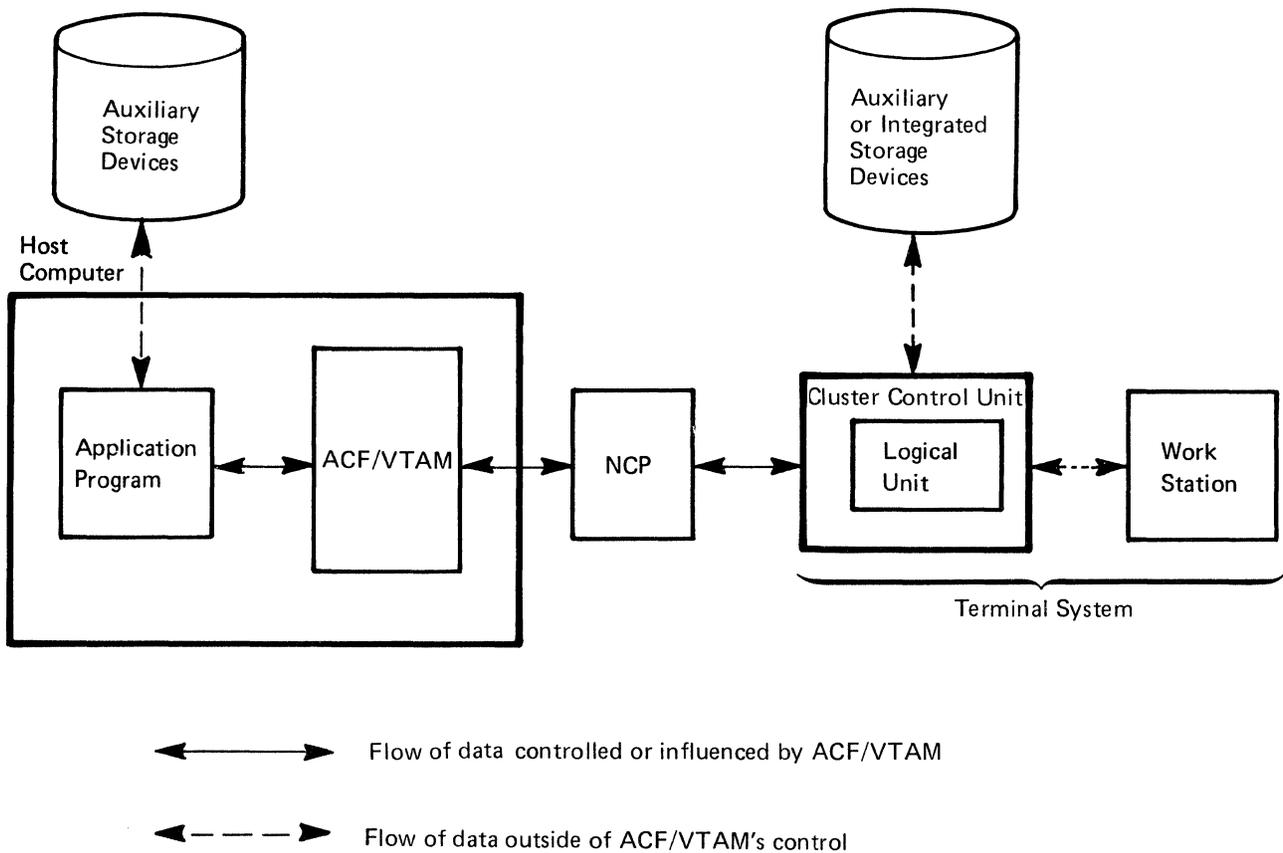


Figure 1-7. Data Flow through an ACF/VTAM System

Resources that make up the paths between application programs and terminals are also shared. ACF/VTAM uses path elements (such as communications controllers and lines) on behalf of application programs and terminals 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 an ACF/VTAM system, network control is distributed throughout the network rather than concentrated in the host computer. Communications controllers handle line scheduling and polling. Programmable terminals handle the network beyond ACF/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.

In addition, in a multiple-domain environment, functions can be distributed among resources in different domains. For example, a terminal in domain A can use an application program in domain B to process one type of data and an application program in domain C to process another type of data.

Elements of an ACF/VTAM System

Figure 1-8 shows the major elements of a single-domain ACF/VTAM system. Each element is described below.

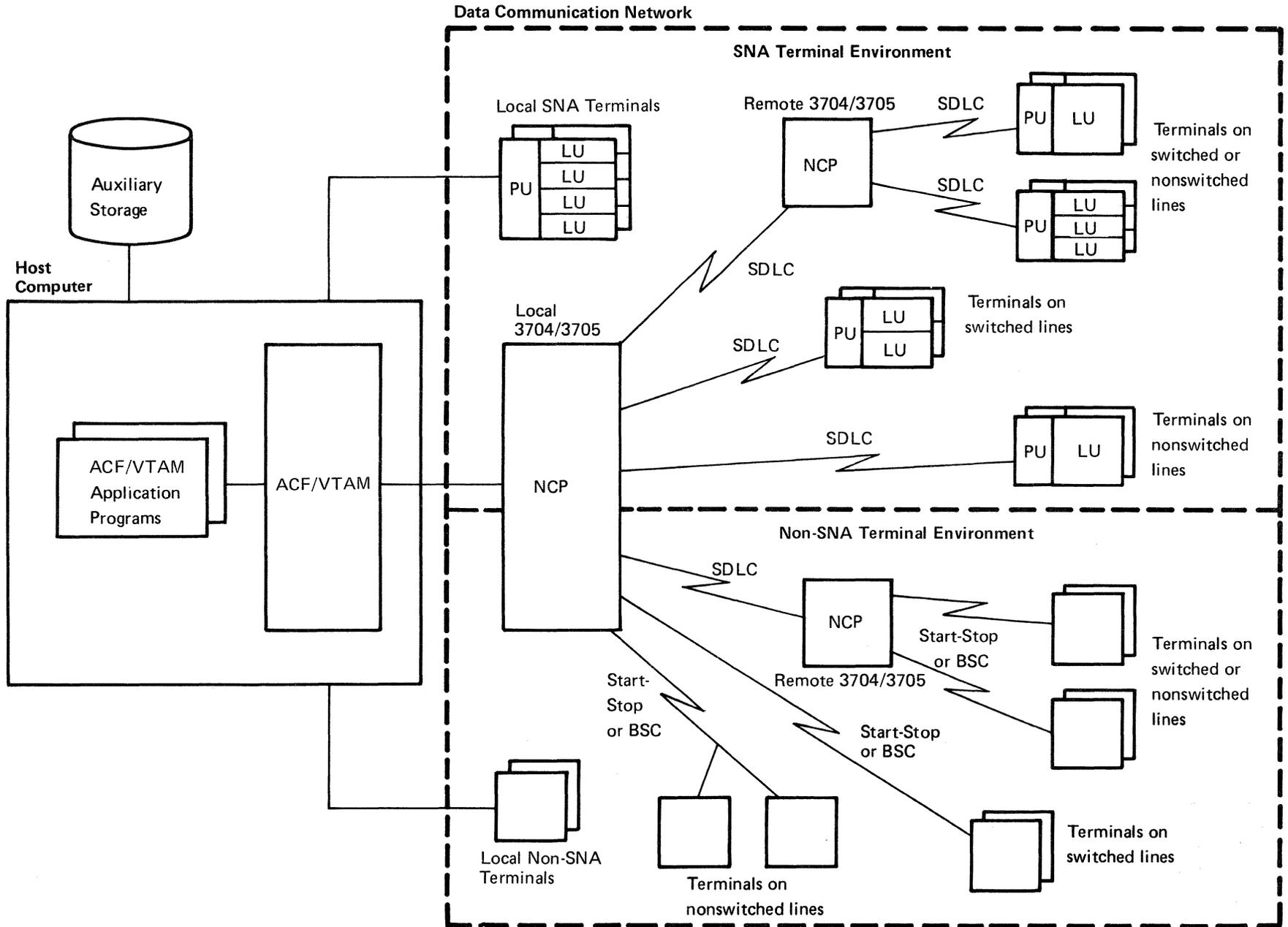


Figure 1-8. The Major Elements in a Single-Domain ACF/VTAM Network

ACF/VTAM Application Programs

An ACF/VTAM application program is any program that uses ACF/VTAM macro instructions. An ACF/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), and is capable of referring to terminals and application programs symbolically.

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

An ACF/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. ACF/VTAM permits the part of an ACF/VTAM application program that processes data to be separate from the part that communicates with terminals and other application programs. This separation allows each part to be created independently and means that changes 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 data communication part (the part that uses ACF/VTAM macro instructions) must be written in assembler language. When using programmable terminals and application programs, the user decides how the processing is divided between them.

Communications Controllers

ACF/VTAM uses the IBM 3704, 3705-I, and 3705-II Communications Controllers to communicate with remote elements in the 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 local communications controller by a communication line). Communications controllers link ACF/VTAM with the remote portions of the network and control the flow of information between terminals and ACF/VTAM. Local 3705 Communications Controllers are used to connect domains; that is, one domain can be connected to another domain only by a communication link between two local 3705 controllers or through a local 3705 controller that is channel-attached to both host computers.

The communications controllers are programmable devices; the program that controls these devices for ACF/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 communication controller to handle separate communication lines in network control and emulation mode at the same time. ACF/VTAM uses only the network control mode of the NCP, with or without PEP.

Two types of NCPs can be used with ACF/VTAM. NCP/VS Version 5 can be used in a 3704 or 3705 Communications Controller for controlling communication within a domain. Advanced Communications Function for NCP (ACF/NCP/VS) can be used in a 3705-I or 3705-II Communications Controller for communications within a domain or for cross-domain communications. In this manual, *NCP* refers to either type of NCP when the discussion involves communication within a domain and to ACF/NCP/VS when the discussion involves cross-domain communication.

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 buffering
- Deleting and inserting line-control characters
- Detecting permanent line errors
- Gathering line statistics
- Activating and deactivating lines
- Closing down the domain
- Handling recoverable errors
- Providing error statistics to ACF/VTAM
- Controlling cross-domain communication (ACF/NCP/VS only)

In addition, for start-stop and BSC (binary synchronous communication) 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 ACF/VTAM. For example, when the NCP is to deactivate a line, the command to deactivate comes from ACF/VTAM. ACF/VTAM, in turn, is controlled by the user's definition of the system, by the network operator, and by the ACF/VTAM application programs.

Terminals

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

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

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

ACF/VTAM also supports the local attachment of non-SNA 3270 terminals and SNA terminal systems through a data channel to the host computer.

Application programs, remote SNA terminals, and specially defined BSC 3270 terminals used in record mode can communicate between domains; local, start-stop, and BSC terminals (except specially defined BSC 3270 terminals) can communicate only within their domain.

SNA Terminals

Each SNA terminal is supported remotely over an SDLC switched or nonswitched line or locally (through a channel interface). To ACF/VTAM, an SNA terminal is either a physical unit (PU) or a logical unit (LU) that is 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 to 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 an ACF/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. In addition, an application program can function as an SNA terminal (that is, as a secondary logical unit) when communicating with another application program. To ACF/VTAM, both ends of a session—the primary end and the secondary end—are logical units.

Non-SNA Terminals

ACF/VTAM supports the local and remote non-SNA terminals listed in Appendix A. Non-SNA terminals include local non-SNA 3270, BSC, and start-stop terminals. With these devices, there is a one-to-one relationship between the terminal and the physical device; for example, a 2741 is a terminal to ACF/VTAM and also a physical end-point in a network.

Chapter 2. An ACF/VTAM System

ACF/VTAM performs services for the elements in its domain. This chapter describes those elements, the relationship among the elements, and the steps involved in creating a data communication system composed of those elements.

Views of an ACF/VTAM System

To understand ACF/VTAM, a user should know what the system looks like from several viewpoints. These viewpoints are shown in Figure 2-1 for a single-domain network and Figures 2-2 through 2-7 for a multiple-domain network.

A Single-Domain System

Part A of Figure 2-1 shows a possible physical configuration of a single-domain system. The system includes a host computer with a system console. The system console is used to enter ACF/VTAM network operator commands to control the system. Also attached to the host computer are auxiliary storage devices that contain data sets used by ACF/VTAM.

The network shown in Section A includes a local non-SNA 3270 Information Display System, a local 3790 Communication System, and a local 3705 Communications 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 data communication system has a definite physical configuration, but its definition and uses are different for the operating system, ACF/VTAM, and application programs. Parts B, C, and D of Figure 2-1 show these differences.

Part B of Figure 2-1 depicts the ACF/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 ACF/VTAM; such support is generated within ACF/VTAM.

The number of auxiliary storage devices used by ACF/VTAM depends upon data requirements, which in turn are influenced by factors such as the size and complexity of the data communication system. In general, data used by or generated by ACF/VTAM falls into one of three categories as shown in Part B of Figure 2-1:

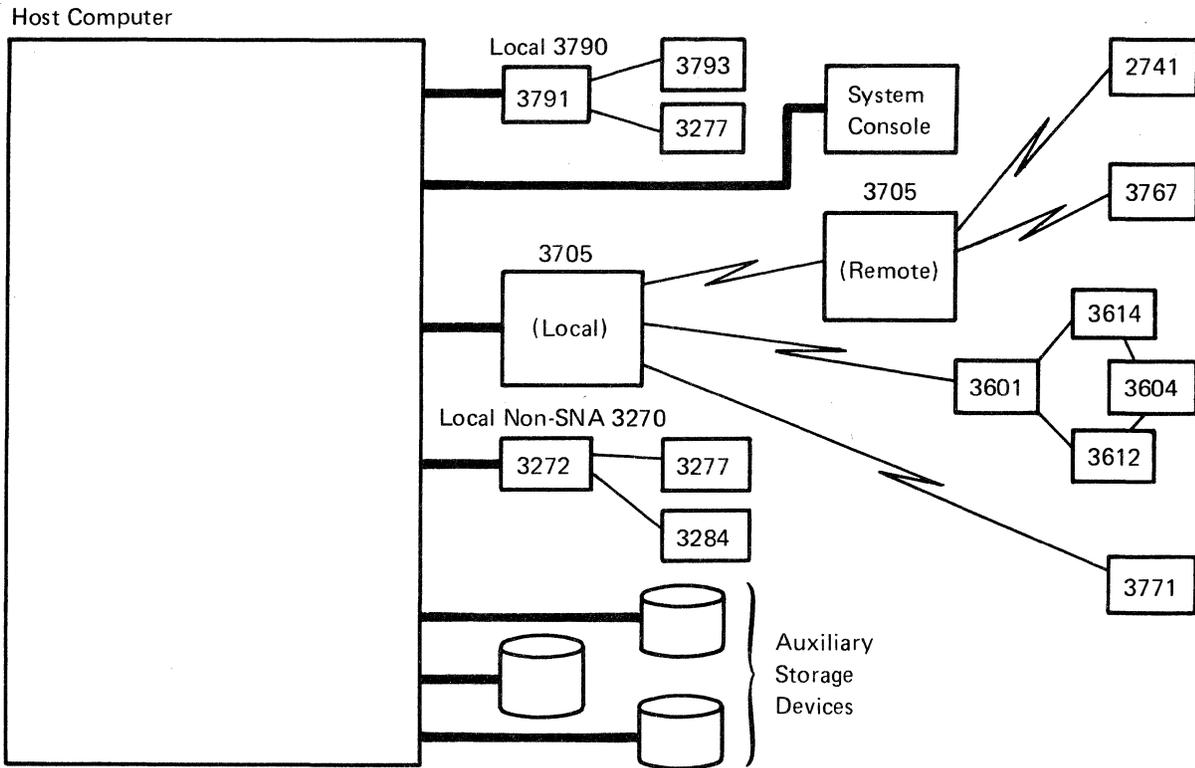
ACF/VTAM libraries, which contain ACF/VTAM load modules, descriptions of the system, and operational specifications of the installation

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 ACF/VTAM system

ACF/VTAM and NCP libraries include ACF/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 ACF/VTAM is being executed. (See Chapter 7 for details on operating system requirements and data set requirements for ACF/VTAM.)

A Physical Configuration of a Data Communication System



B The Data Communication System Viewed by the Operating System

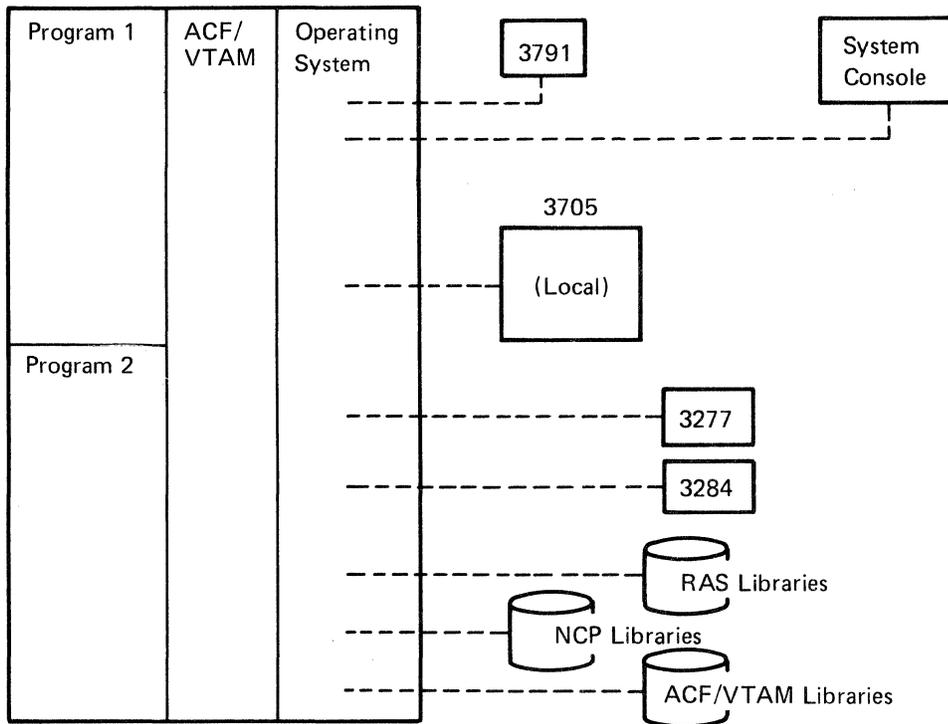
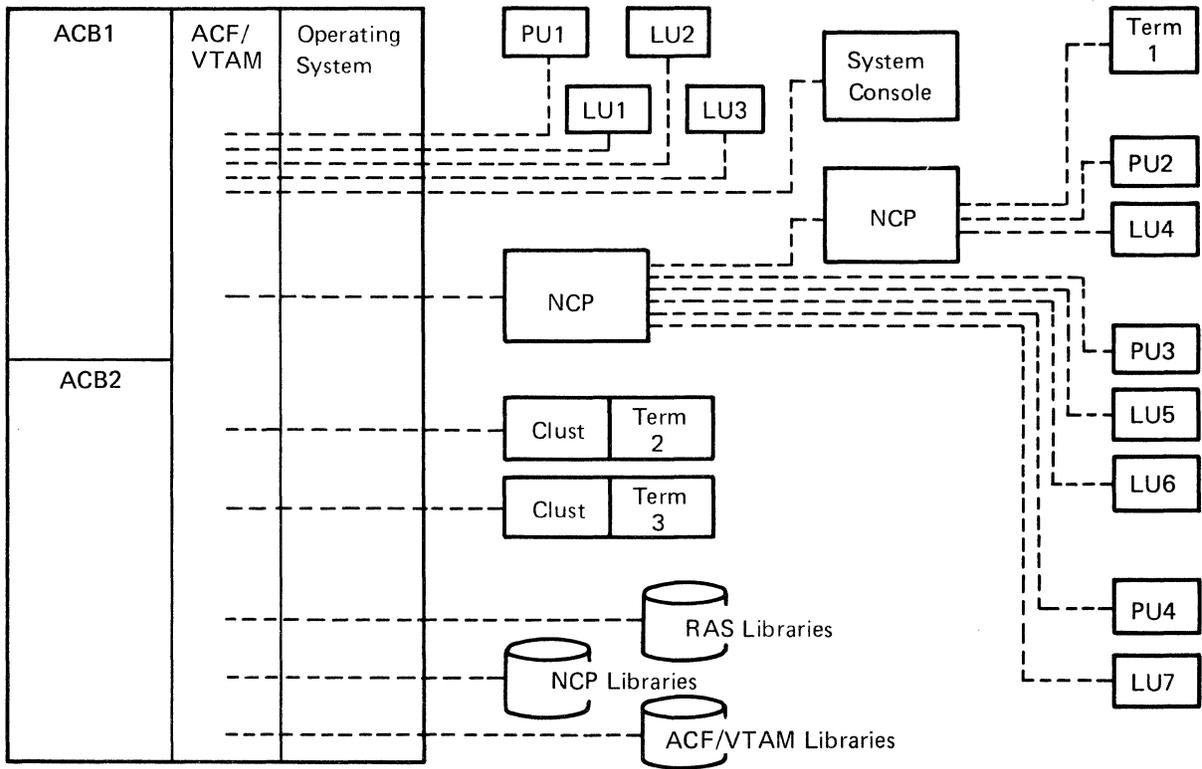


Figure 2-1 (Part 1 of 2). Four Views of an ACF/VTAM Data Communication System

C The Data Communication System Viewed by ACF/VTAM



D The Data Communication System Viewed by ACF/VTAM Application Program 1

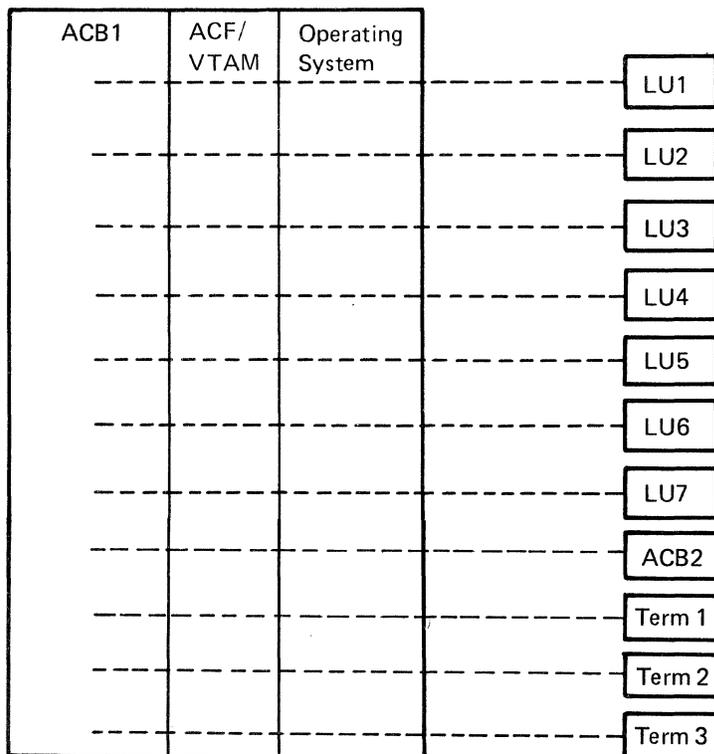


Figure 2-1 (Part 2 of 2). Four Views of an ACF/VTAM Data Communication System

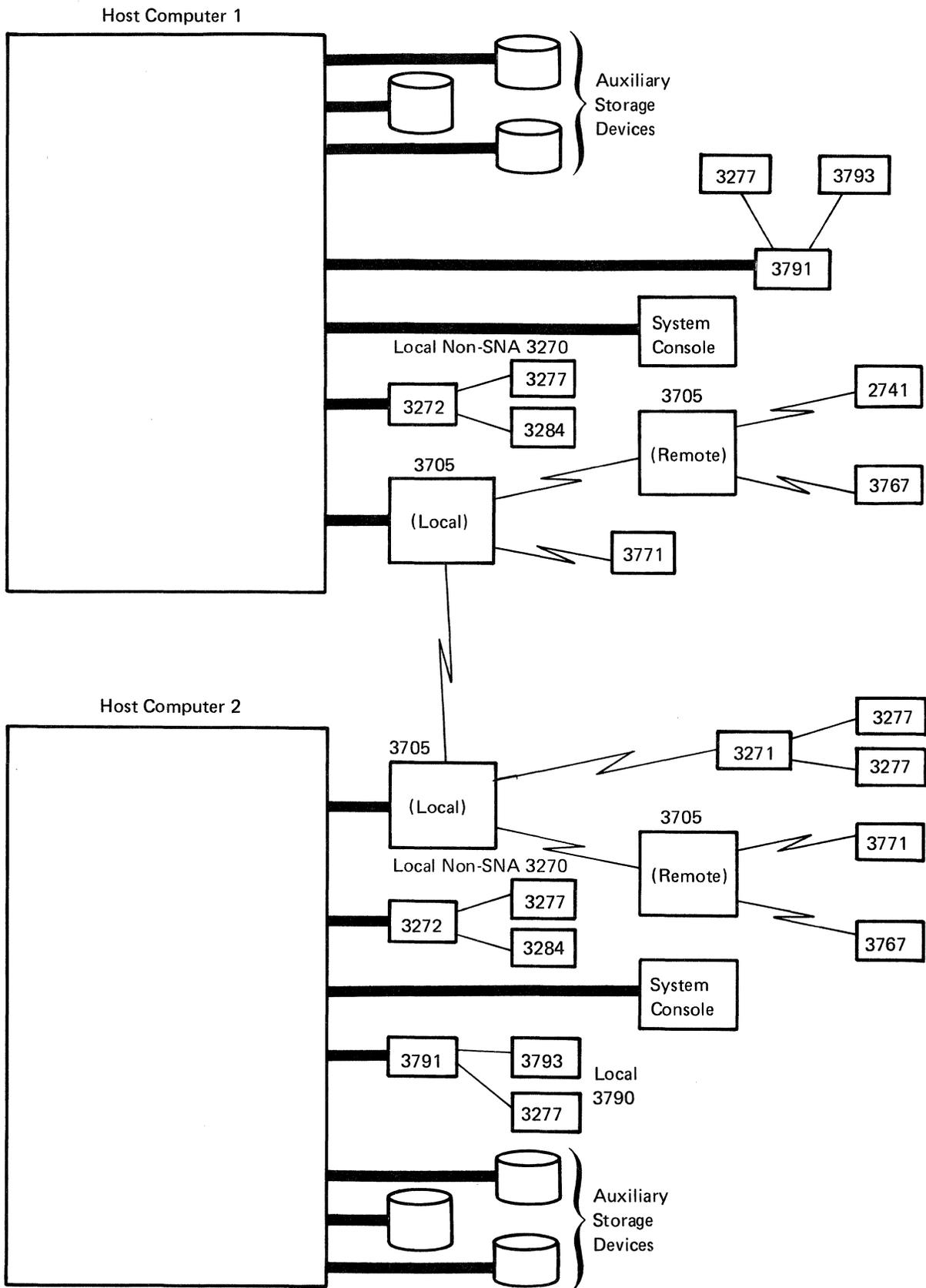
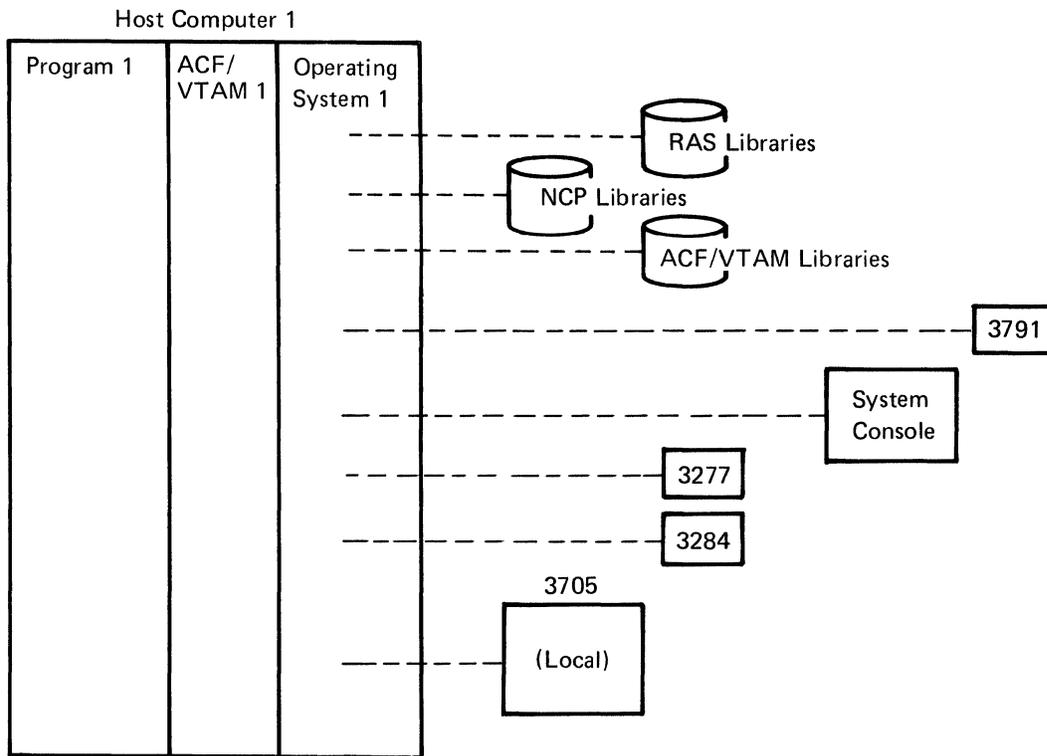


Figure 2-2. Physical Configuration of a Multiple-Domain ACF/VTAM Network

A. Operating System 1's View of the Network



B. Operating System 2's View of the Network

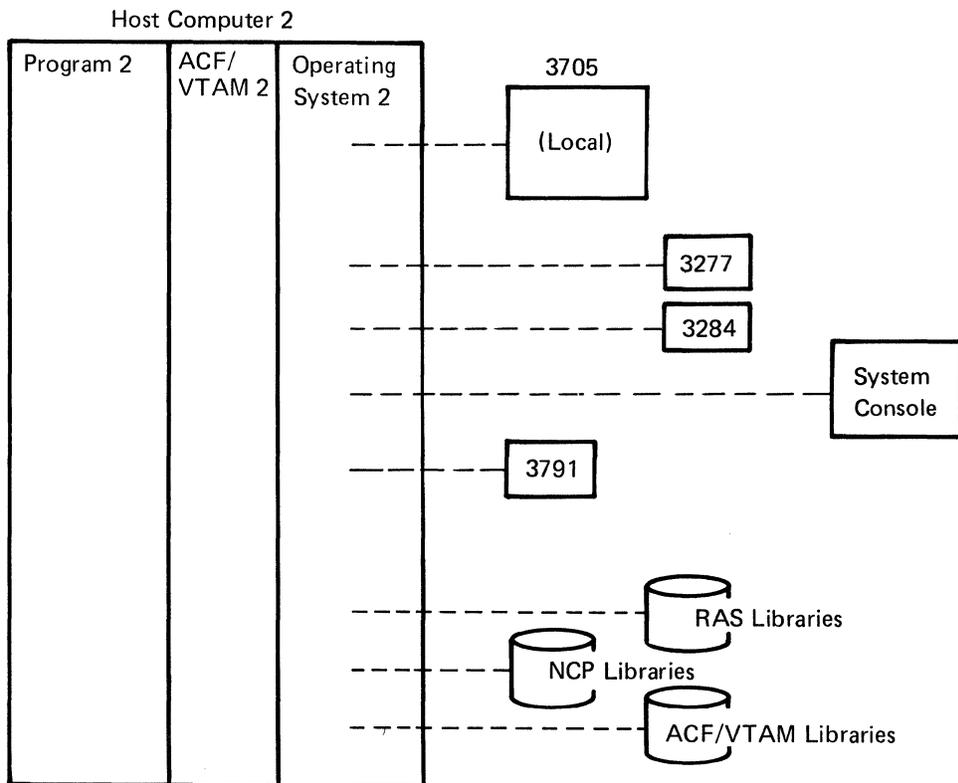
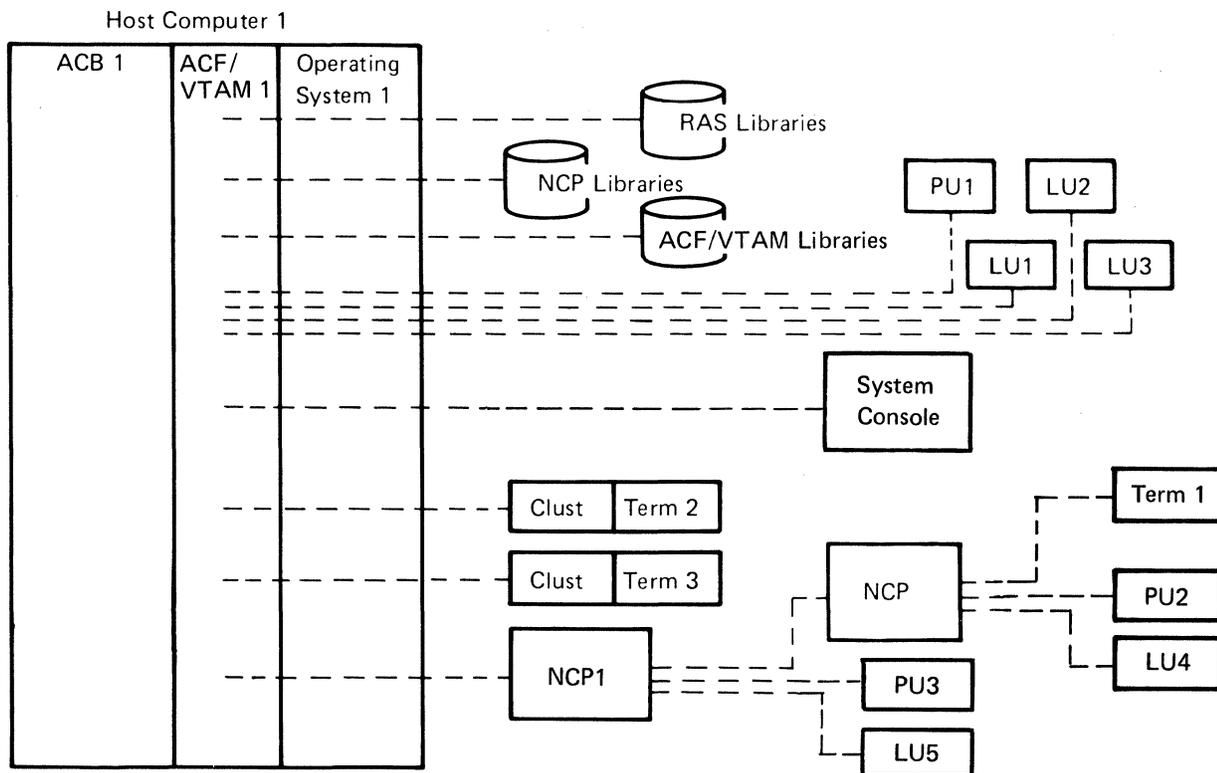


Figure 2-3. The Network As Viewed by the Host Operating Systems

A. ACF/VTAM 1's View of its Domain



B. ACF/VTAM 2's View of its Domain

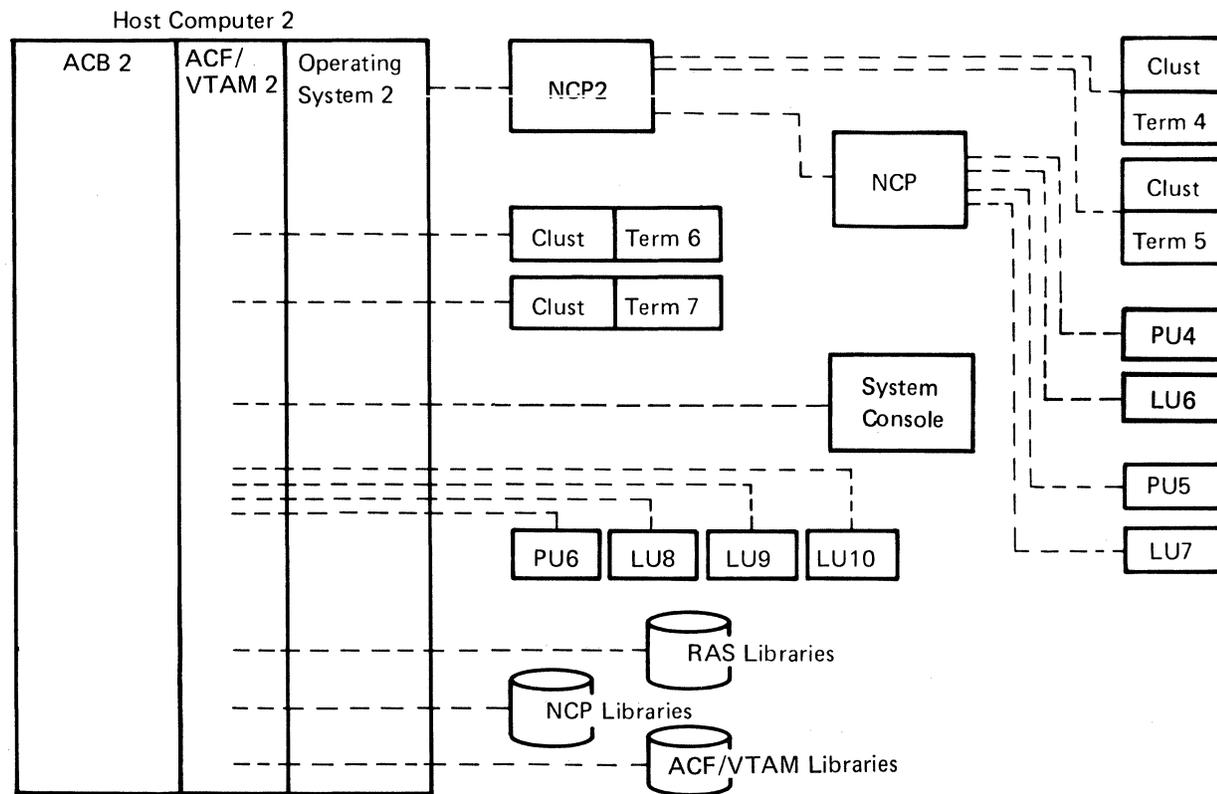


Figure 2-4. The Domains As Viewed by ACF/VTAM

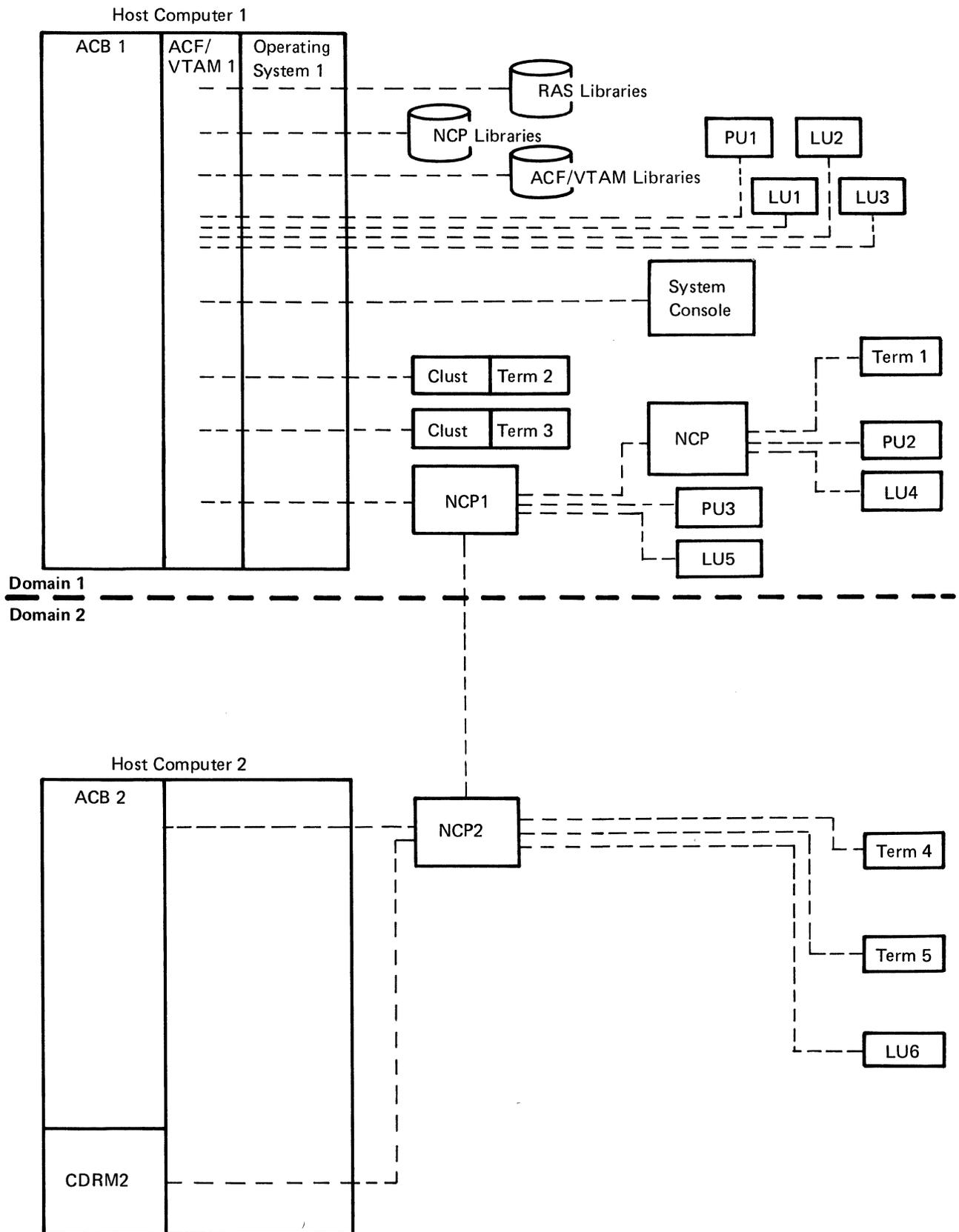


Figure 2-5. The Network As Viewed by ACF/VTAM 1

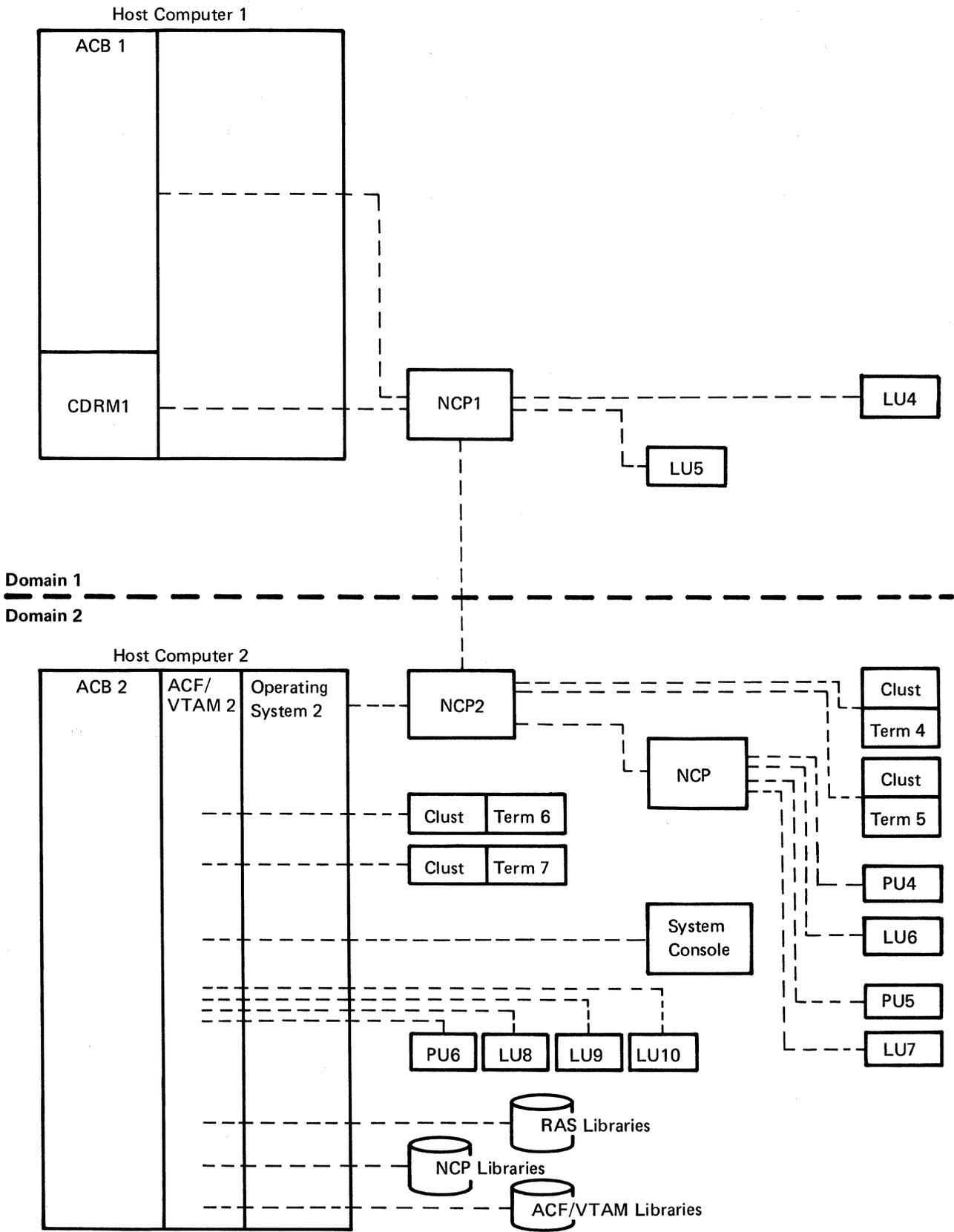


Figure 2-6. The Network As Viewed by ACF/VTAM 2

Part C of Figure 2-1 depicts the data communication system as viewed by ACF/VTAM. The host computer must contain the operating system (DOS/VS, OS/VS1, or OS/VS2), ACF/VTAM, and one or more application programs. To ACF/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 ACF/VTAM is started and begins activating parts of the data communication network, ACF/VTAM acquires the use of these devices.

ACF/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. This is because the logical unit (a program in the cluster control unit) usually controls more than one device attached to the control unit. ACF/VTAM sees each non-SNA terminal product as a single terminal (or terminal with components) or as a cluster controller with associated terminals.

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

Part D of Figure 2-1 shows the data communication system as viewed by an application program. This view results from ACF/VTAM's ownership of all elements in the network and the way ACF/VTAM allocates them. ACF/VTAM connects application programs to terminals (logical units, non-SNA terminals, and other application programs)¹. The intermediate elements are allocated only for the time needed to satisfy a specific transmission request.

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

A Multiple-Domain System

Figures 2-2 through 2-7 show a multiple-domain ACF/VTAM system as viewed in terms of physical components (Figure 2-2), as seen by the host operating systems (Figure 2-3), as seen by the ACF/VTAMs in each host computer (Figures 2-4, 2-5, and 2-6), and as seen by application programs in each host computer (Figure 2-7). Figure 2-2 shows two domains connected by an SDLC link between two local 3705 Communications Controllers. Each domain has its own auxiliary storage devices, system console, and local and remote communication devices.

As shown in Figure 2-3, each operating system sees only the local devices attached to its host computer; the operating systems are not concerned with remote devices either in their own domain or in other domains. Operating system 1 and operating system 2 need not be of the same type as long as they support compatible ACF/VTAMs. For example, operating system 1 can be DOS/VS and operating system 2 can be OS/VS2 MVS.

Figure 2-4 shows how ACF/VTAM in each host computer sees its own domain. To each ACF/VTAM, its domain appears the same as a single-domain network.

¹In this publication, logical units, application programs acting as secondary ends of sessions, and non-SNA terminals are collectively called *terminals*.

Figures 2-5 and 2-6 show the system as viewed by the two ACF/VTAMs. ACF/VTAM 1 sees its domain in the same way as in a single-domain system.² In addition, it sees the cross-domain resource manager for the other ACF/VTAM (CDRM2) and the resources in the other domain that are available for cross-domain communication. A resource in the other domain is available if it has been defined properly to ACF/VTAM 1 as a cross-domain resource. If not defined in that way, ACF/VTAM 1 does not have access to it. ACF/VTAM 1 does not see the physical units associated with the logical units nor does it see the local terminals and remote non-SNA terminals (except specially defined BSC 3270s), because they do not communicate across domain boundaries. The user can also have remote logical units that are not used for cross-domain communication. For example, as shown in Figure 2-5 (as related to Figure 2-2), ACF/VTAM 1 does not see the remote 3767 in domain 2. Similarly, ACF/VTAM 2 sees its domain, ACF/VTAM 1's cross-domain resource manager, and the list of ACF/VTAM 1's remote logical units in the same way, as shown in Figure 2-6.

Figure 2-7, shows the system as viewed by application programs in each host computer. The application program in host computer 1 identified by ACB1 sees the logical units and non-SNA terminals in its own domain and the cross-domain resources of domain 2. It is not aware of whether a resource is locally attached, remotely attached in its own domain, or in another domain. It also is not aware that ACB2 represents an application program; it sees it as a logical unit. This relationship is described in more detail in Chapter 5. Similarly, the application program in host computer 2 sees its own domain's logical units and non-SNA terminals as well as the cross-domain resources of domain 1.

Installing an ACF/VTAM System

To install an ACF/VTAM system, it must be created, application programs must be designed and coded, procedures must be defined for using the system, and the active system must be controlled. These steps are discussed below.

Creating the ACF/VTAM System

An ACF/VTAM data communication system is created by installing the program product modules in the distribution libraries of the operating system, generating the operating system of each host computer, defining the network to the communications controllers and to each ACF/VTAM, and defining special ACF/VTAM facilities.

Generating ACF/VTAM is part of operating system generation. Defining the network to ACF/VTAM is a separate process of identifying and describing elements of the network and then filing these definitions in a ACF/VTAM library. Defining special ACF/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 ACF/VTAM's logon facility to the user's specifications. Chapter 3 describes these procedures in detail.

Controlling the ACF/VTAM System

ACF/VTAM enables a network operator to dynamically control an ACF/VTAM domain. The network operator can start and stop ACF/VTAM, monitor the activity of the domain, activate and deactivate network elements such as data links, physical units, and

²In this discussion and in the figures, the names *ACF/VTAM 1* (meaning "ACF/VTAM No. 1"), *ACF/VTAM 2*, *NCP 1*, and *NCP 2* are used to designate separate ACF/VTAMs and NCPs. The names are not meant to indicate levels or releases of the product (that is, ACF/VTAM 1 does *not* indicate Level 1 of ACF/VTAM).

logical units, and start and stop specified ACF/VTAM facilities. To perform these functions, the network operator uses a set of ACF/VTAM commands. In a multiple-domain network, network operators in all domains must coordinate control of the ACF/VTAM system. See Chapter 4 for detailed information on the responsibilities and actions of the network operator and for a description of the ACF/VTAM network operator facilities.

Designing ACF/VTAM Application Programs

ACF/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 ACF/VTAM services synchronously (the program waits while ACF/VTAM performs the requested operation) or asynchronously (the program continues execution and is interrupted when ACF/VTAM has completed the operation). Chapter 5 introduces the ACF/VTAM facilities available to the application program.

Establishing Procedures for Using the System

Once an ACF/VTAM system has been started, it is available to application programs, terminals, and the network operator. To ensure that the 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 ACF/VTAM system. The user must define how and when to activate and deactivate nodes and specific ACF/VTAM functions. The user must also define what to do when error conditions are encountered and what action to take to avoid unnecessary downtime. These actions might include responding to error messages, collecting status information, or correcting the problem. Because a network operator cannot monitor and control elements in another domain, procedures must be established for coordinating the actions of network operators in different domains.

The application programmer needs to know the conventions to be followed when connecting programs to ACF/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 ACF/VTAM resources. ACF/VTAM facilities can be used to control connections between application programs and terminals. Facilities are also available to restrict the use of certain ACF/VTAM services to authorized users and to protect confidential data.

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

How ACF/VTAM Operates

When ACF/VTAM is started, it initializes its domain according to the specifications in the network definition statements. When the initialization is completed, ACF/VTAM is ready to respond to network operator commands and to serve ACF/VTAM application programs.

After initialization, the network operator may want to issue commands to activate various elements in the network and to initiate various activities, such as tracing.

An ACF/VTAM application program is started like other problem programs in the operating system. After the application program has been started, it cannot use any ACF/VTAM facilities until it has identified itself to and associated itself with ACF/VTAM. The application program does this by opening an access method control block (ACB). After the ACB has been opened, ACF/VTAM can perform other data communication operations requested by the application program.

The application program cannot communicate with a terminal until it has been connected to the terminal. The initiative to establish the connection can come from the application program itself, either as a request to ACF/VTAM to *acquire* a connection with the terminal or a request that ACF/VTAM create a logon and make the logon appear as though it came from the terminal. In either case, if the terminal is available and is willing to accept the connection, the connection is established.

In other cases, the initiative for the connection comes from the terminal itself in the form of a logon that requests connection to the application program. (More specifically, a logon is a request asking the application program to take action to have ACF/VTAM establish the connection.) Besides coming from a terminal, a logon can come from other sources:

- It can come from ACF/VTAM initialization (automatic logon specified in a terminal's definition statement).

- It can come from the network operator (automatic logon specified in a network operator command).

- It can come from another application program that has passed the terminal to the new program.

Regardless of the source of the logon, the connection is not established unless the application program *accepts* the logon by issuing a request to ACF/VTAM to set up the connection. Once the connection has been established, the application program is in *session* with the terminal; that is, ACF/VTAM has set up control information and completed session initialization activities so that the application program and terminal can begin to communicate with each other.

Sessions can also be established between two application programs. In this case, one program (which will function as the secondary end of the session) requests the other application program (which will function as the primary end) to establish a connection. The secondary program's request reaches the primary program in the form of a logon. As with a logon from or for a device-type terminal, the primary application program can either accept the logon or reject it. If the primary program accepts the logon, the secondary program must approve session protocol rules (called *session parameters*) before the session is actually established.

An application program uses macro instructions to send and receive data. When the program sends data to a terminal (rather than to a secondary application program), ACF/VTAM moves the data from the application program's output area to ACF/VTAM buffers, where it is held until ACF/VTAM can issue the I/O instruction that transmits the data. ACF/VTAM then sends the data to the terminal, either directly across a channel (if the terminal is locally attached) or through intervening communications controllers (if the terminal is remote).

Input from a terminal travels a similar (but reverse) route. The input moves from the terminal to ACF/VTAM, travelling through intermediate communications controllers

when it is coming from a remote terminal or through the channel directly from a local terminal. The input is stored in ACF/VTAM buffers until the application program asks for it, at which time it is moved to the application program input area.

When data is being transmitted to another application program, the data is moved from the sending program's output area to ACF/VTAM's buffers. For application programs in the same host computer, ACF/VTAM then moves the data to the receiving program's input area when the receiving program requests it. For application programs in different host computers, ACF/VTAM transmits the data to the receiving program through the intervening communications controllers and through the ACF/VTAM in the other host computer.

When an application program no longer needs a terminal or when a terminal informs the application program that communications are finished, the application program can disconnect the terminal. The application program does this by issuing a macro instruction that requests ACF/VTAM to break the connection. The application program can specify that the terminal is to be *released* for possible use by another application program, or the owning application program can specify that the terminal is to be *passed* to another application program.

When an application program has ended communications with all terminals and has disconnected from them, the application program can disassociate itself from ACF/VTAM (by closing its ACB) and then terminate itself.

When the data communication system is being closed down (either as the result of an operator's command to halt ACF/VTAM or as the result of an internal error), ACF/VTAM in most cases allows the application program to cease communication and terminate processing in an orderly manner. In a few cases (such as abnormal termination of ACF/VTAM), communication and processing are stopped abruptly.

From the time ACF/VTAM is started until the time it is terminated, the network operator controls and monitors the domain. Most modifications to the network can be made dynamically, without terminating ACF/VTAM.

Chapter 3. Creating an ACF/VTAM System

Creating a single-domain ACF/VTAM system consists of:

- Defining ACF/VTAM and local devices to the operating system
- Generating one or more network control programs (NCPs) if ACF/VTAM is to communicate with remote devices
- Defining the domain to ACF/VTAM
- Defining connection and disconnection procedures for terminals
- 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 ACF/VTAM buffer pools to be used for incoming and outgoing data and for control blocks)

Creating a multiple-domain ACF/VTAM system consists of creating each domain (as outlined above) and defining the following to each access method:

- The communication paths to be used for cross-domain communication (path table)
- The portion of ACF/VTAM that controls cross-domain connections (cross-domain resource manager)
- The resources in other domains that can be used (cross-domain resources)

This chapter describes in general how an ACF/VTAM domain is created to meet installation requirements. The *ACF/VTAM System Programmer's Guide* and the *ACF/VTAM Installation Guide* for each operating system describe these procedures in detail. Creating an ACF/VTAM system may also include defining network operating procedures, described in Chapter 4, and writing ACF/VTAM application programs, described in Chapter 5.

Figure 3-1 summarizes the steps in creating an ACF/VTAM 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 ACF/VTAM, consists not only of describing the structure of the network to ACF/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 domain, 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 an ACF/VTAM system.

Defining ACF/VTAM and Local Devices to the Operating System

Before ACF/VTAM can be added to the operating system, the ACF/VTAM modules must be installed in the distribution libraries (DLIBs) for the operating system. Then, to include the ACF/VTAM modules and the required support in the operating system, the user can either (1) employ the System Modification Program (SMP) to incorporate the new modules in the system or (2) perform a system generation. If a system generation is performed, 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.

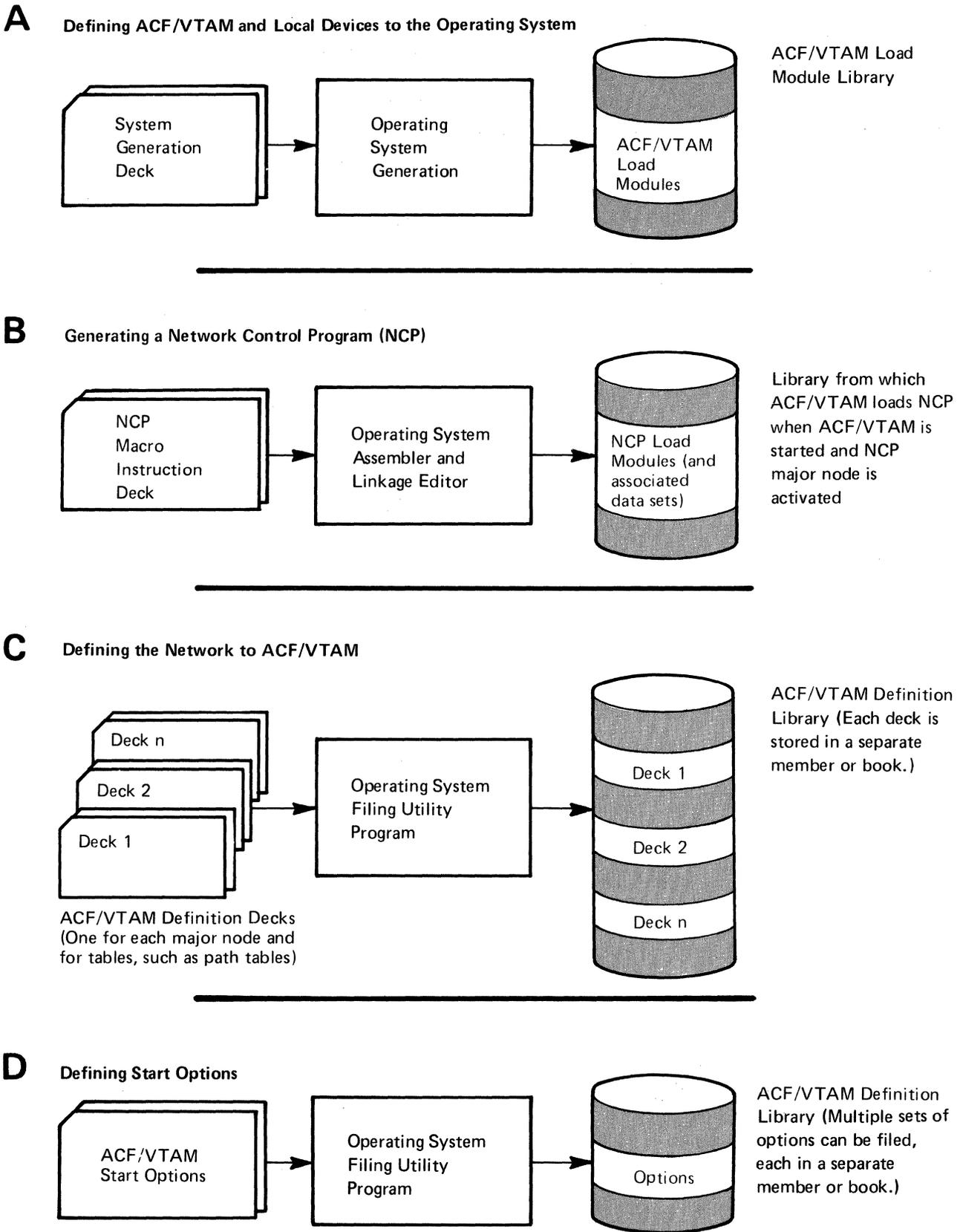
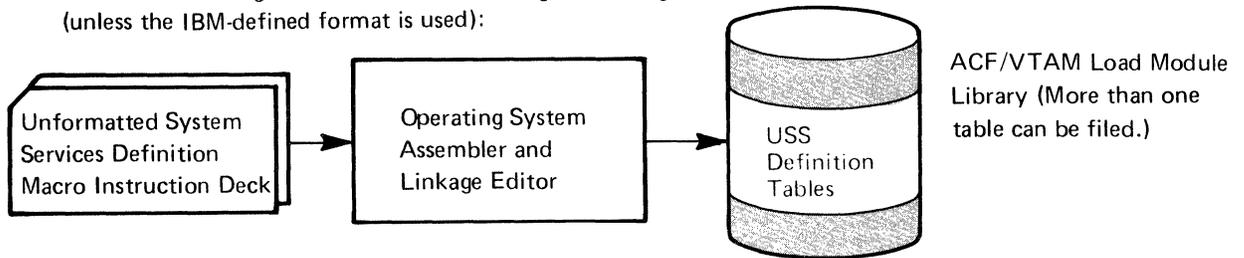


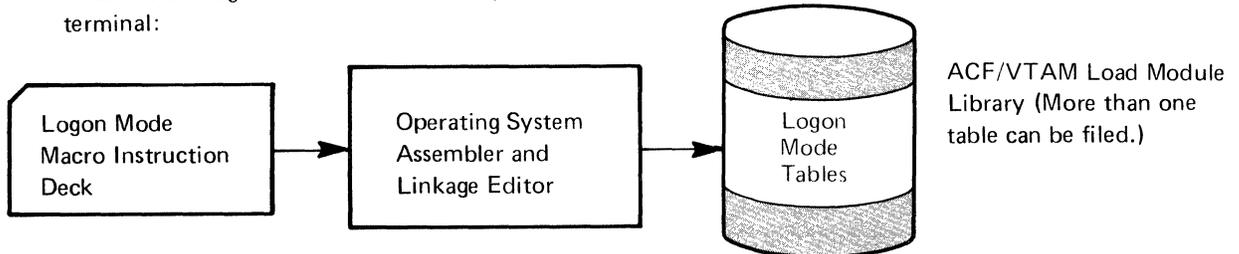
Figure 3-1 (Part 1 of 2). The Steps in Creating an ACF/VTAM Domain

E Defining Connection and Disconnection Procedures

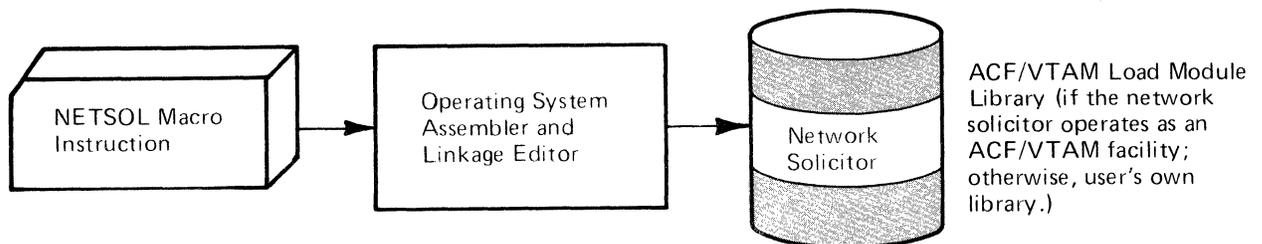
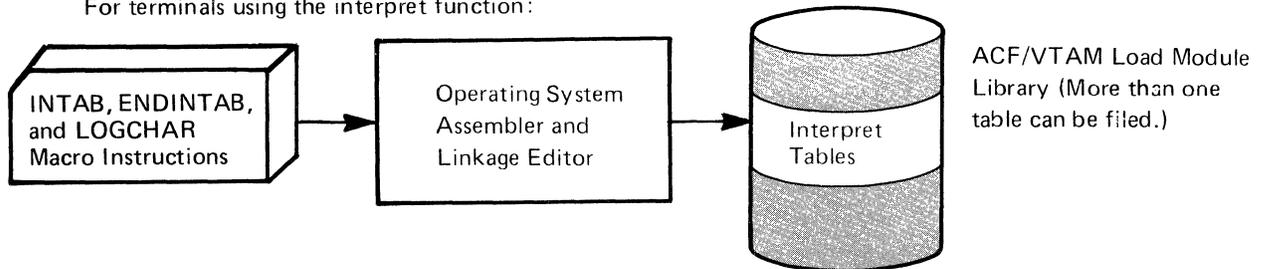
For terminals using character-coded (USS) logons and logoffs (unless the IBM-defined format is used):



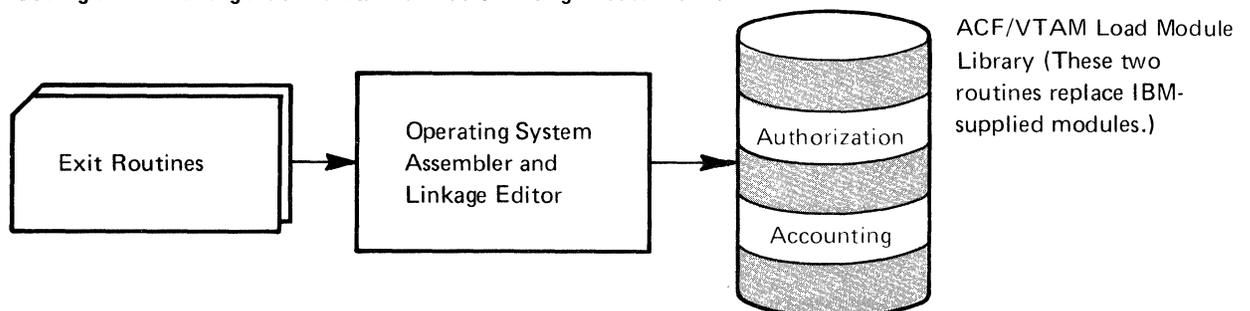
To define the logon mode (set of session parameters) for each terminal:



For terminals using the interpret function:



F Coding and Including Authorization and Accounting Exit Routines



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 an ACF/VTAM Domain

Control unit and device statements are specified only for locally attached devices that use ACF/VTAM, that is, for local non-SNA 3270s, local SNA cluster control units, and local communications controllers.

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

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

Generating a Network Control Program (NCP)

If the ACF/VTAM data communication 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 ACF/VTAM system is started. (The NCP can be loaded automatically, if required, as the result of starting ACF/VTAM). The macro instructions used to code an NCP are described in the *NCP Generation* manual, which can be used with the *ACF/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 ACF/VTAM.

ACF/VTAM considerations when writing the set of NCP macro instructions are described in this chapter in “Defining NCP Major Nodes.”

Defining the Network to ACF/VTAM

Defining the network to ACF/VTAM consists of describing the domain configuration and its characteristics to ACF/VTAM and, for multiple-domain networks, describing resources in other domains that are available to the domain being defined. These descriptions are coded in ACF/VTAM definition statements; the coded definitions are then filed in the ACF/VTAM definition library.

As part of the network definition, major nodes in the domain must be defined to ACF/VTAM. A major node can be any of the following:

- An NCP for a local or remote communications controller, including its attached links, logical units, physical units, BSC terminals, and start-stop terminals
- A set of logical units and physical units on switched lines
- A set of logical units and physical units attached locally
- A set of non-SNA terminals attached locally
- A set of application programs that use ACF/VTAM
- A set of cross-domain resource managers
- A set of cross-domain resources

The definition of each major node is filed separately (as a member in OS/VS or as a book in DOS/VS) in the ACF/VTAM definition library. Normally, a major node definition is filed in the library at some point after system generation but before starting ACF/VTAM,

or at some point after ACF/VTAM is stopped and before it is restarted (with a cold start). The definitions of some major nodes can be replaced in the definition library while ACF/VTAM is running, but the user must take precautions against trying to replace the definition at the time it is being used by ACF/VTAM. In addition, in an OS/VS system, the user must delete from SYS1.VTAMOBJ the table corresponding to the old node definition before activating the new node definition (see Figure 7-4 in Chapter 7).

The sections that follow describe the concepts of major and minor nodes in ACF/VTAM and discuss the manner in which different kinds of major nodes are defined.

The Major and Minor Node Structure

Major and minor nodes¹ are the controllable elements of the ACF/VTAM network. ACF/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 ACF/VTAM.

A major node is a set of controllable elements (minor nodes) in the ACF/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 can be controlled as a whole, or the parts of it (the minor nodes) can be controlled separately.

One or more minor nodes can be defined for each major node. The name of a minor node is the name assigned to the ACF/VTAM statement that defines it. The types of minor nodes in a major node depend on the type of major node being defined. Figure 3-2 shows the types of major and minor nodes in a single-domain ACF/VTAM system. Figure 3-3 shows two additional types of major and minor nodes in a multiple-domain ACF/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.

For some types of major nodes, each major node is given a particular *subarea number* to distinguish it from other major nodes in the network. ACF/VTAM uses the subarea numbers as part of the network address for each element in each major node and later uses the subarea number for routing of commands and messages. Some subarea numbers

¹The terms *major node* and *minor node* are not related to such terms as *host node* and *communications controller node* used in SNA publications.

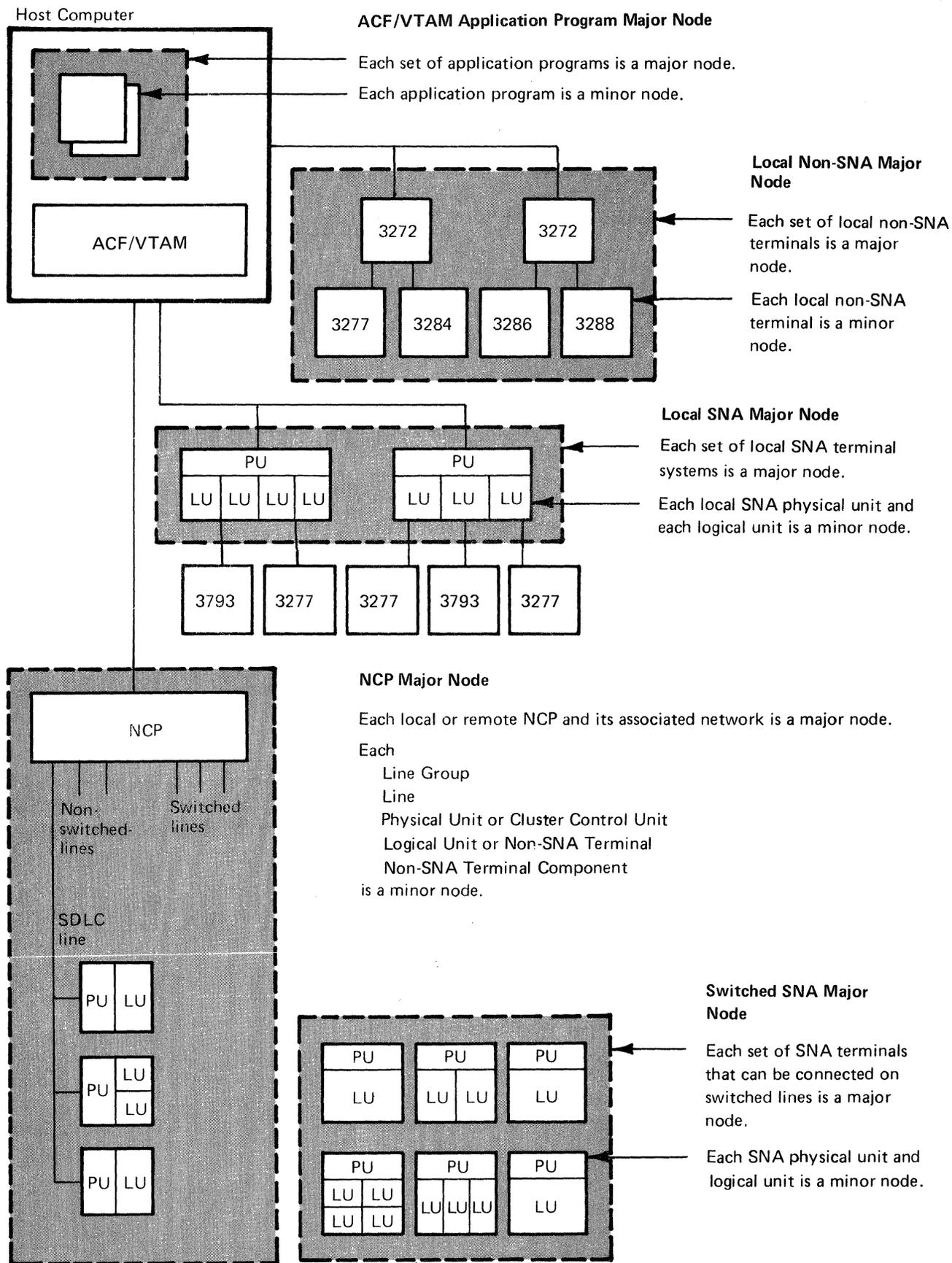


Figure 3-2. Major and Minor Nodes in a Single-Domain ACF/VTAM System

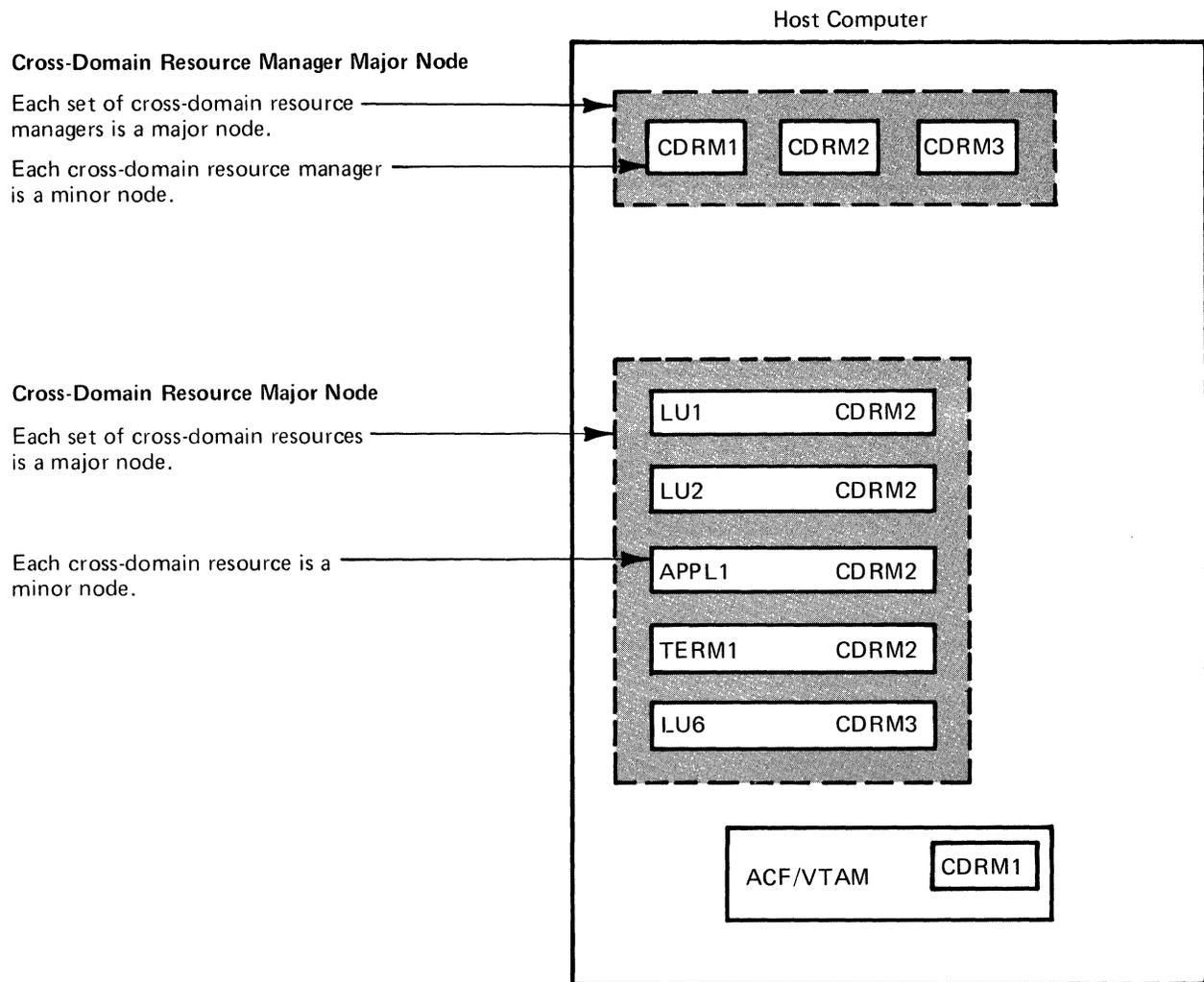


Figure 3-3. Additional Major and Minor Nodes in a Multiple-Domain ACF/VTAM System

are designated by the user in definition statements, while others are assigned by ACF/VTAM. The user specifies a subarea number in a definition statement for each local non-SNA major node, each local SNA major node, each NCP major node, and each ACF/VTAM in the network. ACF/VTAM assigns a subarea number to each application program major node. The application program major node gets the subarea number of the ACF/VTAM with which it becomes associated. ACF/VTAM also assigns a subarea number to each physical unit in a switched SNA major node. Each physical unit in a switched SNA major node gets the subarea number of the NCP to which it becomes attached.

In a single-domain network *and* in a multiple-domain network, each subarea must have a unique subarea number. Without unique subarea numbers, ACF/VTAM cannot properly route messages.

Naming Major and Minor Nodes

Each major and minor node must be named. The name of each major node is the name of the member (in OS/VS) or book (in DOS/VS) that contains 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 within a domain.

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

Duplicate minor-node names can be used in different major nodes as long as only one of those major nodes is active at a time. If the network operator issues an activation request for a major node that contains a duplicate minor-node name, any minor-node with a duplicate name is not activated and is not available.

Duplicate names are not permitted for minor nodes that can be used for cross-domain communication. (If the same application program is to be active in more than one domain and the individual programs are to be accessed by terminals in other domains, the symbolic name of each program—as provided in the name field of the APPL statement—must be unique in the network. However, the same ACBNAME can be specified in the operand portion of the APPL statements for all of the programs. A cross-domain logon for a particular program must specify the unique symbolic name; a same-domain logon can specify the ACBNAME or the symbolic name.)

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, and in an OS/VS system, the names ISTATA00 and VTAMTERM must not be used.

Names beginning with IST should be avoided because ACF/VTAM uses such names.

Controlling the assignment of node names provides some control over the security of the ACF/VTAM system. See “ACF/VTAM Security” in Chapter 7 for more information on using node names to protect the system.

Defining Application Program Major Nodes

The application programs in a domain can be defined as one or more major nodes. Each major node represents a set of application programs; each minor node represents a single application program. Each application program major node is defined with a VBUILD definition statement; each application program minor node is defined with an APPL definition statement. To ACF/VTAM, an application program is an open access method control block (ACB). The name specified by the ACB must match the name in the ACBNAME operand of an APPL statement, or if the ACBNAME operand was not specified, must match the symbolic name of the APPL statement.

APPL statements can be filed separately (as members for OS/VS or books for DOS/VS) in the ACF/VTAM definition library, or they can be grouped in various combinations and filed as sets. An application program can be authorized (by its APPL statement) to perform each of the following through ACF/VTAM:

- Pass terminal connections to another application program

- Issue requests to acquire terminals

- Issue ACF/VTAM network operator commands (except START and HALT) and receive ACF/VTAM network operator messages

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

In addition, the APPL statement indicates whether or not ACF/VTAM is to use the VPACING specifications for the terminals that are connected to the application program defined by the statement. VPACING is a way of restricting the data sent from the host computer to the terminal in such a way that the amount of data does not exceed the terminal's buffer capacity.

Defining Local Non-SNA Major Nodes

One or more major nodes can be defined for local non-SNA terminals. Each major node represents a set of local non-SNA terminals; each minor node represents a non-SNA terminal (such as a 3277). Each local non-SNA major node is defined with an **LBUILD** statement; each local non-SNA minor node is defined with a **LOCAL** statement.

Defining Local SNA Major Nodes

One or more major nodes can be defined for local SNA terminal systems. Each major node represents a set of SNA terminal 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 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 ACF/VTAM. For each NCP major node defined for a particular communications controller, the user generates a network control program (NCP). Only one NCP at a time can be loaded and executed in the 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 control unit (such as a 3601) or a switched-line access (port) to the communications controller. For local NCP major nodes, a physical unit minor node identifies one of the following: (1) a representation of a remote communications controller in the same domain, (2) a representation of a remote station in another independent domain (see Appendix B), or (3) in a system with the Multisystem Networking Feature, a representation of a communications controller in another domain. If the **PU** statement identifies a switched-line access or a communications controller, it does not have any logical unit minor nodes associated with the physical unit.

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 ACF/VTAM as part of a switched SNA major node.

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

Ports (for switched lines only). Each port (a point by which a dial-in line can access a communications controller) is defined partly by a **LINE** statement and partly 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 BSC or start-stop 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.

Terminal components. Each terminal 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 statements (macro instructions)² used to generate an NCP should be used as the major node definition for ACF/VTAM. Using this deck both for NCP generation and for ACF/VTAM network definition requires that ACF/VTAM-only statements and parameters be included in the NCP generation statements. See Chapter 7 for ACF/VTAM requirements for an NCP.

Figure 3-4 shows the steps for generating NCP support in an ACF/VTAM data communication system. The steps are as follows:

1. *Planning the NCP.* Keep ACF/VTAM requirements, restrictions, and considerations in mind.
2. *Coding the NCP generation statements.* Include the parameters and definition statements required by ACF/VTAM as well as those used to generate the NCP.
3. *Generating the NCP.* Use the statements coded in step 2 to generate the NCP and store the generated NCP in the NCP load module library.
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 ACF/VTAM definition library. This deck was coded in step 2 and used in step 3. When ACF/VTAM activates this NCP, ACF/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 ACF/VTAM ensures that the generated NCP agrees with its definition for ACF/VTAM. The deck is filed in the definition library by using an operating system utility program.

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 then an error is encountered during the NCP generation, a corrected deck has to be filed in the ACF/VTAM library.

If the NCP is used in a communications controller that is shared by ACF/VTAMs in two or more host computers, the user must specify which ACF/VTAM controls the resources attached to each line.

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

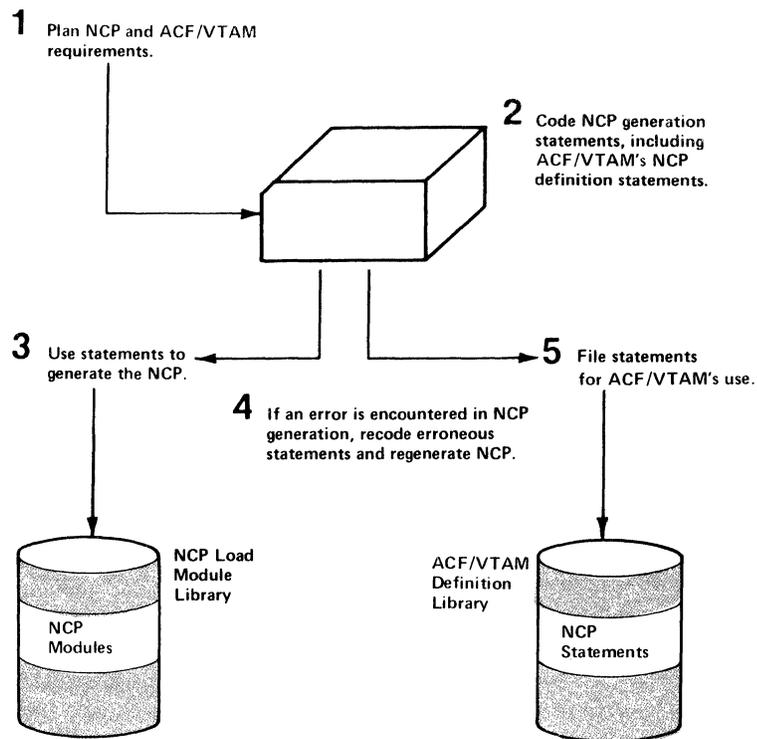


Figure 3-4. Generating NCP Support in an ACF/VTAM System

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 describes the switched SNA major node

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

- 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, the eligible dial-out ports to use, 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 Other Domains to ACF/VTAM

In a multiple-domain network, information about other domains must be provided to ACF/VTAM. This information is provided by listing the cross-domain resource managers in the network, listing the resources in other domains that can be used and associating them with the appropriate cross-domain resource managers, and describing the paths to other domains.

Defining Cross-Domain Resource Manager Major Nodes

One or more major nodes can be defined for cross-domain resource managers. A major node represents a set of one or more cross-domain resource managers; it is defined with a VBUILD definition statement followed by a CDRM statement for each minor node. A minor node defines either this domain's cross-domain resource manager or a cross-domain resource manager in another domain.

Within a domain, in addition to the definition of its own cross-domain resource manager, the cross-domain resource manager of each of the other domains with which resources of this domain will have cross-domain sessions must be defined.

Defining Cross-Domain Resource Major Nodes

One or more major nodes can be defined for cross-domain resources. Each major node represents a set of application programs, logical units, and specially defined BSC 3270 terminals in other domains that can be used for cross-domain communication; each minor node represents an application program, logical unit, or specially defined BSC 3270 terminal in another domain. Each cross-domain resource major node is defined with a `VBUILD` definition statement. Each cross-domain resource minor node is defined with a `CDRSC` definition statement.

Defining Cross-Domain Path Tables

The user can define one or more cross-domain path tables. A path table identifies to ACF/VTAM the channel-attached communications controller within ACF/VTAM's domain to which a message is sent when its destination is not in ACF/VTAM's domain. A `PATH` definition statement identifies the subarea associated with the communications controller's NCP and a list of subareas in other domains whose messages are to be routed through the communications controller. The subarea of the communications controller's NCP is known as the *adjacent subarea* (`ADJSUB`), and the subareas in other domains for which messages are to be routed through the controller are called *destination subareas* (`DESTSUB`).

Figure 3-5 shows a multiple-domain configuration, the desired paths, and the ACF/VTAM `PATH` statements needed to define the paths. The `PATH` statements provided in each domain constitute the path table for that domain. The reader should be aware that, in addition to the ACF/VTAM `PATH` statements, `PATH` statements must also be provided for the NCPs in the NCP definition decks; for information on the NCP statements, see the *NCP Generation* manual.

For information on modifying a path table, see "Activating Path Tables" in Chapter 4.

Defining Connection Procedures

Once a terminal is activated, a request for connection to an ACF/VTAM application program can be made by:

The terminal (a logon)

ACF/VTAM (an automatic logon)

An ACF/VTAM application program (a passed connection or acquisition, which can be a simulated logon or an acquired connection)

The network operator (`VARY LOGON` command)

A request for connection between an ACF/VTAM application program and a terminal usually takes place in two stages:

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

ACF/VTAM passes on a request for connection to an application program either by scheduling the program's `LOGON` exit routine, if one has been designated, or by completing an `OPNDST` macro instruction that specifies `OPTCD=ACCEPT`. When the `LOGON` exit routine is scheduled, the program can issue an `OPNDST` macro instruction that specifies `OPTCD=ACCEPT` to accept the connection request or it can issue a `CLSDST` macro instruction to reject the request. When the application program learns of

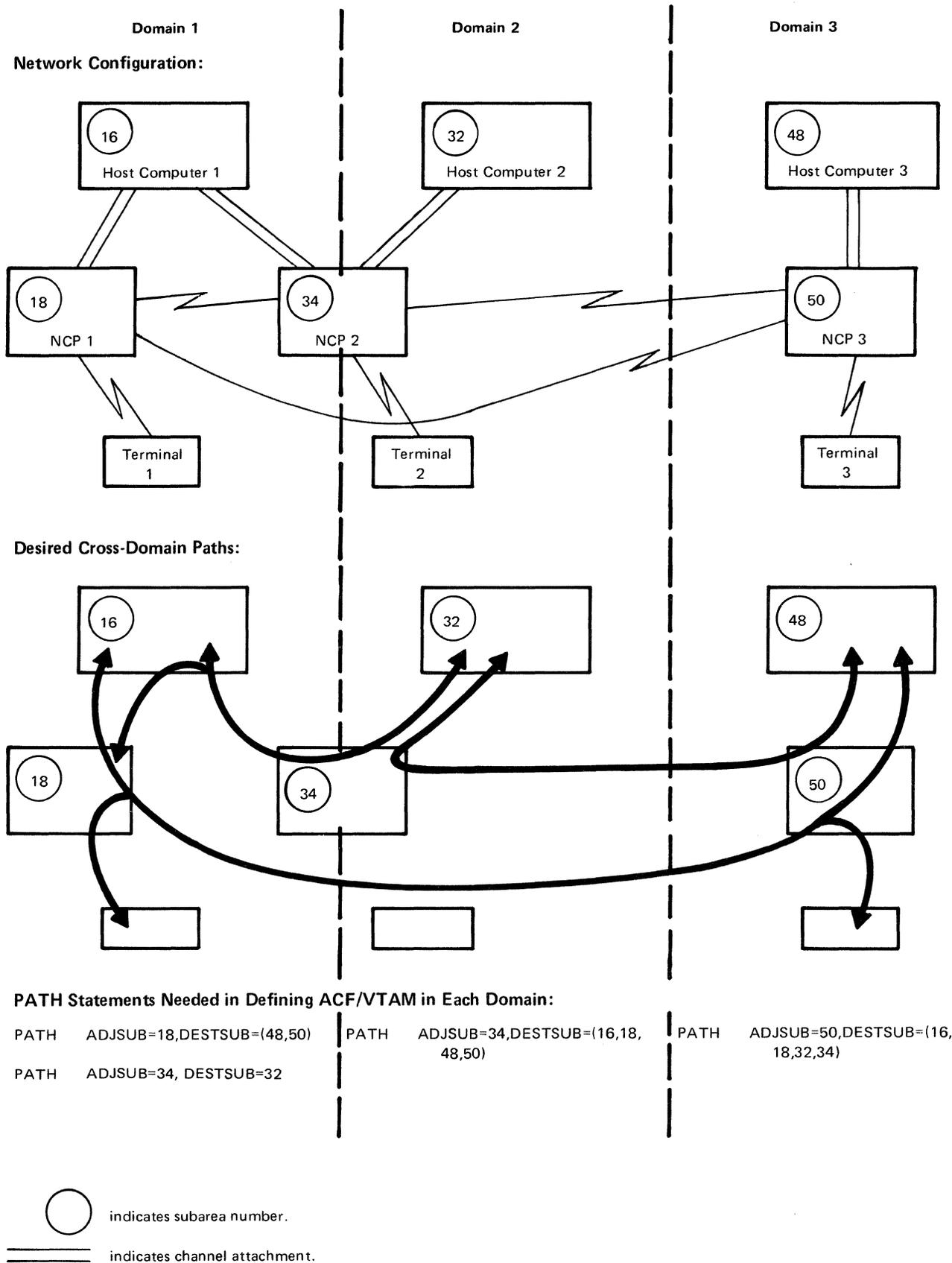


Figure 3-5. ACF/VTAM Path Tables for Cross-Domain Paths

the connection request by having a pending OPNDST OPTCD=ACCEPT completed, the connection request will already have been accepted, but the program can issue a CLSDST macro instruction to cause disconnection. Chapter 5, "Writing an ACF/VTAM Application Program," discusses how an ACF/VTAM application program connects terminals.

Defining Terminal-Initiated Connection

A user can allow a terminal to request connection to an ACF/VTAM application program. Defining this kind of connection requires defining logon formats to ACF/VTAM, the ACF/VTAM application program, and the terminal. This section describes defining these procedures for logical units and BSC 3270 terminals that are defined by the user as using character-coded logons; see Chapter 8 for a description of defining these procedures for non-SNA terminals.

Defining Secondary Application Program Connection

Connection procedures for application programs acting as secondary ends of sessions require no action when defining the ACF/VTAM system, and the procedures are described in Chapter 5.

Types of Logons

ACF/VTAM receives a logon as either a field-formatted Initiate command or a character-coded logon. The logon contains the name of the application program with which connection is requested, an indication (called a *logon mode*) of the set of communication rules (or protocols) to be used during the session, and data that can be passed to the ACF/VTAM application program (called a *user logon message*). With some SNA terminals, a user may not need to know the exact format of a logon, because the terminal, the IBM-supplied terminal system program, or ACF/VTAM formats the message appropriately.

In general, logical units associated with programmable controllers and control units can send field-formatted logons; some can also send character-coded logons. In general, BSC 3270 terminals and logical units associated with nonprogrammable SNA terminals send character-coded logons. See the programming publications for the particular terminal product to determine whether to plan for field-formatted or character-coded logons or both.

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 ACF/VTAM into a field-formatted logon.

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

Defining Field-Formatted Logons

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

In general, field-formatted logons are sent by programmable SNA terminals and secondary application programs.

Programmable terminals can send field-formatted logons because the logic of the terminal program can be written to generate those logons in the proper format. The program in a programmable terminal may generate the field-formatted logon on its own initiative (perhaps because it has reached a point in its processing where it needs to communicate), or the program may generate the logon because of some stimulus from a terminal connected to the program (such as an attention signal from the terminal or receipt of a logon from the terminal in a user-defined format). Regardless of what causes the terminal program to produce the logon, the field-formatted logon must contain the name of the application program with which the terminal wants to be connected. The name of the application program may have been available when the terminal program was written and therefore be hard-coded into the program, or the name of the application program may have been provided by the terminal operator in his or her user-defined logon.

When a secondary application program requests connection, ACF/VTAM converts the request into a field-formatted logon and passes the logon to the primary application program identified in the connection request.

To inform ACF/VTAM that logons from a particular logical unit will be field-formatted, the SSCPFM=FSS operand is included in the LU definition statement for the logical unit. Figure 3-6 shows a field-formatted logon.

Defining Character-Coded Logons

Character-coded logons are used primarily by terminals that are not programmable. The terminal operator directly keys in a request for connection to a specific ACF/VTAM application program, indicating a logon mode (such as interactive or batch) and specifying data (a user logon message) to be passed to the ACF/VTAM application program. ACF/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 his or her own. To specify that logons will be character-coded, each LU statement should specify SSCPFM=USSSCS or SSCPFM=USS3270 (depending on the type of terminal) when defining the network to ACF/VTAM. Only certain 3270 terminals can use USS character-coded logons, depending on the type of terminal and on the ACF/VTAM definition statements that were coded. Of those that can use character-coded logons, the use of either SSCPFM=USSSCS or SSCPFM=USS3270 depends on the type of terminal. For guidance on specification of these parameters for 3270 terminals, see the *ACF/VTAM System Programmer's Guide* for the operating system being used at the installation or see the 3270 programming manuals. Figure 3-7 shows a character-coded logon.

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

For terminals that send character-coded logons, IBM defines logon formats and supplies a related definition table that ACF/VTAM uses to convert them to field-formatted logons. 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)
```

The logon mode is a set of session parameters being suggested for use during the session. The set of session parameters are in a table (called logon mode table) that is available to ACF/VTAM and is logically associated with the logical unit at which the logon originated. The method of defining a logon mode table is described later in this chapter.

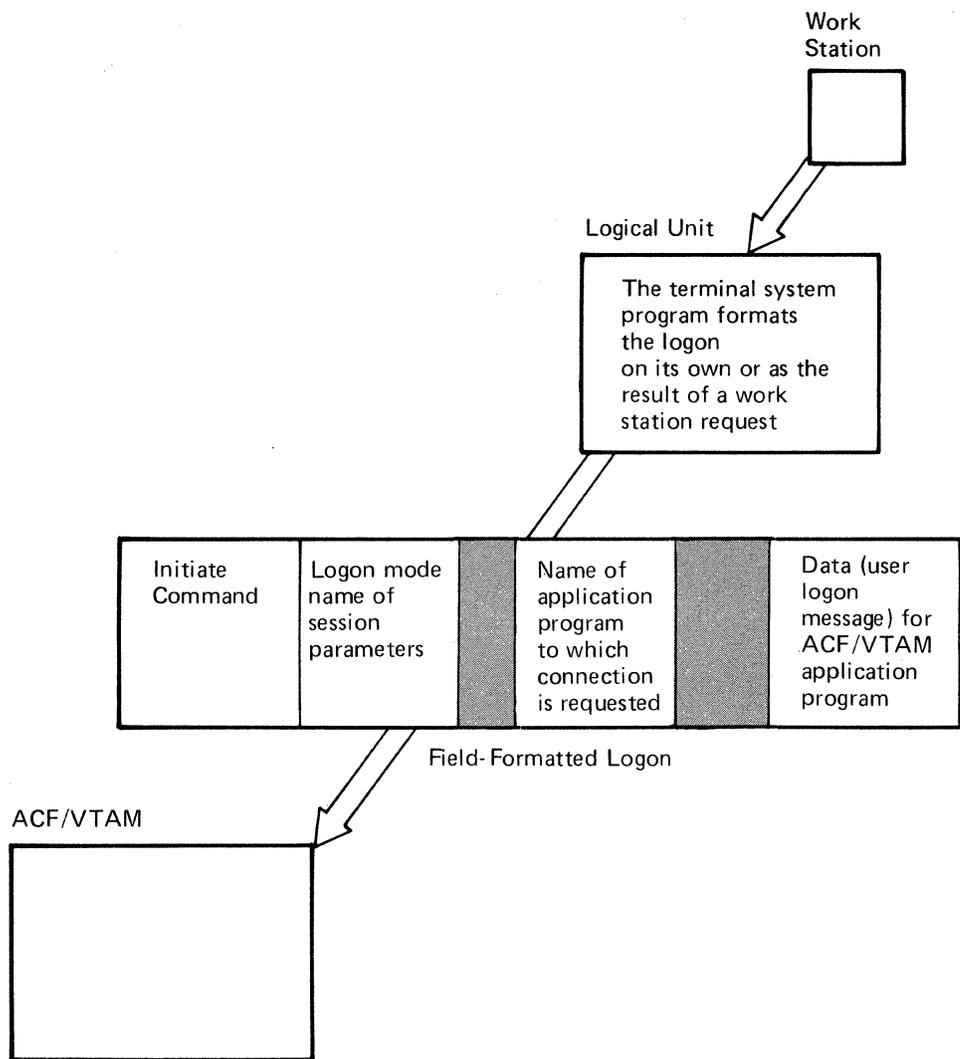


Figure 3-6. A Field-Formatted Logon

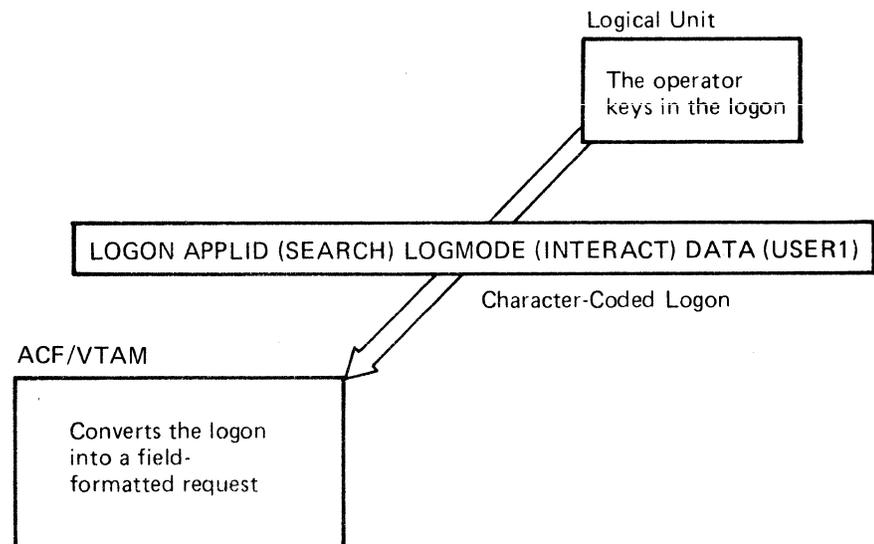


Figure 3-7. A Character-Coded Logon

If the IBM-supplied logon is entered in lowercase, ACF/VTAM translates it to uppercase. If a character-coded logon is incorrectly entered, ACF/VTAM sends an error message to the terminal.

Defining a Character-Coded (USS) Definition Table

Instead of using the logon format and definition table supplied by IBM, a user can define a different logon format and create a definition table to be used to convert character-coded logons in that format to field-formatted logons. This allows a user to select:

A character translation table other than the table supplied by IBM. For example, the user's table can translate numeric characters to alphabetic characters.

An assembler language (BAL) format rather than a PL/I format. For example, the terminal operator can enter:

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

or

```
LOGON application program name, logon mode, logon message
```

Replacements for the IBM-supplied command names or parameter keywords.

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

Alternate text for USS error messages or text for a successful processing message.

To indicate that a different logon format will be used and to specify the definition table to be used in reformatting the logons, the user specifies the name of the definition table in the USSTAB operand of the LU statement when the network is defined to ACF/VTAM. The USS definition table is created by following this procedure (see Figure 3-8):

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 (PL/I 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. Using the USSMSG macro instruction, the user can also define a message to be sent to the terminal when a command (verb) is successfully processed.
5. Code a USSEND macro instruction to indicate the end of the definition table.
6. Assemble, link edit, and load the resulting module into the appropriate library.

Figure 3-9 shows an example of how a USS definition table is used to translate a terminal operator's logon.

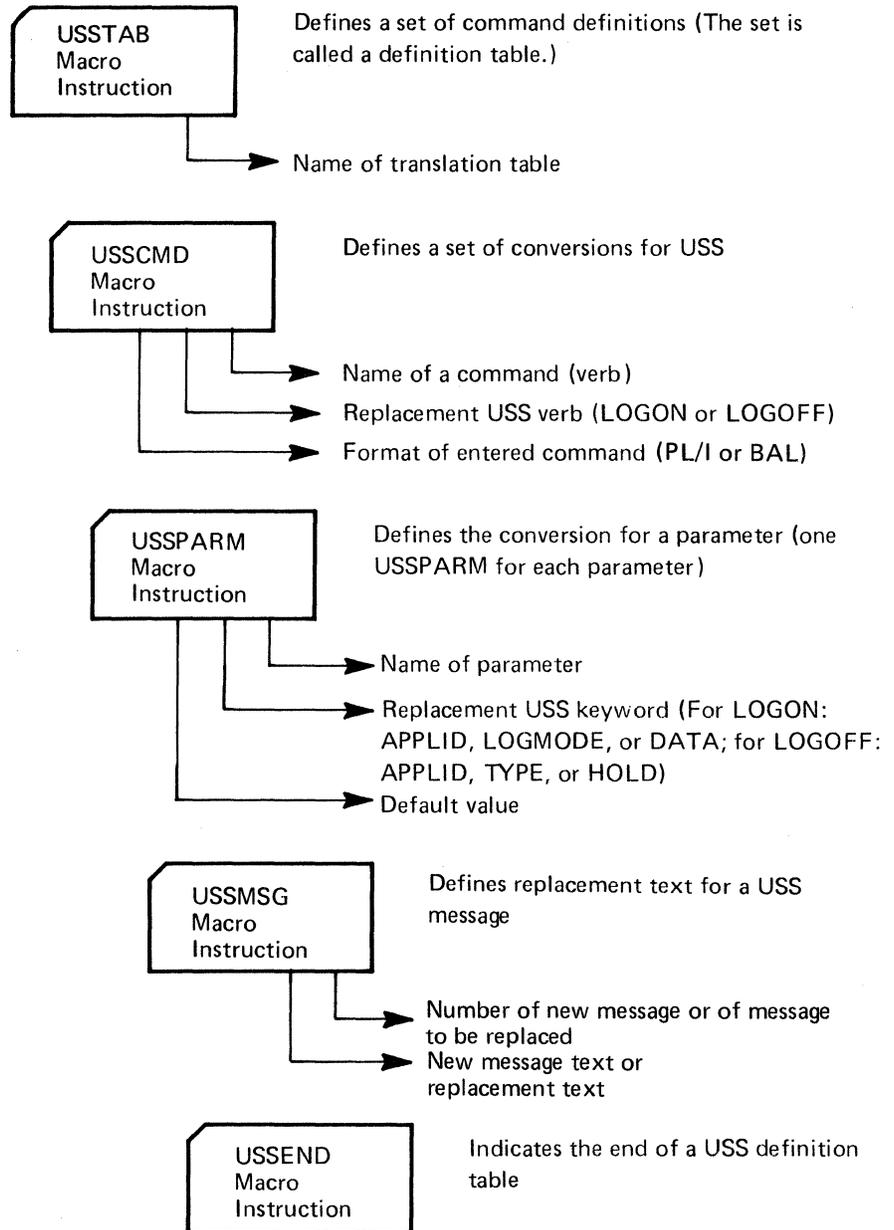


Figure 3-8. Defining a USS Definition Table

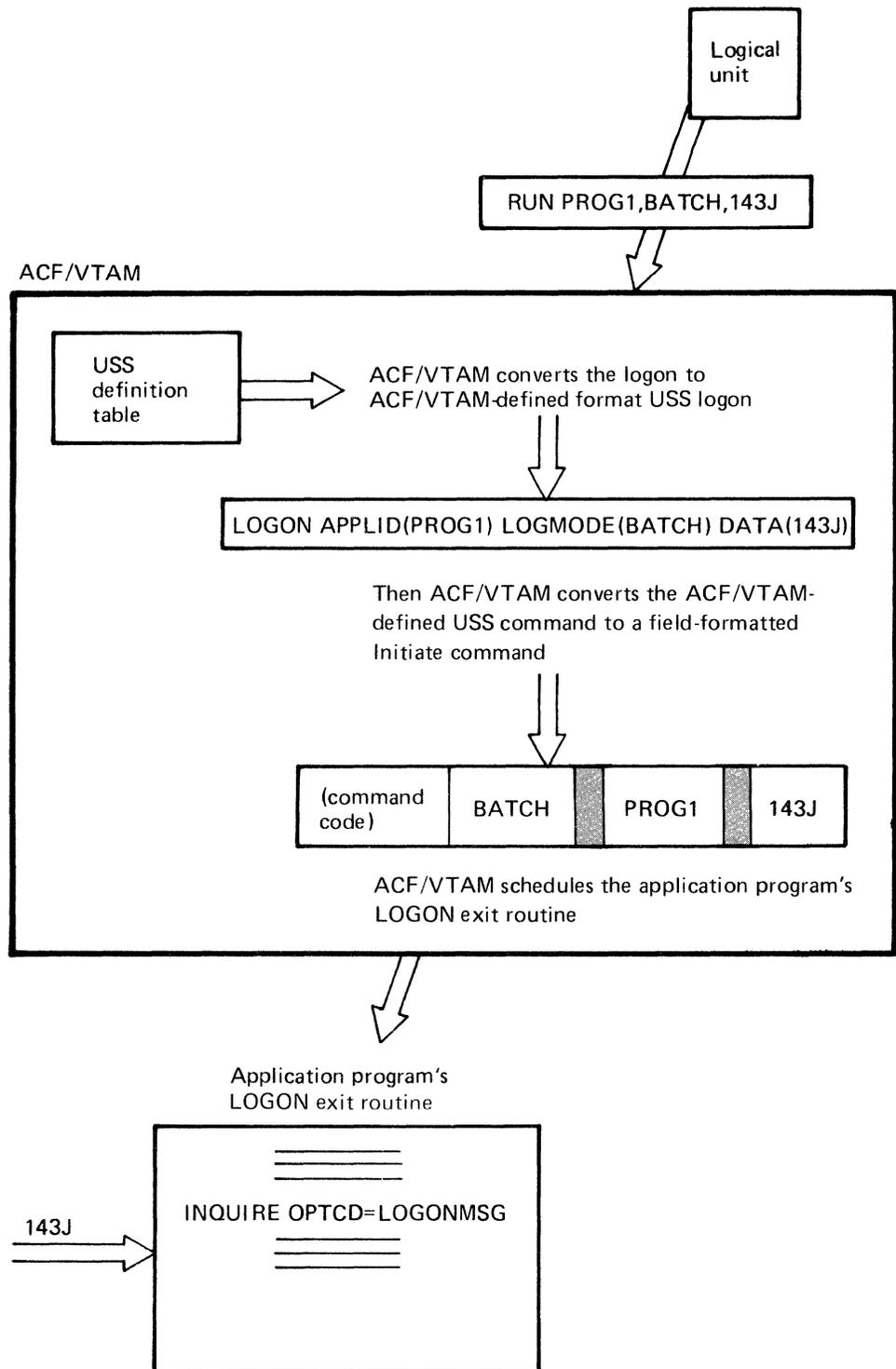


Figure 3-9. Example of ACF/VTAM's Handling of a Character-Coded (USS) Logon

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 name of the application program 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 ACF/VTAM. The interpret table is designed primarily for use in connection procedures for non-SNA terminals (local non-SNA 3270, BSC, and start-stop terminals). The table is described in Chapter 8 under “Specifying Interpret Tables.”

Defining ACF/VTAM-Initiated Connection (Automatic Logon)

A user can specify that a logon for a selected application program is to be automatically generated on behalf of a terminal (a logical unit or a terminal but not a secondary application program) whenever that terminal is active and not already connected to or queued for connection to another program. This specification is made by coding the name of the application 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. (Non-SNA 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 information on the network solicitor.) When considering use of ACF/VTAM-initiated connections, the user should note that a user logon message cannot be made a part of a VTAM-initiated logon.

For logical units that can call in on switched lines, ACF/VTAM holds pending an automatic logon designation for an activated logical unit until a dial-in operation for the logical unit occurs. ACF/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 by issuing a VARY LOGON command. See “Defining Network-Operator-Initiated Connection” below for a description of this facility.

Defining ACF/VTAM Application-Program-Initiated Connection

An ACF/VTAM application program acting as the primary end of a session can initiate a request for connection of a terminal (a logical unit, secondary application program, or a non-SNA terminal) to itself or to another application program. To request that a terminal be connected to itself, an ACF/VTAM application program can:

Issue an OPNDST macro with OPTCD=ACQUIRE to directly acquire the terminal. (This technique is called *acquisition of a terminal*.)

Issue a SIMLOGON macro instruction, which causes ACF/VTAM to create a logon and pass it back to the application-program as though it had come from the terminal. (In this technique, the logon is called a *simulated logon*.)

An ACF/VTAM application program can also request that a terminal currently connected to it be passed to another active ACF/VTAM application program. This is done by issuing a CLSDST macro instruction that specifies OPTCD=PASS and, at the same time, supplying the name of the program to which the terminal is to be passed. Optionally, a logon message can also be passed. As a result of this CLSDST, ACF/VTAM completes an outstanding OPNDST for that terminal or, if there is no outstanding OPNDST, schedules the receiving program’s LOGON exit routine.

Considerations for using these forms of application program-initiated connection are discussed below. For a description of the process by which a secondary application program is connected, see “Secondary Application Program Connection” in Chapter 5.

Acquiring Connection (OPNDST with OPTCD=ACQUIRE)

This form of connection might be useful when:

The ACF/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 an ACF/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 ACF/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 user logon message. (Logons automatically initiated by ACF/VTAM cannot specify a user logon message.)

It is uncertain whether or not a terminal will be available for connection. The SIMLOGON macro instruction can request that the logon 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 ACF/VTAM.

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

This form of connection might be useful when a set of ACF/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 ACF/VTAM.

Defining Network-Operator-Initiated Connection

The user can plan for a network operator to initiate connection for a terminal (a logical unit or non-SNA terminal but not a secondary application program) 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 ACF/VTAM application program. The network operator can initiate a connection request with the VARY LOGON command, specifying the names of the terminal and the application program and, for SNA terminals, the logon mode.

The VARY LOGON command modifies any automatic logon designation that already exists for the terminal. Once the command is issued, the terminal specified in that command is automatically logged on to the application program named in the command whenever it is available unless that program has specifically released it. The new automatic logon specification remains in effect until (1) the network operator issues another VARY LOGON 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 ACF/VTAM network was defined are in effect.

If the network operator initiates the connection request, a user logon message cannot be specified.

Defining Logon Modes (Session Parameters)

As part of the connection process for a record-mode session, both ends of the proposed session must agree on the communication rules to be followed during the session. Those communication rules, called *session parameters*, enable each end of the session to know what the other end will do in different communication situations. The session parameters are a string of bit-settings that indicate the rules to be followed in the session for such things as message and response sequences, resolution of contention, use of message chaining, use of brackets, and use of change-direction indicators. By agreeing on session parameters, each end of the session is telling the other end how it will operate and what the other end can expect.

The process of agreement on session parameters follows this pattern: When a request for a session begins with a logon, the logon is accompanied by a logon mode name (which identifies a particular set of session parameters). The session parameters that are indicated by a logon are being *suggested* by or on behalf of the secondary end of the proposed session. The primary end of the proposed session can decide either to use the suggested set of parameters or to use a different set. The set chosen by the primary end is sent to the secondary end in the Bind command (an internal SNA command that results from issuance of the OPNDST macro instruction in an ACF/VTAM application program). When the secondary end receives the session parameters, it can either accept them or reject them. If it rejects them, the session is not established.

In many cases, the ends of the session work with predefined sets of session parameters. When a set is defined, a name is associated with the set. That name is known as the *logon mode name*. Several sets of session parameters, each with its own name, can be grouped into a table known as a *logon mode table*. The table itself is identified by a *logon mode table name*, which is the name specified in the linkage-editor NAME statement when the table is link-edited. (In lieu of using a predefined set of session parameters, a primary application program can build a set of session parameters when it is needed, and it can have that set sent in the Bind command.)

Tables That Contain Session Parameters

In each operating system is an IBM-supplied logon mode table, with predefined sets of session parameters and a predefined logon mode name for each set. This table is known as the *default logon mode table*. If the IBM-supplied table meets user needs, additional logon mode tables need not be defined. (The user can also replace the IBM-supplied default table by filing a user-defined table under the same name.)

The user can define one or more additional logon mode tables by using the MODETAB, MODEENT, and MODEEND macro instructions (see Figure 3-10). These macro instructions are described in detail in the ACF/VTAM system programmer's guide for the operating system being used. After a table has been coded, it is assembled and link-edited into the appropriate library, depending on the operating system being used at the installation.

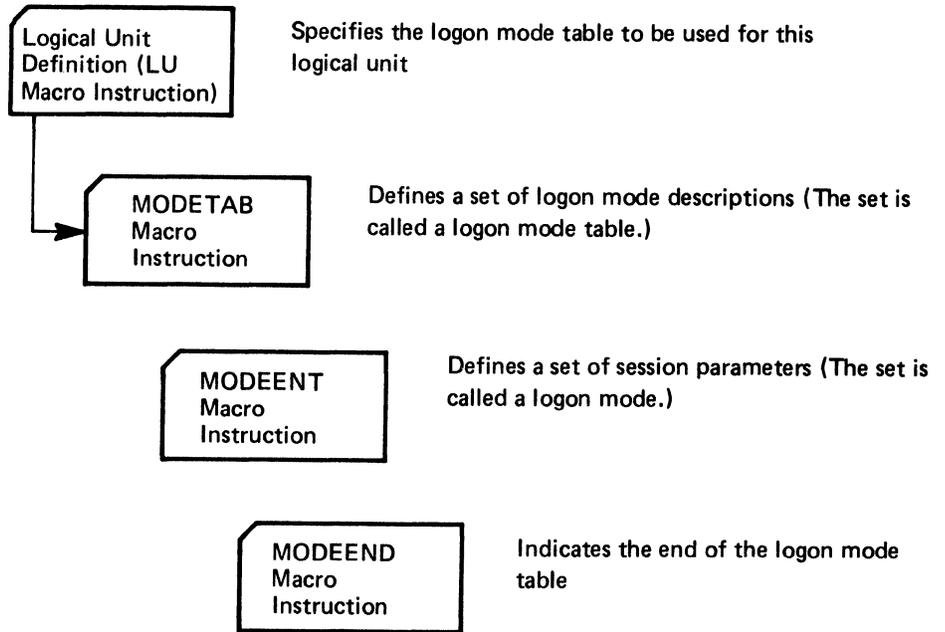


Figure 3-10. Defining a Logon Mode Table

Processing of Session Parameters

During ACF/VTAM definition, a logon mode table is associated with each logical unit (a device-type logical unit or a secondary application program). If the user supplies a logon mode table name in the **MODETAB** operand of the **LU** statement (or a higher-level definition statement, such as a **PU** statement) or **APPL** statement, the table by that name is associated with the logical unit. If the user does not code the **MODETAB** operand, the default logon mode table is associated with the logical unit. The table associated with a logical unit is the first one searched for a logon mode name when such a name is specified for the logical unit during the connection process. The table is also the first one searched for a default entry when no logon mode name is supplied (the default entry is the entry specified by the **DLOGMOD** operand in the logical unit's definition statement, or in the absence of a **DLOGMOD** specification, the first entry in the table). If the logon mode name is not found in the first table searched, the default logon mode table is searched. Note that the logon mode table for a logical unit must be in the appropriate library *in the same domain* as the secondary logical unit with which it is associated.

When the application program starts processing the logon, the session parameters associated with the logon mode name in the original logon (or the default set) are available for inspection by the application program. (Only the actual session parameters—not the logon mode name—are available to the application program.) The program gets the session parameters by issuing the **INQUIRE** macro instruction with **OPTCD=SESSPARM**. The application program can modify the session parameters, using the **ISTDBIND DSECT** to construct a set of session parameters in a portion of its own storage known as the *bind area*.

When the application program issues the **OPNDST** macro instruction with **OPTCD=ACCEPT**, the program designates which set of session parameters is to be included in the **Bind** command. The set may be:

- The session parameters associated with the logon

- A named set of session parameters from the logon mode table associated with the secondary logical unit (if the logical unit is in the same domain as the program issuing the **OPNDST ACCEPT** macro instruction)

The default set of session parameters from the logon mode table associated with the secondary logical unit (if the program is in the same domain as the program issuing the OPNDST ACCEPT macro instruction)

The session parameters that the program built in its bind area

Detailed information on the bits and fields in the session parameters is provided in Appendix J of *ACF/VTAM Macro Language Reference*. More information on the use of session parameters and the designation of parameters by the primary and secondary ends of the session is provided in *ACF/VTAM Macro Language Guide*.

Defining Disconnection Procedures

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

The terminal

The ACF/VTAM application program

The network operator

ACF/VTAM

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

Disconnection Requested by the Terminal

For logical units, ACF/VTAM provides a logoff facility similar to the logon facility described previously in this chapter under "Defining Connection Procedures." A logical unit can send either a field-formatted or character-coded logoff. The field-formatted logoff is a Terminate command (an internal SNA command); the character-coded logoff is a character string that ACF/VTAM converts into a Terminate command. Consult the programming publications for each terminal product to determine the kind of logoff used (field-formatted or character-coded or either). A secondary application program issues a macro instruction to send a logoff to a primary application program. No action is required when defining the ACF/VTAM system to prepare for this type of logoff.

For non-SNA terminals, the ACF/VTAM application program may search each data message from the terminal for a predesignated character or character-string indicating a request for disconnection. ACF/VTAM does not provide any special facility for logoff from non-SNA terminals. Although ACF/VTAM allows BSC 3270 terminals to use character-coded (USS) logons, it does not allow them to use character-coded (USS) logoffs. When the application program receives a logoff from a non-SNA terminal, it may issue a CLSDST macro instruction, causing ACF/VTAM to disconnect the terminal from the program.

If a character-coded logoff is used, the user can either use the IBM-defined logoff format, or a logoff format can be defined by means of 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 command or in a character-coded logoff defined by IBM or the user, specify:

The ACF/VTAM application program from which the logical unit is to be disconnected

Whether disconnection is mandatory (UNCOND) or at the discretion of the primary 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

As a result of a logoff, the primary application program from which disconnection is requested will get control. The action taken by the application program depends on the conditional or unconditional parameter.

If the logoff specifies conditional termination, ACF/VTAM schedules the application program's LOSTERM exit routine. The program can then ignore the logoff or request disconnection by issuing a CLSDST macro instruction. This kind of logoff allows the application program to exchange some final messages before ending the session. If the logoff specifies unconditional termination or omits the parameter, ACF/VTAM stops all communication, disconnects the terminal, and then schedules the program's LOSTERM exit routine with an indication that the application program must issue a CLSDST macro instruction (the macro instruction must be issued even though the terminal is already disconnected).

Holding the Physical Unit's Physical Connection

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

When defining the physical unit to ACF/VTAM, the user specifies whether the physical unit is to be physically disconnected when the last of its logical units ends a session (DISCNT=YES) or whether the question of physical disconnection is to be decided by parameters provided by the logical units as they log off (DISCNT=NO).

When DISCNT=YES is specified for the physical unit in its definition statement, the physical unit is physically disconnected when its last logical unit ends its session, regardless of any disconnection parameters provided by the logical units in their logoffs.

When DISCNT=NO is specified for the physical unit in its definition statement, each logoff request for a logical unit can specify whether or not the physical unit's physical connection is to be held. The specification consists of a Last or Not Last indicator in a field-formatted Terminate command or a HOLD NO or HOLD YES indication in a character-coded logoff. (Not Last or HOLD YES indicates that the physical unit's physical connection should be held.) When the last logical unit associated with a physical unit logs off, the physical unit is physically disconnected if: (1) at least one logical unit ended its last session with a Last or HOLD NO indication and (2) none of the logical units ended its last session with a Not Last or HOLD YES indication. If any logical unit ended its last session with a Not Last or HOLD YES indication, the physical unit remains connected. (A session between an application program and a terminal can be ended purely at the initiative of the application program; that is, the

application program can issue the CLSDST macro instruction without receiving a logoff. If all sessions are ended with unsolicited CLSDST macro instructions, the physical unit remains connected. Similarly, if all logical units are deactivated with the VARY INACT command, the physical unit remains physically connected. Thus, it is possible for all logical units to be disconnected without a logoff, and since no logical unit ended its session with a Last or HOLD NO indication, the physical unit connection is retained.)

When DISCNT=NO is specified for the physical unit in its definition statement and ACF/VTAM receives a Request Disconnect command (an internal SNA command) from the physical unit, the physical unit is physically disconnected regardless of the holding parameters specified by the logical units. A Request Disconnect Immediate command causes ACF/VTAM to physically disconnect the physical unit immediately, even if logical units are in session. A Request Disconnect Normal command causes ACF/VTAM to physically disconnect the physical unit as soon as the last logical unit ends its session, even if one of the logical units ended its last session with a Not Last or HOLD YES indication. Receipt of the normal version of the command also causes physical disconnection of the physical unit when all logical units have ended their sessions but the physical unit's physical connection is being held because of a Not Last or HOLD YES indication. The Request Disconnect command can result from timing logic in the program associated with a terminal, or it can result from action by an operator at the terminal or cluster control unit represented by the physical unit.

The user may want to hold the physical connection to a physical unit when logical unit connections to application programs are expected to be frequent but of short duration, thus saving the processing time that would otherwise be expended by ACF/VTAM in frequently physically disconnecting and reconnecting the physical unit. For physical units on switched lines, holding the physical unit's physical connection allows control over the frequency of dial-in or dial-out operations for the physical unit.

Physical disconnection of the physical unit from ACF/VTAM is different for physical units on nonswitched lines than for physical units that are on switched lines or that are locally attached. When a physical unit on a nonswitched line is physically disconnected from ACF/VTAM, it is also deactivated. When a physical unit that is on a switched line or that is locally attached is physically disconnected, it remains active.

Using the Request Shutdown Protocol

A logical unit or a secondary application program can also use a Request Shutdown command to request session disconnection. This command advises the primary application program that it should issue a CLSDST macro instruction to end the session as soon as possible.

Disconnection Requested by a Primary Application Program

A primary application program can decide that communication with a connected terminal is finished and that the terminal is to be disconnected. This might be the normal procedure for disconnection at the end of batch output to a terminal. The application program issues a CLSDST macro instruction, causing ACF/VTAM to logically disconnect the program from the terminal. OPTCD=RELEASE can be specified to allow the terminal to be available for connection to other programs. OPTCD=PASS can be specified to notify a specified application program that the terminal is available for connection. The user must authorize an ACF/VTAM application program to use OPTCD=PASS when the network is defined to ACF/VTAM.

Disconnection Requested by a Secondary Application Program

In a session between two application programs, the secondary application program can take the initiative in requesting disconnection. The secondary application program can request disconnection either (1) by sending a Request Shutdown (RSHUTD) command to the primary application program or (2) by issuing a TERMSESS macro instruction.

The sending of a Request Shutdown command is a way a secondary application program can send a disconnection request to a primary application program without involving ACF/VTAM in the notification process. The command goes from one application program to the other without being recognized or acted upon by ACF/VTAM. To the primary application program, the command represents a request to disconnect the secondary program when it is convenient for the primary program to do so. This allows the primary program to continue communications with the secondary program in order to do any cleanup operations that are necessary.

An alternative way for the secondary application program to request disconnection is to issue a TERMSESS macro instruction. This macro instruction specifies (in its OPTCD operand) whether the disconnection is to be conditional or unconditional. ACF/VTAM converts the macro instruction into a Terminate command, either conditional or unconditional according to the OPTCD operand in the macro instruction. When the Terminate command reaches the primary end of the session, it causes the primary application program's LOSTERM exit routine to be scheduled.

A conditional TERMSESS macro instruction is similar to a Request Shutdown command in that disconnection of the secondary application program (via the CLSDST macro instruction) is entirely at the discretion of the primary application program. Disconnection can be requested immediately or after cleanup operations.

An unconditional TERMSESS macro instruction causes ACF/VTAM to immediately terminate the session. The primary program must still issue the CLSDST macro instruction, but no further communication with the secondary application program is possible.

Disconnection Requested by the Network Operator

The network operator can deactivate a terminal (or a higher-level major or minor node) *normally* or *immediately*. Normal deactivation does not cause the terminal to be immediately disconnected; it simply indicates to ACF/VTAM the next time the terminal becomes disconnected from an ACF/VTAM application program, it is to become inactive. Immediate deactivation, however, acts as a logoff on behalf of a terminal. The ACF/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, and the ACF/VTAM application program is not aware of the problem.

Disconnection Requested by ACF/VTAM

ACF/VTAM requests that a terminal be disconnected when certain hardware and software errors occur that cause loss of contact with that terminal. ACF/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 program 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, ACF/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 of disconnection in Chapter 5.

Authorization, Accounting, and Logon-Interpret Exit Routines

ACF/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 ACF/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 ACF/VTAM as part of ACF/VTAM definition. These routines are executed as part of ACF/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 ACF/VTAM.

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

Each type of exit routine is described below.

Authorization Exit Routine

ACF/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 ACF/VTAM.) This exit routine is scheduled whenever a terminal is to be queued for logon to, or connected to an application program. Thus, the routine at this exit is executed whenever a connection is to be made as a result of an OPNDST, SIMLOGON, REQSESS, OPNSEC, or CLSDST (with the PASS option) macro instruction in the application program, or whenever a terminal is to be queued as the result of a logon. In a multiple-domain network, the authorization exit routines in both domains are executed for cross-domain connections.

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

The type of connection request that has been made.

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

Using this input, the authorization routine can determine whether each connection or logon should be processed by ACF/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 ACF/VTAM. If the authorization exit routine determines that the request is valid, ACF/VTAM proceeds with making the connection. If the exit routine finds the request is invalid, ACF/VTAM rejects the request. The output from the exit routine, which is passed by setting a register, informs ACF/VTAM of whether the request is valid or invalid.

In DOS/VS, the authorization routine is included in ACF/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 ACF/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 ACF/VTAM's protection key. Therefore, errors within the routine may cause damage to ACF/VTAM's control blocks and modules. Also, security violations can occur because such a routine has access to much of the ACF/VTAM partition and private address space.
- The exit routine is executed under the task for which the request is being authorized or ACF/VTAM's task for cross-domain requests. 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 ACF/VTAM processing. Therefore, performance may be degraded if the routine requires lengthy processing time. While this routine is being executed, no new connection or logon requests are processed by ACF/VTAM, and requests involving ACF/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 ACF/VTAM's network solicitor and the port solicitor (this facility supports start-stop and BSC switched networks) 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 ACF/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

ACF/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.

When the routine is entered, the name of the application 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 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 ACF/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 ACF/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 ACF/VTAM's protection key. Therefore, errors within the routine may cause damage to ACF/VTAM's control blocks and modules. Also, security violations can occur, because such a routine has access to much of the ACF/VTAM partition and private address space.
- The exit routine is executed under the task about which the accounting information is being collected or ACF/VTAM's task for cross-domain requests (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 ACF/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 ACF/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, and the port solicitor (this facility supports start-stop and BSC switched networks). 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 ACF/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 ACF/VTAM Start Options

When ACF/VTAM is started, options can be used to define the initial domain configuration and to select optional ACF/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 ACF/VTAM (in DOS/VS or OS/VS). The options can also be predefined and filed in the ACF/VTAM definition library. (See "Starting ACF/VTAM" in Chapter 4 for information on specifying options from the system operator's console.)

The ACF/VTAM start options tailor ACF/VTAM to the user's needs each time ACF/VTAM 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 ACF/VTAM configuration. With different sets of predefined options, the user can initialize a particular ACF/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 ACF/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 ACF/VTAM trace facility
- Which major nodes and path tables are to be activated during start processing
- Whether the status (active or inactive) of major nodes is to be recorded
- 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
- The size and characteristics of ACF/VTAM storage pools and, in OS/VS, which pageable storage pools should be fixed
- The maximum number of major nodes defined at any one time
- The maximum number of application programs active within the host computer at any one time
- Whether certain network operator messages are to be suppressed
- The host computer's subarea number
- Whether prompting messages should be sent to the network operator to obtain additional start information
- Whether ACF/VTAM is to collect tuning statistics to be used for adjusting start parameters for more efficient operation of ACF/VTAM
- Whether the ACF/VTAM logon monitor facility (referred to as the network solicitor) for local non-SNA 3270, start-stop, and BSC terminals is to be started when ACF/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 as long as the terminals are active or until the network operator stops the trace. See "Starting and Stopping ACF/VTAM Facilities" in Chapter 4 for information on stopping the ACF/VTAM traces. See "Serviceability Aids" in Chapter 6 for a description of ACF/VTAM's trace facility.

Active Major Nodes: To activate major nodes during ACF/VTAM start processing, the network operator can request that ACF/VTAM activate the major nodes listed in either a member of the ACF/VTAM definition library (a cold start) or a VSAM configuration restart data set (a warm start). The member of the ACF/VTAM definition library defines the network's *initial status*; the VSAM data set defines the network's *status prior to deactivation*.

Active Minor Nodes: 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 telecommunication configuration.

Storage Pools: ACF/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 ACF/VTAM Buffering" later in this chapter for more information on specifying storage pools and on ACF/VTAM's use of these pools.

Prompting Message: If the network operator is to be prompted, ACF/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 ACF/VTAM during ACF/VTAM initialization. See "Starting ACF/VTAM" in Chapter 4 for information on the role of the network operator in starting ACF/VTAM.

Defining ACF/VTAM Buffering

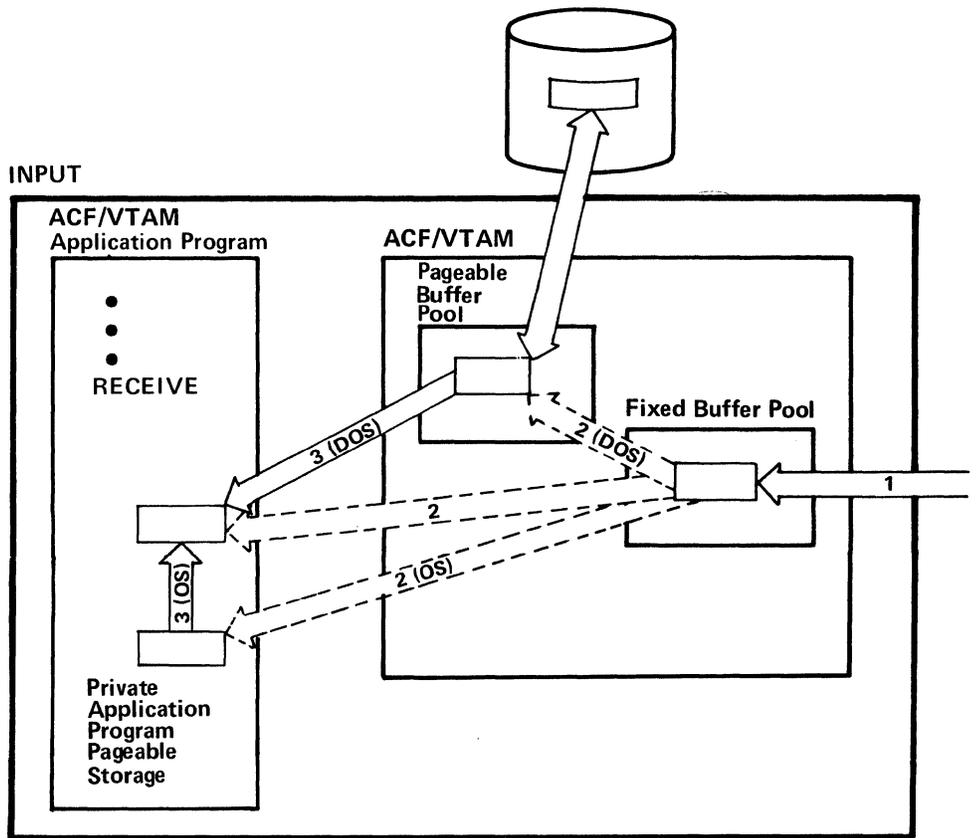
ACF/VTAM can hold data temporarily until an ACF/VTAM application program requests it or until it can be sent to a terminal. Figure 3-11 shows how ACF/VTAM buffers are used in input and output operations.

ACF/VTAM has both fixed and pageable buffer pools for buffering data. In addition, buffer pools are required for the ACF/VTAM control blocks that keep track of incoming data that has not yet been requested by an ACF/VTAM application program and to keep track of individual input and output requests. Each buffer pool serves all active application programs. Because users' needs vary, ACF/VTAM allows the user to specify the size and characteristics of each buffer pool. In OS/VS, the user can specify that one or more of these pools are to be fixed in main storage. The names of the buffer pools are different in each operating system and may be found in the appropriate *ACF/VTAM System Programmer's Guide*.

When ACF/VTAM is started, the user specifies the number of buffers (elements) in each pool, the size of each buffer, the slowdown point for the pool, the expansion point for the pool, and the expansion number for the pool. Both the slowdown point and expansion point are expressed as numbers of buffers in the pool that are not in use. When the number of free buffers falls below the slowdown point, ACF/VTAM limits access to the buffer pool. When the number of free buffers falls below the expansion point, ACF/VTAM adds additional buffers to the pool in increments specified by the expansion number (if dynamic expansion of the pool has been specified).

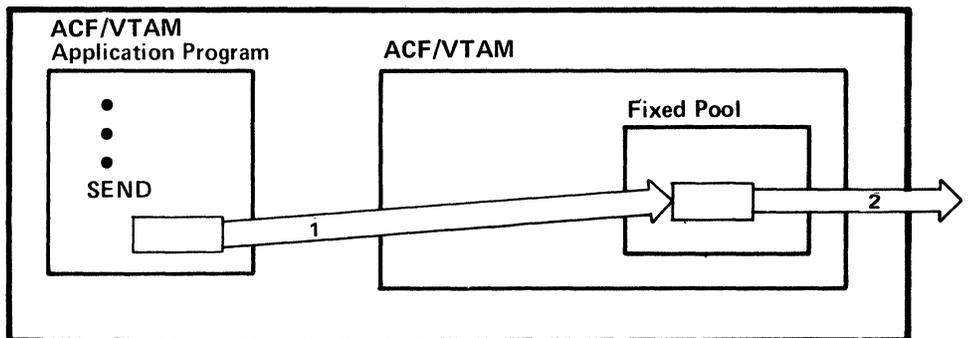
In addition to specifying the size and characteristics of buffer pools when starting ACF/VTAM, the user can control the amount of buffer storage that can be used for data for each basic-mode connection between an application program and a terminal. Controlling buffer storage for basic-mode connections between application programs and terminals prevents an individual connection from monopolizing data buffers and slowing down communication on other connections. The number of data buffers for each basic-mode connection is specified during ACF/VTAM definition. ACF/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, TERMINAL, COMP, or VTERM statement). For incoming data from basic-mode terminals, if the threshold is exceeded, ACF/VTAM issues a RESET unconditional macro instruction, which cancels the I/O operation. To determine whether any data was lost, the application program must check the return codes in the RPL fields for the I/O request. This situation will not occur, however, if the application program requests input at a rate that is not below the rate at which data may be coming into ACF/VTAM.

Input data may have to be buffered for a longer time than output data. For input data, ACF/VTAM must buffer the data until it is requested by an application program. For output data, ACF/VTAM decides when to begin handling the data; after receiving an output request, ACF/VTAM does not move the data from the application program until



- 1 Data arrives from a terminal independent of any ACF/VTAM application program request for data. It is placed in a buffer in the fixed buffer 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 buffer pool to the specified application program data area (shown with dotted lines).
If there is no request yet for the data, it is moved from ACF/VTAM's fixed pool to an ACF/VTAM pageable pool (in DOS) or to the application program's private pageable storage (in OS/VS). 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 application program data area.

OUTPUT



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

Figure 3-11. How ACF/VTAM Uses Input and Output Buffers

it has ensured that it has buffers for it. In general, an application program should be written so that 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 reduces the possibility of exceeding the number of buffers for data and control block storage pools.

In OS/VS, the user can specify that certain buffer pools are to be fixed in main storage. A user who 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 buffer pools are provided in the *ACF/VTAM System Programmer's Guide* for the operating system being used by the installation.

Chapter 4. Controlling an ACF/VTAM System

This chapter describes in general how the user controls ACF/VTAM and its domain, using predefined specifications and network operator commands. The *ACF/VTAM System Programmer's Guide* for the operating system being used at the installation describes predefined specifications in detail, and the appropriate *ACF/VTAM Network Operating Procedures* describes network operator commands in detail.

Levels of Control

A user can control ACF/VTAM by means of the specifications made during ACF/VTAM definition and by using network operator commands. During ACF/VTAM definition, a user defines and tailors an ACF/VTAM system. This defines the operating limitations of the ACF/VTAM system. Planning is required because the definition of the system and the generated NCPs cannot be easily changed. ACF/VTAM network operator facilities allow the network operator¹ to monitor and control an ACF/VTAM domain between the time ACF/VTAM is started and the time it is halted.

In a multiple-domain network, coordination of network operators in different domains may be required to control cross-domain connection and communication; a network operator can enter ACF/VTAM commands to monitor and control only his or her domain. A program operator (an application program written to receive ACF/VTAM messages and issue ACF/VTAM commands) can communicate with a program operator in another domain to request control across domain boundaries. For a description of a program operator, see "Network Operator Control from an Authorized Application Program" later in this chapter. Alternatively, the IBM program product Network Operations Support Program (NOSP) can be used to fulfill this function. For information on NOSP, see the *NOSP General Information* manual, GC38-0251.

ACF/VTAM Network Operator Commands

ACF/VTAM's *network operator commands* enable the network operator to monitor and control his or her domain. ACF/VTAM commands are a subset of the operating system commands. Incorrect commands are rejected by ACF/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 ACF/VTAM application program) can:

- Monitor the data communication system
- Activate and deactivate nodes
- Initiate requests for connection
- Start and stop selected ACF/VTAM facilities
- Change line-scheduling specifications for start-stop and BSC lines

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

¹In this manual, *network operator* refers to either the person at a system console or an authorized application program (*program operator*), except when referring to starting ACF/VTAM (with the START command) or stopping ACF/VTAM (with the HALT command), which can only be done from a system console.

Starting ACF/VTAM

ACF/VTAM is started by issuing the `START` command from the system console. This command causes ACF/VTAM and its domain to be initialized. Starting ACF/VTAM can include the activation of some or all of the nodes. It can also include the activation of selected ACF/VTAM facilities.

At the time ACF/VTAM is started, the user tailors the data communication system either by entering ACF/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 (for OS/VS) or books (for DOS/VS) of the ACF/VTAM definition library. See "Defining ACF/VTAM Start Options" in Chapter 3 for information on predefining ACF/VTAM start options in `ATCSTRxx` data sets.

The ACF/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` data set. The options that can be entered through the console include:

- An identification number (the SSCP identification number) that a physical unit can use to identify the host computer

- The host computer's subarea number

- Whether ACF/VTAM's network solicitor is to be activated

- Whether ACF/VTAM's trace facility is to be activated and, if activated, for which nodes

- Which major nodes and path tables 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 recorded in their configuration restart data set

- The name of a VSAM data set on which ACF/VTAM records configuration restart data

- Sizes and characteristics of ACF/VTAM buffer pools

- Whether the ACF/VTAM storage management trace facility is to be activated

- Whether certain messages are to be suppressed

- Whether ACF/VTAM is to collect data that can later be used to adjust start options for more efficient operation (called *tuning statistics*)

- The highest subarea number assigned to a subarea in the network. In a multiple-domain network, the number must be the same in all domains.

- The number of application programs that are expected to be active in the host computer at one time

- 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 ACF/VTAM differs slightly between DOS/VS and OS/VS. Both are discussed below. One difference is that, in OS/VS, the network operator can specify start options in the `START` command; the operator cannot do this in DOS/VS. Because of this, the hierarchy by which start options from different sources override each other differs slightly in OS/VS and DOS/VS. For the exact override hierarchy, see the *ACF/VTAM System Programmer's Guide* for the operating system being used at the installation.

Starting ACF/VTAM in DOS/VS

ACF/VTAM is started in DOS/VS by first starting the partition in which ACF/VTAM is to be executed and then invoking the ACF/VTAM cataloged procedure. No ACF/VTAM start options can be entered from the console until ACF/VTAM requests them with a prompting message.

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

Starting ACF/VTAM in OS/VS

The START command starts ACF/VTAM in OS/VS. This command can be entered with or without any of the ACF/VTAM start options described previously under “Starting ACF/VTAM.”

When the network operator starts ACF/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, ACF/VTAM prompts the network operator for start options. If any errors are encountered in the start processing, ACF/VTAM prompts the network operator for final options. These options are the last to be processed before ACF/VTAM completes initialization.

Halting ACF/VTAM

The ACF/VTAM HALT command can be used for an orderly, quick, or, in OS/VS, cancel shutdown. Orderly shutdown is designed to halt ACF/VTAM under normal, planned conditions. Quick and cancel shutdown are designed for emergency situations in which halting ACF/VTAM takes precedence over loss of terminal connections and loss of data.

Orderly Shutdown

If the network operator specifies an orderly shutdown (that is, enters a HALT command without the QUICK or CANCEL option), ACF/VTAM prevents any new application programs from associating themselves with ACF/VTAM (by opening their ACBs) and prevents application programs from making new connections with terminals. Communication on existing connections, however, is allowed to continue. ACF/VTAM notifies application programs of the pending shutdown by scheduling each program's TPEND exit routine if it has one. (The TPEND exit routine is an application program exit routine that halts the application program's communications when ACF/VTAM is terminating.) ACF/VTAM allows normal operations to continue until all application programs have closed their ACBs. Then ACF/VTAM deactivates all nodes and closes down the domain.

For ACF/VTAM's orderly shutdown, 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 disassociate the program from ACF/VTAM (close its ACB). If an application program does not have a TPEND exit routine, it is not notified of the pending shutdown. This delays the halt until either the application program closes its ACB or the network operator cancels the application program.

Quick Shutdown

If the network operator specifies a quick shutdown (that is, enters a HALT command with the QUICK option), ACF/VTAM prevents any new application programs from associating themselves with ACF/VTAM and prevents application programs from making

new connections with terminals. In addition, ACF/VTAM prohibits any further communication between terminals and application programs. No additional input or output requests are accepted, although data already read into ACF/VTAM buffers from non-SNA terminals can be read by application programs. ACF/VTAM also notifies application programs of the pending shutdown by scheduling each program's TPEND exit routine. Having scheduled each exit routine, ACF/VTAM waits until all application programs have closed their ACBs before it closes down the domain.

So that the quick shutdown functions properly, the user should ensure that each application program has a TPEND exit routine. Because a quick shutdown is probably indicative of an emergency situation, the exit routine should disconnect terminals as rapidly as possible and give control to the main part of the application program, which should close the ACB.

Cancel Shutdown (OS/VS only)

In OS/VS, if the network operator specifies a cancel shutdown (that is, enters a HALT command with the CANCEL option) ACF/VTAM notifies application programs 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 ACF/VTAM becomes immediately inactive. Local devices and storage used by ACF/VTAM are released. Cancel shutdown is intended for emergency use only. The ability to cancel ACF/VTAM using the HALT command is not available in DOS/VS, but the network operator can cancel ACF/VTAM using the DOS/VS CANCEL command.

Monitoring ACF/VTAM Status with the DISPLAY Command

The network operator monitors the ACF/VTAM system by requesting status information for nodes in the system. ACF/VTAM's DISPLAY command enables the network operator to request status information and to verify changes resulting from previous operator requests.

The DISPLAY command can be used to request a display of the status of a particular resource by including the name of the resource in the command. The resource can be:

- An application program
- A terminal (logical unit or non-SNA terminal)
- A communication line
- A BSC or start-stop switched-line port. (A status display for a port is the same as for a communication line.)
- A physical unit or a BSC cluster control unit
- A network control program (NCP)
- A local non-SNA major node
- A local SNA major node
- A switched SNA major node
- The current path table
- A cross-domain resource
- A cross-domain resource major node
- A cross-domain resource manager
- A cross-domain resource manager major node

The **DISPLAY** command can also be used to get a display of the status of all nodes of a particular type or all nodes in a particular category. One of these displays will show the status of one of the following:

- All major nodes
- All terminals (logical units and non-SNA terminals)
- All application programs
- All physical units and BSC cluster control units
- All communication lines
- All ACF/VTAM buffer pools
- All active cross-domain resource major nodes in a domain
- All active cross-domain resource manager major nodes in a domain
- All nodes with I/O in progress for session establishment or session termination

To display the status of a particular minor node or major node, the network operator specifies the symbolic name of the node in the **DISPLAY** command. A minor node specified in an **ACF/VTAM DISPLAY** command must be part of an active major node. A major node specified in a **DISPLAY** command must itself be active.

The following paragraphs tell what information is displayed for a given node (minor node or major node).

For an Application Program: The job and step names for the application program (in an OS/VS system); whether the application program is currently connected to ACF/VTAM; optionally the names of terminals connected to the application program; and the names of terminals queued for logon to the application program. Because an open ACB is an application program to ACF/VTAM, a program can be executed in the system but not recognized by ACF/VTAM as an application program if it does not have an open ACB for ACF/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 ACF/VTAM).

For a Terminal: Whether the terminal is active or inactive; whether the terminal has power; the name of the application program (if any) to which the terminal is connected; 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; whether the terminal was acquired from another host computer; 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 non-SNA terminal).

For a Communication Line: Whether the line is active or inactive; whether the line is a cross-domain link; the name of the group to which the line is assigned; optionally the names and status of all nodes assigned to the line; whether the line was acquired from another host computer; 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.

For a Port: The name of the port; whether the port is active, inactive, or not acquired; the name and status of the associated line; whether the line is switched; the name of the line group; the name of the NCP to which the port is subordinate; whether a line trace is in effect; and optionally, the names of start-stop or BSC terminals associated with the port and whether they are active.

For a Physical Unit or BSC Cluster Control Unit: Whether the unit is active or inactive; whether the unit has power; whether the unit was acquired from another host computer; whether a trace is active for the unit; optionally, the names of the 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 an NCP: 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); whether the NCP is shared by more than one host computer (that is, multiple-channel attached); whether the NCP was acquired from another host computer; 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. Optionally, ACF/VTAM displays a list of the lines emanating from the NCP, their status (active, inactive, or not acquired), and an indication of whether they were originally owned by the host computer from which the display is being produced.

For a Cross-Domain Resource: The name of the cross-domain resource; whether the cross-domain resource is ready to begin a session with an application program or terminal; the name of the cross-domain resource manager controlling the cross-domain resource; the number of active sessions for the cross-domain resource; the number of sessions pending for the cross-domain resource; and optionally, the names of all resources in the domain that have active sessions with the cross-domain resource; and the names of all resources in the domain that have sessions pending with the cross-domain resource.

For a Cross-Domain Resource Manager: The name of the cross-domain resource manager; the status of the cross-domain resource manager; whether the cross-domain resource manager is in this domain or another domain; the network address of the cross-domain resource manager; and optionally, additional information about active sessions, pending session requests, associated cross-domain resource managers, and cross-domain resources (the information varies according to operands in the command and according to whether or not the cross-domain resource manager is in this domain or another domain).

For a Local Non-SNA Major Node: The name and status of the major node, and if requested, the names, channel unit addresses, and status of individual terminals that are part of the node.

For a Local SNA Major Node: The name and status of the major node, and if requested, the names and status of logical units associated with the major node.

For a Switched SNA Major Node: The name and status of the major node, and if requested, the names and status of physical units and/or logical units that are part of the node.

For the Current Path Table: Either (1) a list of each remote subarea and its associated adjacent subarea, (2) a list of all remote subareas that can be accessed through a specified adjacent subarea, or (3) the adjacent subarea that is used to access a specified remote subarea.

For a Cross-Domain Resource Major Node: The name of the cross-domain resource, and optionally, the status and the name of the controlling cross-domain resource manager for active cross-domain resources, inactive cross-domain resources, or all cross-domain resources in the major node.

For a Cross-Domain Resource Manager Major Node: The names of all cross-domain resource managers in the major node, and optionally, the network addresses and status of active cross-domain resource managers, inactive cross-domain resource managers, or all cross-domain resource managers in the major node.

When the command calls for a display of all nodes of a particular type, the information displayed includes:

For All Major Nodes: For each active major node in the domain, its name, its type, and, for local NCP major nodes, its communication controller's channel unit address.

For All Terminals: For each terminal in the domain, the major node for which the terminal is a minor node, the name of the terminal, the type of terminal, the nodes between it and the host computer (such as lines and communications controllers), and, for logical units, the physical unit it is associated with and whether the physical unit is active or inactive.

For All Application Programs: The name of each application-program major node in the domain, the name of each application program in the major node, and whether each application program is active or inactive.

For All Physical Units and BSC Cluster Control Units: For each physical unit or cluster control unit in the domain, its name, its type, the major node with which it is associated, the major node type, and whether the unit is active or inactive.

For All Lines: For each line in the domain, whether the line is active or inactive, the names of the local and remote communications controllers between it and the host computer, and, for lines connected to a remote communications controller, the name of the line between the local and remote communications controller.

For All ACF/VTAM Buffer Pools: For each buffer pool in the domain, the name of the buffer pool, the total number of buffers in the pool, the maximum number of buffers that have been used at any one time, and the number of buffers currently being used.

For All Active Cross-Domain Resource Major Nodes in the Domain: The names of all active cross-domain resource major nodes in the domain, and optionally, for each cross-domain resource major node, the names of active cross-domain resource managers, inactive cross-domain resource managers, inactive cross-domain resources, and the status and network address of the cross-domain resource manager that controls each listed cross-domain resource.

For All Active Cross-Domain Resource Manager Major Nodes in the Domain: The names of all active cross-domain resource manager major nodes in the domain, and optionally, for each cross-domain resource manager major node, the names of active cross-domain

resource managers, inactive cross-domain resource managers, or all cross-domain resource managers associated with it, and the status and network address for each listed cross-domain resource manager.

For Pending Session I/O: The name and type of each node with I/O in progress for session establishment or session termination.

Activating and Deactivating Nodes

Before a node can be used in an ACF/VTAM system, the node must be active. The user can control the activation and deactivation of many of the nodes in ACF/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 non-SNA 3270s
- Sets of application programs
- Sets of cross-domain resources
- Sets of cross-domain resource managers

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

To activate or deactivate a major node after ACF/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 ACF/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 ACF/VTAM" and "Halting ACF/VTAM" earlier in this chapter for information on activating and deactivating nodes in ways other than by using 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 ACF/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 non-SNA terminals
- BSC and start-stop terminal components
- Cross-domain resources
- Cross-domain resource managers

When ACF/VTAM is started, the user can also specifically request the activation of any of these minor nodes 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.

As noted earlier in this chapter, halting ACF/VTAM (by using ACF/VTAM's HALT command) automatically deactivates all active nodes. The orderly mode of ACF/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 ACF/VTAM termination is not completed until all ACBs have been closed. The cancel mode (in OS/VS) of the HALT command causes immediate termination of ACF/VTAM and causes all nodes to be immediately deactivated.

To activate or deactivate a minor node while ACF/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 ACF/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.

When *immediate deactivation* is specified, all queued requests for connections to nodes included in the 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 non-SNA terminals, data that is already in ACF/VTAM buffers prior to the deactivation can still be obtained by the application program that was connected to the terminal; data in transit to ACF/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 and immediate deactivations are 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 ACF/VTAM to deactivate the node internally, without waiting for the responses to deactivation commands. *Restart deactivation* causes a forced deactivation and then a warm restart.

Starting and Stopping Application Programs

Starting an ACF/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 ACF/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 ACF/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 ACF/VTAM, an application program is one that is defined within an active major node and that has an open ACB for ACF/VTAM. Thus, for ACF/VTAM, stopping an application program requires only that the ACB be closed. But the major node itself cannot be deactivated until all subordinate application program 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 Non-SNA Terminals

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

Activating a major node for a set of local non-SNA terminals 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 non-SNA terminals with the WARM option activates those terminals that are available and were active when the major node was deactivated. ACF/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 local non-SNA terminal returns the terminal to the operating system. Deactivating the major node for a set of local non-SNA terminals returns all active terminals in that set to the operating system. (Inactive terminals in that set are not currently allocated to ACF/VTAM.)

If an automatic logon is specified for a local non-SNA terminal, 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 non-SNA terminal is activated, it is available for connection to application programs using ACF/VTAM. When the terminal is deactivated, it is unavailable for connection through ACF/VTAM, but it is available to non-ACF/VTAM users.

Activating and Deactivating Local SNA Major Nodes

A local SNA major node can be controlled by activating or deactivating the entire node (including its minor nodes) or by activating or deactivating separate minor nodes (its physical units and logical units). To activate or deactivate the entire node, the name used in filing the major node is specified in the VARY command. An individual minor node is activated or deactivated by using the minor node name (the name of a PU or LU definition statement) in the VARY command.

Activating and Deactivating a Nonshared Local NCP

A local communications controller can be attached to one or more host computers through channels. This section describes activating and deactivating an NCP in a local communications controller that is connected to only one host computer. "Activating and Deactivating a Shared Local NCP" describes these procedures for a shared NCP. Activating an NCP for a local communications controller that is attached to only one host computer allocates the controller to ACF/VTAM. Until the NCP is deactivated, the communications controller remains allocated to ACF/VTAM and is not available to other users of the operating system except through ACF/VTAM facilities. Allocating a local communications controller to ACF/VTAM also implicitly allocates the associated remote attachments to ACF/VTAM. (Only local devices are recognized by the operating system and need to be explicitly allocated to ACF/VTAM; the remote devices are "allocated" to ACF/VTAM, because they are part of the network controlled by the local communications controllers.)

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. ACF/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 next user is responsible for loading another control program if the current one is not acceptable.

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 ACF/VTAM. Deactivation does not halt emulation processing in the controller.

ACF/VTAM's activation and deactivation requests for communication lines may also affect emulation mode. ACF/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

ACF/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.

Activating and deactivating a local NCP can affect cross-domain communication. When deactivating a local NCP, the network operator specifies whether cross-domain links associated with the NCP are to be deactivated. If the cross-domain links are left active, the NCP can still perform a transit node function. If the links are deactivated, cross-domain communication through the local communications controller is stopped.

Activating and Deactivating a Shared Local NCP

When a local communications controller is attached to more than one host computer, the NCP can be loaded from any of the host computers. By coordination among the operators in the domain, an attempt to load a shared communications controller simultaneously from more than one domain can be prevented.

The network operators for each host computer using the NCP activates it using the VARY command and specifies cold or warm restart as desired. Each network operator activates the NCP minor nodes independently to activate the desired resources. For resources that are defined to more than one host access method, ownership of the resources is decided by which access method activates the resources first. Deactivation of the NCP by one network operator does not affect the use of the NCP by other users.

A local communications controller can also be attached to one host computer through more than one channel as long as ACF/VTAM uses only one of the channels. The same rules for activating, deactivating, and loading the NCP attached by multiple channels to the same host computer apply as when attached to different host computers.

Activating and Deactivating a Remote NCP

Terminals attached to a remote communications controller are not available for use by ACF/VTAM until the remote controller has been activated. Before a remote communications controller can be activated, the local communications controller to which it is attached must be activated. Before deactivating a local communications controller in an ACF/VTAM system, the network operator does not have to deactivate remote communications controllers. Deactivating an NCP for a local communications controller automatically deactivates the NCPs in remote communications controllers attached to the local controller. For nonshared local communications controllers, all associated remote communications controllers are deactivated when a local communications controller is deactivated; for shared communications controllers, only the remote communications controllers associated with the host computer that deactivated the local communications controller are deactivated.

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

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 ACF/VTAM. Remote attachments remain implicitly allocated to ACF/VTAM as long as the NCP for the local communications controller is not deactivated by ACF/VTAM.

For ACF/VTAM to treat a terminal as active (that is, for ACF/VTAM to permit a terminal to be connected to or queued for logon 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.

Active application programs can be connected to active terminals (logical units or non-SNA 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 ACF/VTAM. Each of these minor nodes is deactivated by using ACF/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

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 either 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.

Activating and Deactivating Cross-Domain Resource Managers

Before resources can communicate across domain boundaries, ACF/VTAM must activate the cross-domain resource manager in its domain and establish contact with the cross-domain resource managers in other domains. A cross-domain resource manager in a particular domain must be activated before a cross-domain resource manager in another domain can establish contact with it. Thus, activation of a cross-domain resource manager in a particular domain prepares it to receive contact from other domains. Activation of the representation of a cross-domain resource manager in another domain (that is, activation of the minor node definition in this domain of the other domain's manager) causes ACF/VTAM to attempt to contact that manager. The attempt fails unless the other manager has been activated in its own domain.

When a cross-domain resource manager other than the host's is deactivated, communication with the associated domain is no longer possible. Deactivating the host's cross-domain resource manager ends its sessions with all other cross-domain resource managers.

Loss of contact between cross-domain resource managers (for example, as the result of a link failure) is not equivalent to deactivation and does not terminate existing cross-domain sessions. However, in this case, new cross-domain sessions cannot be established until contact between the cross-domain resource managers is reestablished by deactivation and reactivation, in either host computer, of the minor node definition of the other domain's cross-domain resource manager.

Activating and Deactivating Cross-Domain Resources

ACF/VTAM maintains a list of cross-domain resources (terminals and application programs in other domains that can be used by terminals and application programs in the domain). The network operator can make cross-domain resources available to the domain by activating and deactivating cross-domain resource major and minor nodes. For communication to take place between resources in different domains, the cross-domain resource managers must be in contact, the resources must have been activated in their own domains, and the cross-domain resource definitions in the other domains must have been activated.

In the definition statement for each cross-domain resource, a cross-domain resource manager is named as the owner of the resource, and that manager controls the resource. After the cross-domain resource major node containing the definition of the resource has been activated, the network operator can use a **MODIFY** command to transfer ownership of the resource from one cross-domain resource manager to another cross-domain resource manager. This command is used primarily for resource takeover (see "Resource Takeover" in Chapter 6). In resource takeover, when one host computer takes over control of resources that were under a second host computer, the network operator at a third host computer uses this **MODIFY** command to assign cross-domain resources to the cross-domain resource manager in the new controlling host computer. This **MODIFY** command can also be used to transfer control of a resource from an ACF/VTAM cross-domain resource manager to an ACF/TCAM cross-domain resource manager.

Activating Path Tables

The network operator uses the **VARY** command to activate a path table. Activating a path table makes each path defined in that table available for cross-domain communication. The path table cannot be deactivated using the **VARY** command. Once a path table is activated, it is available until ACF/VTAM is halted. The network operator can add paths to destinations for which there is no path defined or replace paths through failed or inactive adjacent communications controllers by activating additional path tables.

When necessary, the network operator can modify a path table. The network operator does this if, because of configuration changes, one or more destination subareas that were formerly reached through one adjacent subarea now have to be reached through a different adjacent subarea. (*Destination subareas* and *adjacent subareas* are described under “Defining Cross-Domain Path Tables” in Chapter 3.) To modify the path table, the network operator uses the VARY command to activate another path table (a set of ACF/VTAM PATH statements stored in the ACF/VTAM definition library under a different member or book name from the name under which the current path table was stored). The new path table contains only those statements needed to define new path relationships. When the new path table is activated, ACF/VTAM modifies its path control information to reflect the new paths defined in the new table. Path definitions in the old path table that are not affected by statements in the new path table are not modified and remain in effect.

As an example, consider the configuration shown in Figure 3-5 in Chapter 3. If the communication’s controller at subarea 18 in domain 1 were lost, the path table in that domain would have to be changed. The new path statement for domain 1 would be:

```
PATH ADJSUB=34,DESTSUB=(32,48,50)
```

This statement would cause the ACF/VTAM in domain 1 to change its path information to make subarea 34 (instead of subarea 18) the adjacent subarea for paths to subareas 48 and 50.

Special Considerations for Activation

When ACF/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.) ACF/VTAM uses the entries in the table to represent the minor nodes. It is these entries that ACF/VTAM allocates to users. (ACF/VTAM does not allocate the node the entry represents; 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 non-SNA terminals are exceptions.

An Active Application Program: ACF/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 ACF/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 ACF/VTAM by only one application program at a time; that is, only one open ACB at a time can refer to it.

ACF/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, ACF/VTAM treats it and its associated table entry as an active application program.

An Active Terminal: When ACF/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 ACF/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 ACF/VTAM activates a non-SNA terminal, it indicates in the terminal’s definition table entry that it is active; no activation command is sent for non-SNA 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 ACF/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.)

Acquiring and Releasing Resources

The network operator can acquire and release resources using the VARY command. This facility is used to take over resources of an NCP whose host computer has failed or to back up a failing communication link. See "Resource Takeover" and "Switched Network Backup" in Chapter 6 for a description of these facilities.

Initiating Requests for Connection

In addition to its use in activating and deactivating nodes, ACF/VTAM's VARY command can initiate connection on behalf of terminals (not including application programs acting as 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. ACF/VTAM then processes the command as though it were an automatic logon for the terminal; ACF/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.

A VARY command cannot be used to establish an automatic logon of one application program to another application program. ACF/VTAM does not provide an automatic logon relationship between application programs.

Starting and Stopping ACF/VTAM Facilities

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

- Network solicitor, a facility for non-SNA terminals, described in Chapter 8
- ACF/VTAM activity traces and internal event traces, described in Chapter 6
- Tuning statistics recording (if gathering of tuning statistics was started with the START command)
- 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 ACF/VTAM activity traces. The network solicitor and the activity traces can also be activated by ACF/VTAM start options. See "Starting ACF/VTAM," earlier in this chapter, for more information on activating these two facilities at start time.

Suppressing Network Operator Messages

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

ACF/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.

The line-scheduling specifications are changed by the network operator by using an option of ACF/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.

Changing NCP Configurations

The device configuration controlled by a communications controller can be changed by deactivating the communications controller and then reactivating it, specifying that a different NCP is to be loaded into the controller. Before this method can be used, the new NCP must be generated.

Network Operator Control from an Authorized Application Program

This section gives a general description of a *program operator*, that is, an ACF/VTAM application program used as a network operator. An ACF/VTAM application program can be authorized to use the SENDCMD macro instruction to issue ACF/VTAM network operator commands (except START and HALT) and the REPLY command, and the RCVCMD macro instruction to receive ACF/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 an ACF/VTAM application program to respond to a prompting message from ACF/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 ACF/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 ACF/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 ACF/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 ACF/VTAM messages.

Within a domain, 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 ACF/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 ACF/VTAM connection and communication macro instructions to communicate with terminals in the network and using the SENDCMD and RCVCMD macro instructions to communicate with ACF/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 ACF/VTAM.

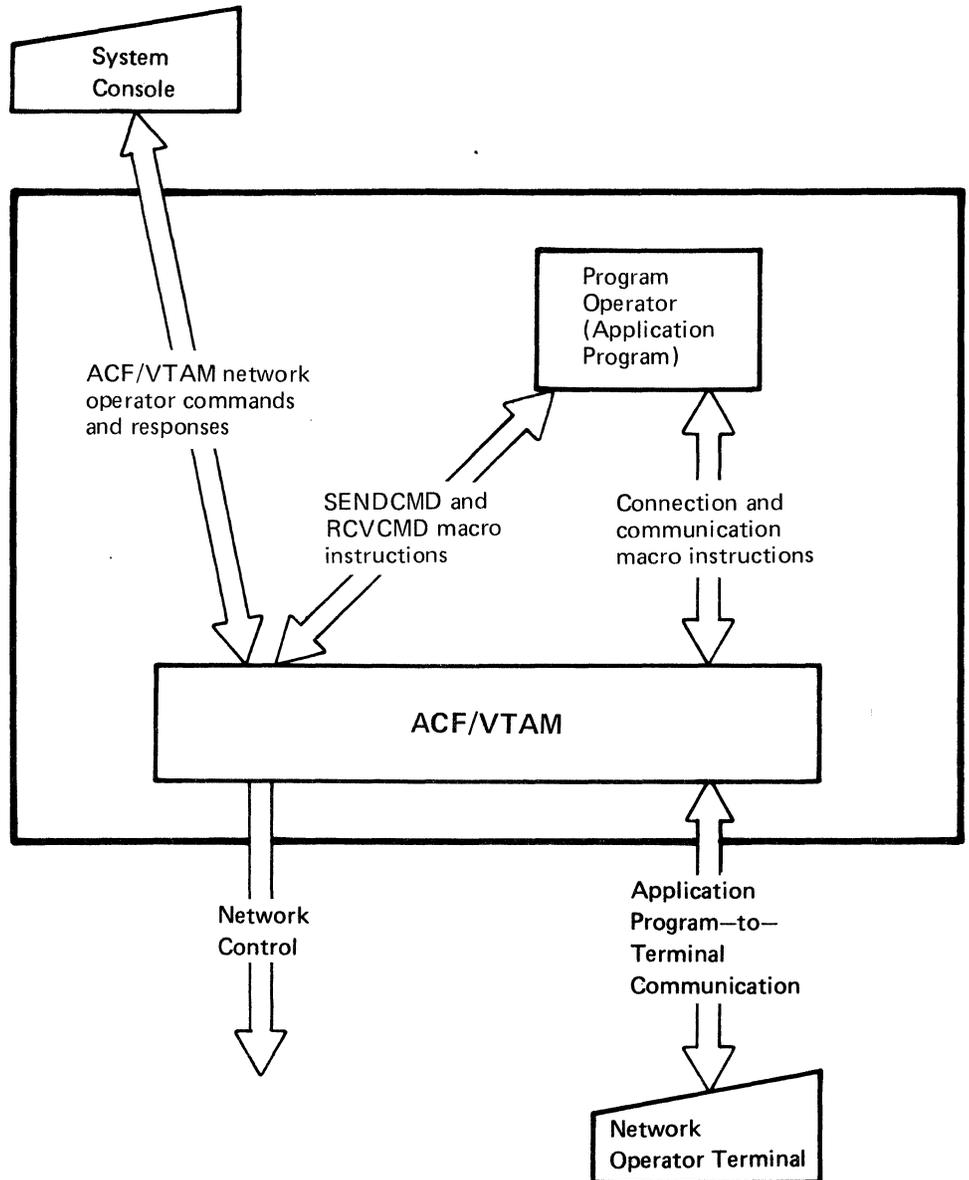


Figure 4-1. Network Operator Control of the ACF/VTAM System

To receive an operator message from ACF/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 ACF/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 ACF/VTAM messages.

ACF/VTAM Macro Language Reference has a detailed description of the SENDCMD and RCVCMD macro instructions. *ACF/VTAM Program Operator Guide* 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 ACF/VTAM network.

Considerations for Network Operator Control

To use ACF/VTAM's network operator facilities effectively, the user must establish procedures for monitoring and controlling the ACF/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: The operator must know the names of all major and minor nodes that are to be used in ACF/VTAM commands or that can be received in ACF/VTAM messages. The operator should also know the names and uses of all ACF/VTAM data sets. (See "ACF/VTAM Data Sets" in Chapter 7 for a description of the data sets used by ACF/VTAM.)

Relationship between the names of ACF/VTAM application programs and job names: Application programs are identified to ACF/VTAM through APPL statements and ACBs. For ACF/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 ACF/VTAM relates to job names or step names.

Relationship between the symbolic and the physical network: Because ACF/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 ACF/VTAM on emulation in an NCP with PEP: Although ACF/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 ACF/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 ACF/VTAM region or partition size can be selected. ACF/VTAM also allows the system console operator to make storage pool specifications. If these two specifications are not predefined to the system or to ACF/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 ACF/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. ACF/VTAM Application Programs

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

Most of the ACF/VTAM application program concepts and facilities described in this chapter apply to all types of terminals supported by ACF/VTAM. Some of the information, however, does not apply to non-SNA terminals used in basic mode. Users of non-SNA terminals in basic mode should read this chapter and then read Chapter 8 for additional concerns for these terminals.

The ACF/VTAM Language

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

- Begin and end the association of an application program with ACF/VTAM

- Establish and terminate connection between an application program and terminals, and also between two application programs

- Transfer messages and responses between an application program and terminals, and between two application programs

- Transfer network operator commands and messages between an application program and ACF/VTAM

- Create, modify, and interrogate control blocks that hold information that is passed between an application program and ACF/VTAM

- Provide support for connection and communication activities

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

ACF/VTAM Macro Instructions

The macro instructions are grouped into the following categories:

- Connection macro instructions

- Communication macro instructions

- Network control macro instructions

- Control-block macro instructions

- Support macro instructions

An application program can function as the primary or secondary end of a session. When communicating with logical units or non-SNA terminals, an application program acts as the primary end of the session; the logical unit or non-SNA terminal acts as the secondary

end. In a connection between two application programs, one application program acts as the primary end of the session and the other application program acts as the secondary end of the session. Primary application programs and secondary application programs use different macro instructions and procedures for establishing and terminating connection; they use the same macro instructions for communication. In certain situations, however, a primary application program can only issue certain commands to the secondary program, and the secondary can only issue certain commands to the primary program. Note that an application program can be primary in one session and secondary in another session at the same time.

Connection Macro Instructions

These macro instructions tell ACF/VTAM that a particular application program is in operation and wants to be associated with ACF/VTAM and, subsequently, request ACF/VTAM to connect the application program to one or more terminals. Connection macro instructions also request ACF/VTAM to disconnect the program from one or more terminals and to disassociate the program from ACF/VTAM.

Application-Program-to-ACF/VTAM Macros: These macro instructions tell ACF/VTAM that an application program is beginning or ending operation.

OPEN: Associates an application program with ACF/VTAM.

CLOSE: Terminates the association with ACF/VTAM.

Primary-to-Secondary Macros: These macro instructions are used by a primary application program to establish and terminate connection to a terminal (remember that a terminal can be a logical unit, a non-SNA terminal, or a secondary application program).

OPNDST: Requests that ACF/VTAM connect the application program to a particular terminal or to a list of terminals.

CLSDST: Requests that ACF/VTAM terminate the connection between the application program and a particular terminal.

SIMLOGON: Causes ACF/VTAM to create a logon for a terminal (or logons for a list of terminals) and to pass the logon (or logons) to the application program that issued the macro instruction as though the logon (or logons) had come from the terminal(s).

Secondary-to-Primary Macros: These macro instructions are used by a secondary application program to establish and terminate connection to a primary application program.

REQSESS: Requests ACF/VTAM to initiate a logon to a designated primary application program.

OPNSEC: Requests ACF/VTAM to complete the connection between the secondary application program and a designated primary application program.

SESSIONC: Requests that ACF/VTAM send a negative response to a connection command (a Bind command) sent by a primary application program. SESSIONC is also used for other types of session control and is described in "Communication Macro Instructions" below.

TERMSESS: Requests ACF/VTAM to terminate the connection with a designated primary application program.

Communication Macro Instructions

VTAM provides two types of communication macro instructions: *basic-mode* macro instructions and *record-mode* macro instructions. In general, an application program uses basic-mode macro instructions to communicate with non-SNA terminals; an application

program uses record-mode macro instructions to communicate with logical units and other application programs. However, some terminals can be used with either mode (see Appendix A). This section describes the record-mode communication macro instructions. Basic-mode communication macro instructions are described in Chapter 8.

RECEIVE: Requests ACF/VTAM to transfer a message, command, indicator, or response 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 a command, an indicator, or a response).

SEND: Requests ACF/VTAM to transmit a message, command, or response to a terminal. Data is transferred from an area in the application program. Commands, indicators, and response information are specified symbolically in the SEND macro instruction or in the RPL, and no output area is required.

SESSIONC: Requests ACF/VTAM to send a command that either (1) starts or stops the possibility of exchanging messages with the SEND and RECEIVE macro instructions, (2) assists in synchronizing the message sequence numbers, or (3) sends a negative response to a primary application program that has sent a connection command (Bind).

RESETSR: Changes the mode of receiving input from a particular logical unit, application program, or non-SNA 3270 terminal. The modes are *continue-any* mode and *continue-specific* mode. These modes are described later in this chapter. RESETSR can also be used to cancel outstanding requests for input from a specified terminal.

Network Control Macro Instructions

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

RCVCMD: Receives network operator messages from ACF/VTAM.

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

Control-Block Macro Instructions

These macro instructions build and manipulate control blocks required by ACF/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 ACF/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 ACF/VTAM schedules when certain conditions occur (for example, when a logon is received from a terminal).

NIB: Creates a node initialization block (NIB), which provides ACF/VTAM general information about the communication that will occur between the application program and a particular terminal. This information is provided to ACF/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 describes a connection, data-transfer, or related request to ACF/VTAM. On completion of the requested operation, the RPL contains information that ACF/VTAM passes to the application program.

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 any future releases of ACF/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.

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

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 ACF/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).

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 ACF/VTAM application can manipulate values in control blocks by using DSECT and other assembler language instructions.

Support Macro Instructions

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

CHECK: Checks and, if necessary, awaits completion of an asynchronous 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 one of those exit routines to be executed.

EXECCRPL: Requests that ACF/VTAM perform an operation that is currently specified in a designated RPL. EXECCRPL can be used:

To reissue a specified request. When a SYNAD exit routine is entered with a return code indicating a retrievable condition, EXECCRPL can be issued with assurance that the RPL contains the desired values. Use of EXECCRPL 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 EXECCRPL, 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 EXECCRPL macro is issued. The use of EXECCRPL can result in fewer instructions being executed or in less storage being taken up by macro expansions or both. Note that EXECCRPL 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 user logon message associated with a logon; the set of session parameters associated with the request; the number of terminals connected to, or queued for logon to, the application program; the device characteristics of a terminal; the name of the terminal whose logon has been queued the longest for an application program; 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 **ACF/VTAM** to start and stop the application program's ability to receive connection requests.

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

Relating ACF/VTAM Control Blocks and Executable Macro Instructions

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

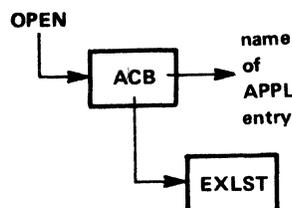
The most frequently used control block is the **RPL** (request parameter list). An **RPL** is required each time the program makes a request for a connection, for an I/O operation, or for any service except opening and closing an **ACB**. The other control blocks, the **ACB**, **EXLST**, and **NIB** (node initialization block), are used infrequently.

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** field to the address of **INFO** and the **AREALEN** field to 200 before initiating the input operation. These values remain in these fields for all subsequent requests that use **RPL1** until the values are changed again.

Opening and Closing an Application Program

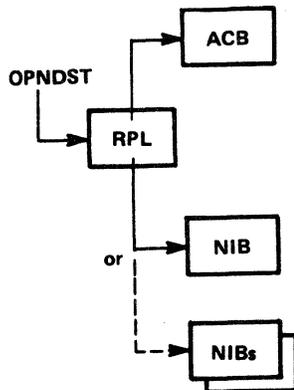


The OPEN macro instruction indicates that a program is associating itself with ACF/VTAM so it can use ACF/VTAM facilities. The OPEN specifies an ACB; the ACB in turn points to a location in the program that contains the name of the application program as defined in an APPL statement during ACF/VTAM definition. 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 were named in the exit list are eligible for scheduling by ACF/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 ACF/VTAM as more than one application program. (ACF/VTAM views each open ACB as a separate application program.) Many ACF/VTAM users will find it satisfactory to open only one ACB for each program.

The CLOSE macro instruction notifies ACF/VTAM that an application program is detaching itself from the ACF/VTAM system. A single CLOSE macro instruction can close more than one ACB. Besides disassociating the program from ACF/VTAM, the CLOSE macro instruction disconnects any terminals that are still connected to the program when the CLOSE is issued.

Primary Application Program Connection



Before communicating with a terminal (logical unit, secondary application program, or non-SNA terminal), a primary application program must have itself connected to the terminal. Connection can be initiated by the terminal, the network operator, ACF/VTAM, another primary application program (using the CLSDST macro instruction with the PASS option), or the application program itself (using the SIMLOGON macro instruction or the OPNDST macro instruction with the ACQUIRE option). In each 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. 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, when the connection is to be for record-mode operations, 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 that were identified when the ACB was opened.

When a terminal is connected (as the result of an OPNDST macro instruction), ACF/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, ACF/VTAM places a communication identifier (CID), which identifies the session with the terminal. On all subsequent I/O requests for the terminal, the application program must be sure that this CID is present in the RPL. In addition to the CID, ACF/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.

Primary Application Program Disconnection

A 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, an efficient method is to issue a series of separate asynchronous CLSDST macro instructions to disconnect the terminals and then issue the CLOSE macro instruction. A less efficient method is to omit the CLSDST macro instructions and issue the CLOSE macro instruction, which not only closes the ACB but also disconnects all connected terminals.

Secondary Application Program Connection

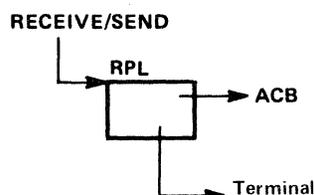
A secondary application program requests connection to a primary application program by issuing a REQSESS macro instruction that specifies an RPL that is associated with the request. The RPL contains the address of a NIB that identifies the primary application program and contains information that applies to the subsequent communication. The REQSESS macro instruction causes ACF/VTAM to create a logon and to pass the logon to the primary application program. The logon appears to the primary application program as a logon from a logical unit.

After the primary application program issues an OPNDST macro instruction (as described earlier), the secondary application program issues an OPNSEC macro instruction to complete the connection process. The OPNSEC macro instruction specifies an RPL that is associated with the request and specifies a NIB. Upon completion of the OPNSEC macro instruction, the NIB contains a CID to be used in all subsequent I/O operations with that primary application program.

Secondary Application Program Disconnection

A secondary application program requests disconnection from a primary application program by issuing a TERMSESS macro instruction. The TERMSESS macro instruction causes a logoff that appears to the primary application program as a logoff from a logical unit. The TERMSESS macro instruction indicates whether disconnection is conditional (at the discretion of the primary application program) or unconditional. (ACF/VTAM disconnects the secondary application program immediately and informs the primary program.)

Communicating in Basic Mode or Record Mode



Having opened the ACB and connected one or more terminals, a primary application program can communicate with each connected terminal by issuing basic-mode or record-mode communication macro instructions as appropriate for that terminal. In performing each operation, ACF/VTAM obtains from the RPL the name of the application program that made the request, the identity of the terminal (if a specific terminal is being addressed), and other information about the request.

Record-mode macro instructions are used to communicate with logical units, secondary application programs, and certain specially defined terminals. Basic-mode macro instructions are used to communicate with non-SNA terminals.

A secondary application program uses record-mode macro instructions to communicate with a primary application program. The macro instructions are the same as those used by a primary application program to communicate with a secondary application program; the only difference is that a secondary application program cannot send certain commands to the primary program.

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

The application program (whether it performs primary or secondary functions or both types of functions) can handle control blocks in a number of ways. It can:

- Define ACBs, RPLs, NIBs, OR EXLSTs in the application program during assembly or generate them, using the GENCB macro instruction, during program execution

- Create a control block for each function or use available control blocks for different functions as required

- 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 sessions concurrently, it may be useful to have a control block work area other than the RPL or NIB associated with a particular session. The queuing of input or output by the application program may also require a session-related control block work area. The RPL contains a USER field that may be used to associate storage with a particular session. The application program initially associates the storage with the session when connection is established by specifying an address in the USERFLD field of the NIB. Thereafter, whenever input is received from the terminal, ACF/VTAM puts that address into the RPL's USER field.

Organization of the Application Program

ACF/VTAM provides facilities that allow an application program (whether it performs primary or secondary functions or both types of functions) to be organized in a variety of ways. This section describes some of the major ACF/VTAM facilities that affect program organization. See the *ACF/VTAM Macro Language Guide* for other considerations.

Overlapping ACF/VTAM Requests with Other Processing

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

Synchronous request handling means that ACF/VTAM does not return control to the next sequential instruction until the requested operation has been completed. Program execution is halted until ACF/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 ACF/VTAM returns control to the next sequential instruction as soon as ACF/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 ACF/VTAM that will carry out the operation. While the operation is being completed, the application program is free to initiate other data-transfer operations or do other processing. For example, an application program might issue a RECEIVE macro instruction and indicate that it is to be handled asynchronously; as soon as ACF/VTAM accepts the request and prepares the operation, ACF/VTAM returns control to the application program. The application program can then do other processing (such as writing to a direct-access storage device or sending a message to another terminal) while ACF/VTAM is performing the RECEIVE operation.

When asynchronous request handling is used, there are two ways that ACF/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, ACF/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. When the operation is completed, ACF/VTAM interrupts the application program and 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.

Regardless of whether an application program waits on an ECB or uses an RPL exit routine, a CHECK macro instruction must be issued after an asynchronous operation to mark the RPL inactive and to make it available for another operation. The CHECK macro instruction also clears the ECB.

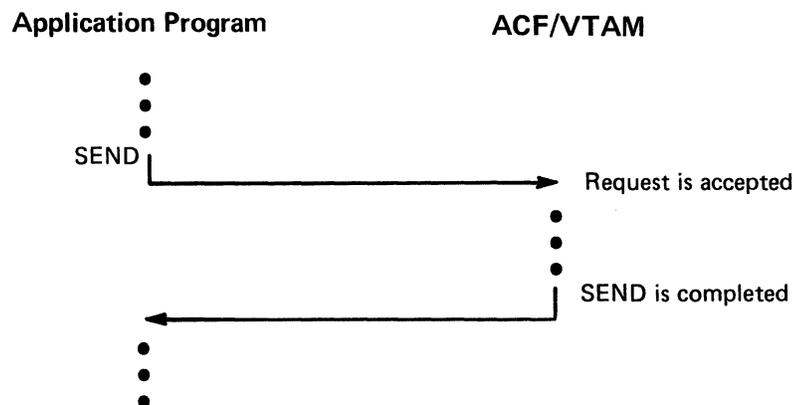


Figure 5-1. Processing Pattern for a Synchronous Request

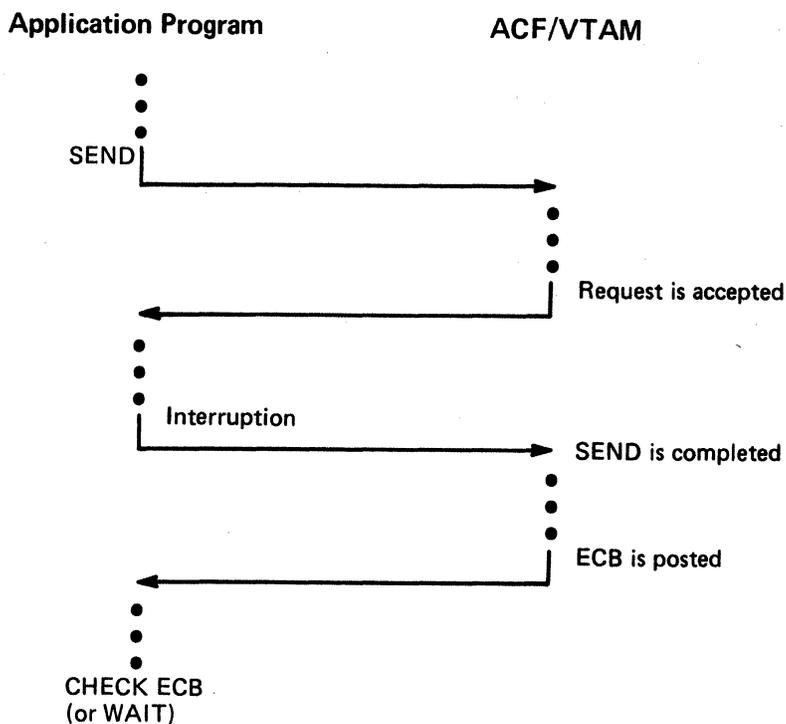


Figure 5-2. Processing When an ECB is Used with an Asynchronous Request

EXLST Exit Routines

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

Event	Exit Routine
A logon has been received from or has been entered on behalf of a terminal, and the logon is waiting to be processed.	LOGON
Connection with a terminal has been temporarily interrupted or permanently lost; the terminal has requested that the session be terminated; or an event has occurred that may affect future operation of the session.	LOSTERM
A session with a terminal has been broken because of a session outage, or a session that is partially established cannot be fully established.	NSEXIT
A connected terminal is wanted by another application program.	RELREQ
ACF/VTAM is being shut down.	TPEND
The application program has made an invalid 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 non-SNA terminals.

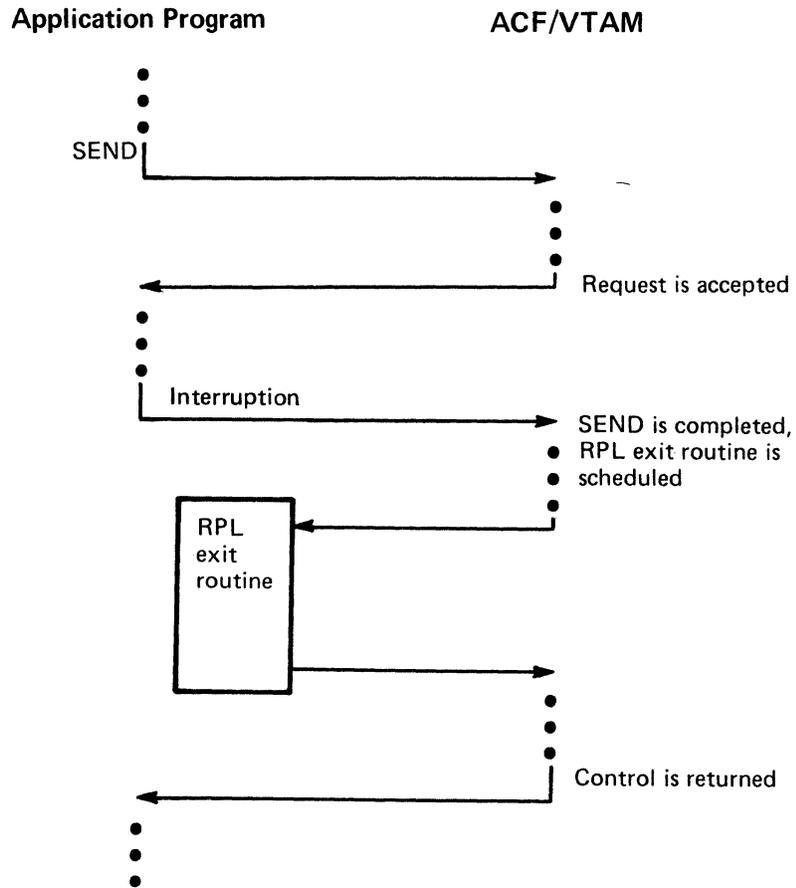


Figure 5-3. Processing Pattern When an RPL Exit Routine is Used with an Asynchronous Request

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 ACF/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 is 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, ACF/VTAM indicates whether it has accepted or rejected the request. When the actual operation has been completed, ACF/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 5-4 shows how errors are reported during asynchronous processing.

Information about success or failure is sent to the ACF/VTAM application program in register 15 and, under some circumstances, in register 0. If the request is rejected or the operation is unsuccessful, ACF/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 user can code two error-handling routines that ACF/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 ACF/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.

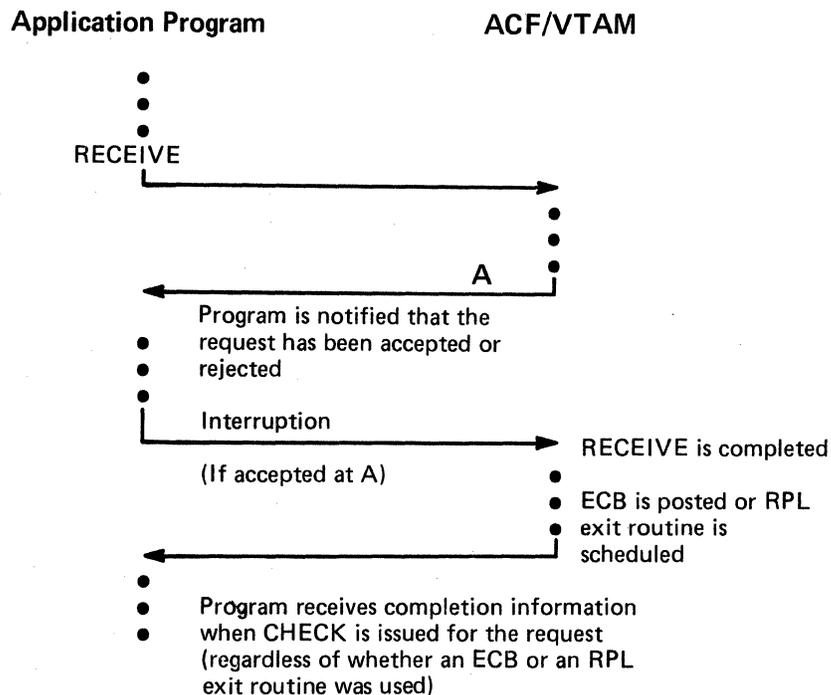


Figure 5-4. Processing Pattern for Reporting Errors during an Asynchronous Operation

The ACF/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. ACF/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.

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 may be able to continue or it might have to terminate itself.

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 terminating the application program.

SYNAD Processing: Physical and environmental errors usually require more extensive treatment than logical errors. 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 problem is a recurring hardware check for the logical unit, 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 ACF/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 was 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 the exit routine cannot correct the error, it can set 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, an ACF/VTAM application program can use ACF/VTAM and operating system facilities that allow multithreading, multitasking, and use of multiple ACBs. In OS/VS2 MVS, an ACF/VTAM application program can specify execution of individual SEND, RECEIVE, RESETSR, and SESSIONC macro instructions in a path that requires fewer instructions (authorized path). See the *ACF/VTAM Macro Language Guide* for descriptions of these facilities.

Application Program Concepts and Facilities

ACF/VTAM's primary purpose is to provide communication between an application program and terminals. This section discusses an application program's ability to have a terminal allocated to it and its ability to communicate with that terminal. It also describes an application program's ability to act as a terminal (that is, as a secondary end of session) when communicating with other application programs.

Opening an Application Program

Before using any ACF/VTAM facilities, an ACF/VTAM application program must issue an OPEN macro instruction to identify itself to ACF/VTAM. An ACF/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 ACF/VTAM normally lasts for the duration of the application program's execution.

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

Primary Application Program Connection

A primary application program must issue an OPNDST (open destination) macro instruction to establish connection with a terminal (logical unit, secondary application program, or non-SNA terminal) before communication with that terminal can begin. OPNDST initializes ACF/VTAM's and the application program's control blocks to indicate that the terminal and the application are connected. For terminals used in record mode, 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 ACF/VTAM's terminal-sharing facility, connections to terminals can be made, broken, and remade throughout the duration of the application program's execution. A primary application program can establish connection either by *accepting* a terminal or by *acquiring* a terminal.

Acceptance

When an application program accepts a 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 primary application program, the terminal itself, or ACF/VTAM—the request usually originates *outside* of the application program. (Logons 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 ACF/VTAM notifies an application program of a logon from a terminal other than a secondary application program, 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 rejected by the application program, the user should ensure that application programs avoid leaving terminals on this queue any longer than necessary.*

Accepting Terminals with an Exit Routine: An application program can maintain a LOGON exit routine that ACF/VTAM schedules whenever a logon for the application program is received. ACF/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 (OPNDST with OPTCD=ACCEPT) that ACF/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, ACF/VTAM queues logons but does not schedule the application program's LOGON exit routine. After a SETLOGON with OPTCD=START is issued, the LOGON exit routine is scheduled for each logon. An application program can issue another SETLOGON macro instruction to stop the scheduling of the LOGON exit routine (SETLOGON with OPTCD=STOP) or to stop the queuing of logons (SETLOGON with OPTCD=QUIESCE). The SETLOGON macro instruction has no effect, however, on completion of an outstanding OPNDST with OPTCD=ACCEPT. If the application program has issued an OPNDST with OPTCD=ACCEPT to accept a particular terminal (or with OPTCD=ACCEPT,ANY to accept a logon received from any terminal), and a logon is received from the terminal, the logon is processed and the connection is made, whether or not SETLOGON START has been issued.

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 ACF/VTAM the identity of the terminal; connection is not made until a logon has been issued for that particular terminal. In accepting a logon from any terminal, after connection is established, ACF/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 ACF/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 for OPNDST with OPTCD=ACQUIRE specifies that *as many terminals from the set as are available* are to be acquired. The CONALL option is used with OPNDST OPTCD=ACQUIRE 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). ACF/VTAM provides information so that the program can determine which terminals were available. The CONALL option for SIMLOGON with a NIB list specifies that simulated logons are to be generated for *all* terminals identified in the NIB list (if they are all active and available); if any one is not active or available, no simulated logons are to be generated, and failure of the macro instruction is to be reported to the application program.

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 acceptable.

Logical units and non-SNA terminals can be connected to only one primary application program at a time; secondary application programs can be connected to more than one application program at a time. If an application program attempts to acquire a logical unit or non-SNA terminal that is already connected to another application program, no reconnection is possible until the current owner disconnects it. ACF/VTAM allows an application program to request another application program's logical unit or non-SNA 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.

ACF/VTAM notifies the owning application program by scheduling the program's RELREQ exit routine. The owning application program can prevent being notified of this type of connection request by not including a RELREQ exit routine. Notification can therefore only occur when both application programs so indicate.

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 the terminal will be returned when it is no longer being used. When the terminal is disconnected from the new owner, it is reconnected 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.

Queuing Connection Requests

Application program requests for connection always fail if the terminal is inactive. An application program can indicate whether its connection request should fail or remain pending (queued) if the requested terminal is active but unavailable. OPNDST (with the ACCEPT option) and SIMLOGON macro instructions can request queuing; OPNDST (with the ACQUIRE option) cannot request queuing. Figure 5-5 lists the effects of different kinds of connection requests.

Type of Connection Request	Meaning When Connection Request is to be Queued	Meaning When Connection Request is not to be Queued
<p>OPNDST ACCEPT:</p> <p>A specific terminal (SPEC)</p> <p>Any terminal (ANY)</p>	<p>Connect the specified terminal if a logon 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:</p> <p>A set of one (CONANY)</p> <p>Any one of a set (CONANY)</p> <p>As many as are available in a set (CONALL)</p>	<p>(Cannot be queued)</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>Connect all terminals in the set (NIB list) that are available. If none are available, indicate failure in the return code.</p>
<p>SIMLOGON:</p> <p>A specific terminal (CONANY or CONALL with a single NIB)</p> <p>Any one of a set (CONANY)</p>	<p>A. If the terminal is active¹, live², and available³, generate the logon and either pass it to the application program⁴ or queue it for the application program. The return code for the SIMLOGON request indicates successful completion.</p> <p>B. If the terminal is active but is not live or is not available, queue a session initiation request for the application program. Generate the logon when the terminal becomes live and available. The return code for the SIMLOGON request indicates successful completion.</p> <p>C. If the terminal is inactive, indicate failure of the SIMLOGON request in the return code.</p> <p>Functionally equivalent to a series of SIMLOGONS, with one SIMLOGON attempted in sequence for each terminal in the set (NIB list). For each SIMLOGON attempt for a terminal in the list, items A and B above in this column apply to the attempt. The attempts stop when ACF/VTAM finds an available terminal and generates the logon.</p> <p>If no one terminal in the NIB list is live and available, a session initiation request is queued for each terminal in the list that is active. When one of those terminals becomes live and available, a logon is created for that terminal, and all other queued session initiation requests generated by the SIMLOGON are canceled.</p> <p>If no terminal in the list is active, failure of the SIMLOGON is indicated in the return code.</p>	<p>A. If the terminal is active¹, live², and available³, generate the logon and either pass it to the application program⁴ or queue it for the application program. The return code for the SIMLOGON request indicates successful completion.</p> <p>B. If the terminal is inactive, or is not live, or is not available, indicate failure of the SIMLOGON request in the return code.</p> <p>Functionally equivalent to a series of SIMLOGONS, with one SIMLOGON attempted in sequence for each terminal in the set (NIB list). A logon is created for the first terminal in the list that is active, live, and available.</p> <p>If no terminal in the list is active, live, and available, failure of the SIMLOGON is indicated in the return code.</p>

Figure 5-5 (Part 1 of 2). Queued and Nonqueued Connection Requests

Type of Connection Request	Meaning When Connection Request is to be Queued	Meaning When Connection Request is not to be Queued
All in a set (CONALL)	Functionally equivalent to a series of SIMLOGONS, with one SIMLOGON attempted for each terminal in the set (NIB list). For each terminal that is immediately available, ACF/VTAM generates a logon and passes it to the application program or queues it for the application program. For each terminal that is active, but not live or not available, ACF/VTAM queues a session initiation request, which is converted to a logon when the terminal becomes physically connected or available. If any one terminal in the set is inactive, failure of the SIMLOGON is indicated in the return code.	Functionally equivalent to a series of SIMLOGONS, with one SIMLOGON attempted for each terminal in the set (NIB list). When all terminals in the set are active, live, and available, a logon is generated for each terminal and passed to or queued for the application program. If any terminal in the set is not active, or not live or not available, the request fails and the failure is indicated in the return code. In this case, no logon is generated.
<p>1 "Active" means that the terminal has been activated and additionally, for switched logical units that are dial-in only, that a dial connection has been established.</p> <p>2 All active terminals are "live" except for dial-in start-stop and BSC terminals that have been activated but have not yet dialed in.</p> <p>3 "Available" means that the terminal is not connected to or queued for connection to another application program.</p> <p>4 The logon is passed immediately to the application program if the program has issued an OPNDST ACCEPT for the specific logical unit or an OPNDST ACCEPT,ANY to accept any logical unit, or if the program's LOGON exit routine can be scheduled (the program has issued SETLOGON START). Otherwise, the logon remains pending, awaiting an OPNDST or the issuance of a SETLOGON START.</p>		

Figure 5-5 (Part 2 of 2). Queued and Nonqueued Connection Requests

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 logical unit or non-SNA terminal, it 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 acquire a secondary application program, it is available if it has opened its ACB and has issued a SETLOGON (with the START option) macro instruction; being connected to a primary application program does not make a secondary application program unavailable. 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.)

Establishing Sets of Session Parameters

When a primary 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). Some of 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. *ACF/VTAM Macro Language Reference* describes session parameters used by an ACF/VTAM application program.

Session parameters have significance only for sessions with terminals used in record mode. For most record-mode terminals, the session parameters are sent to the terminal during the connection process and are used by the terminal to govern its operations during the session. But for local 3270 and BSC 3270 terminals used in record mode, session parameters can be specified by the application program or acquired by ACF/VTAM by

default, but the parameters are not sent to the terminal and have no effect on terminal operations. (Although in these two cases the session parameters are not sent to the terminals, the parameters must be appropriate for 3270 operating characteristics.) Non-SNA terminals used in basic mode do not use session parameters, and any session parameters specified for them during connection are ignored.

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. ACF/VTAM uses this name to search a logon mode table for the desired session parameters.

Specify that the session parameters contained in a 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 is a logon mode name. For logical units and secondary application programs, ACF/VTAM uses the logon mode name to search a logon mode table for the associated session parameters to be used. An application program can examine the session parameters associated with a pending logon by using the INQUIRE macro instruction.

Secondary Application Program Connection

When a primary application program issues a OPNDST macro instruction (with either the ACCEPT or ACQUIRE option) for a secondary application program, ACF/VTAM notifies the secondary application program by scheduling its SCIP exit routine. ACF/VTAM provides the address of the Bind command associated with the request. The Bind command includes the name of the primary application program and a set of session parameters. The secondary application program can accept the connection request by issuing an OPNSEC (open secondary) macro instruction or reject the request by issuing a SESSIONC macro instruction.

As described in “Primary Application Program Connection” earlier in this chapter, the primary application program can acquire a terminal or accept a logon from the terminal. To log onto a primary application program, a secondary application program issues a REQSESS (request session) macro instruction. The secondary application program indicates the name of the primary application program requested, a logon mode name, and optionally, a user logon message. The resulting logon appears to the primary application program as a logon from a logical unit.

If the logon is accepted by the primary application program, the secondary application program is notified in its SCIP exit routine, as described above. If the logon is rejected, the secondary application program is notified in its NSEXIT exit routine.

Primary Application Program Disconnection

To disconnect a terminal, a primary application program can *release* it or it can *pass* it to another primary application program. Releasing and passing are accomplished by using the RELEASE and PASS 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 ACF/VTAM.

Secondary Application Program Disconnection

A secondary application program can request disconnection from a primary application program by issuing a TERMSESS macro instruction. The TERMSESS macro instruction indicates whether the request is for *conditional* disconnection (at the discretion of the

primary application program) or *unconditional* disconnection (immediate disconnection by ACF/VTAM).

If conditional disconnection is specified, ACF/VTAM notifies the primary application program that the secondary application program has requested disconnection. The primary application program can ignore the request or issue a CLSDST macro instruction to end the session. When the CLSDST macro instruction is issued, ACF/VTAM notifies the secondary application program in its SCIP exit routine. If unconditional disconnection is specified, ACF/VTAM disconnects the secondary program and then notifies the primary application program that the session is ended and that it is to issue a CLSDST macro instruction. ACF/VTAM notifies the secondary application program that the session is ended by scheduling its SCIP exit routine.

A secondary application program can also request disconnection by sending a Request Shutdown command to the primary application program. See "Using the Request Shutdown Protocol" in Chapter 3 for a description of this facility.

Record-Mode Communication

ACF/VTAM provides two modes of data transfer, basic and record. Basic mode is used for communication between a primary application program and a non-SNA terminal. Record mode is used for communication between a primary application program and a logical unit or secondary application program. Local and BSC 3270 terminals can be used in either mode. Note that when two application programs communicate, they both use record-mode macro instructions. Record-mode concepts apply to both ends of application-program-to-application-program sessions. 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.

Communication in record mode requires an understanding of concepts related to:

- Messages and responses

- ACF/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 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 furnished by the ACF/VTAM application program in a designated data area (perhaps an answer to an inquiry previously forwarded) and other information that is specified symbolically to ACF/VTAM when the program requests ACF/VTAM to send the message (such as the kind of acknowledgment wanted). Or, in some cases, a message can contain no data but be intended to control the further exchange of data. (When a message contains control information only, it is often called a *command*.) For example, the program could specify that ACF/VTAM send a message on its behalf to a terminal indicating that the terminal should stop sending until released to send by the ACF/VTAM application program.

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

A message or a response is sent by the ACF/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 an ACF/VTAM application program by issuing a RECEIVE macro instruction. (In some cases, input can also be received as the result of ACF/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 ACF/VTAM receives a SEND macro instruction request, it moves the data from the application program to its own buffers and, together with the control information specified in the SEND macro instruction, formats the message.

Normal and Expedited Message Flow

The ACF/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 an ACF/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 ACF/VTAM application program to the logical unit.

Session Control and Data Flow Control

Ordinarily, an ACF/VTAM application program and a terminal (or two connected application programs) are in a level of message control in which data and related control information and responses are exchanged. Using special commands, however, an ACF/VTAM application program can stop and restart the exchange of data messages and related control information and responses. While this flow is stopped, special *session control* commands can be used to exchange information about *sequence numbers* with which each normal-flow message is associated.

The SESSIONC macro instruction is used to start, stop, and exchange information about sequence numbers. Figure 5-6 shows an example of SESSIONC used to stop and restart the data flow.

Requesting a Response

A message specifies the conditions under which a response to that message is wanted. The message sender can request a definite response (either a positive or a negative response is to be returned as appropriate) or an exception response (a response is to be returned only if a negative response is appropriate). The receiver of the message must respond as appropriate. An ACF/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 ACF/VTAM application program determines the type of response required after it has received a message by examining the RESPOND field of the RPL. Figure 5-7 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.

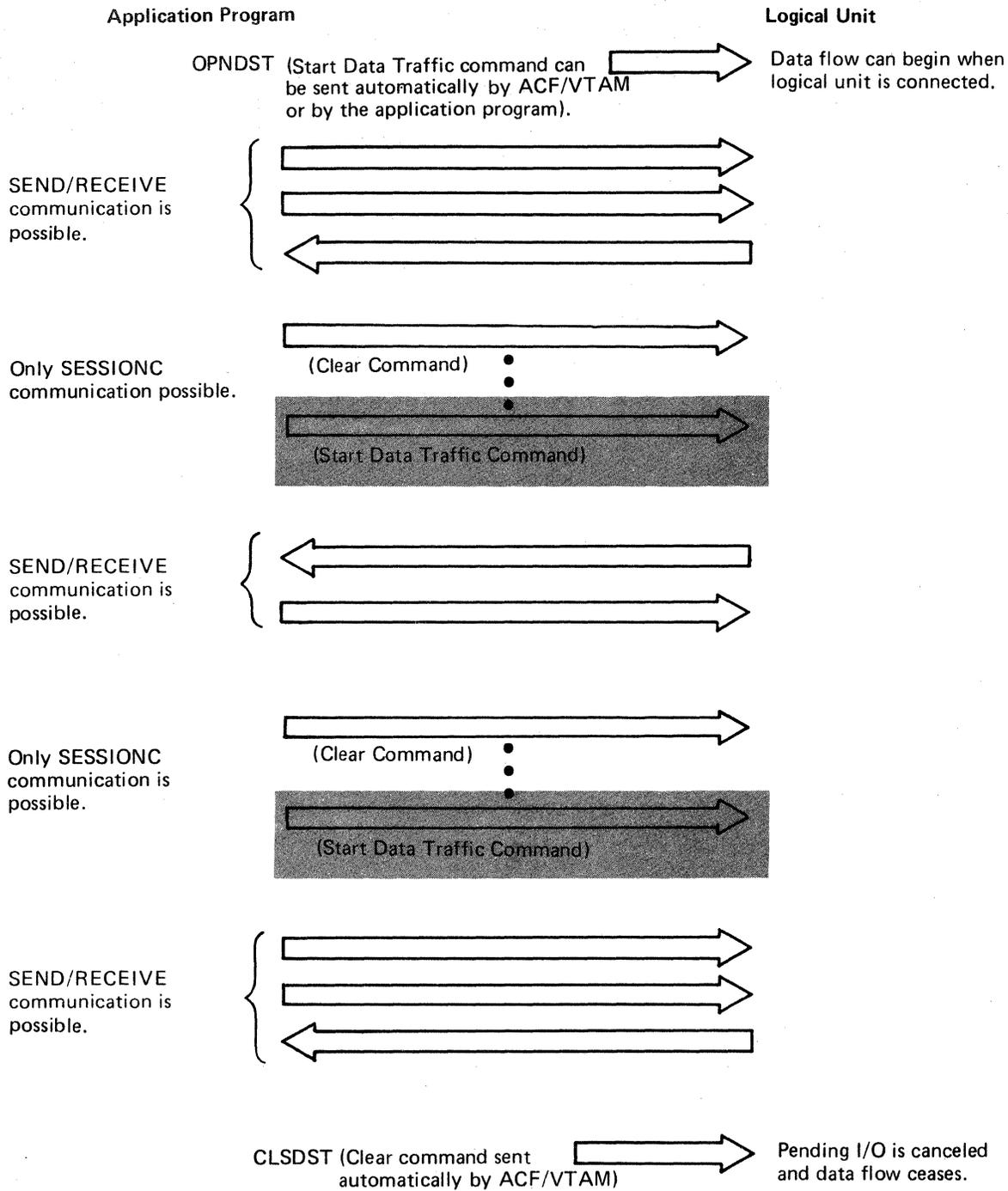
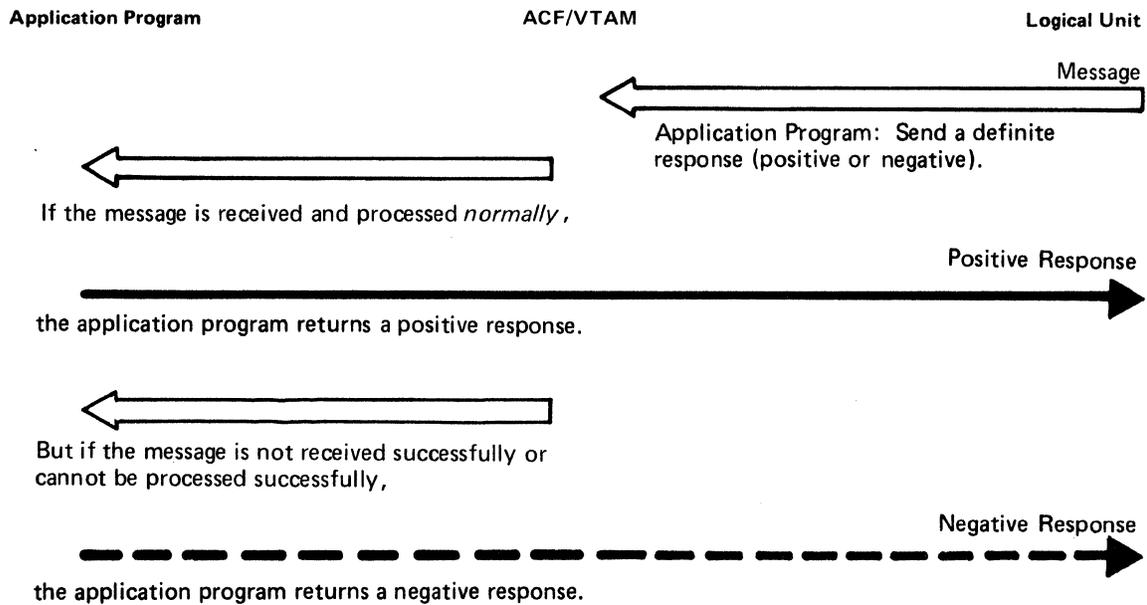
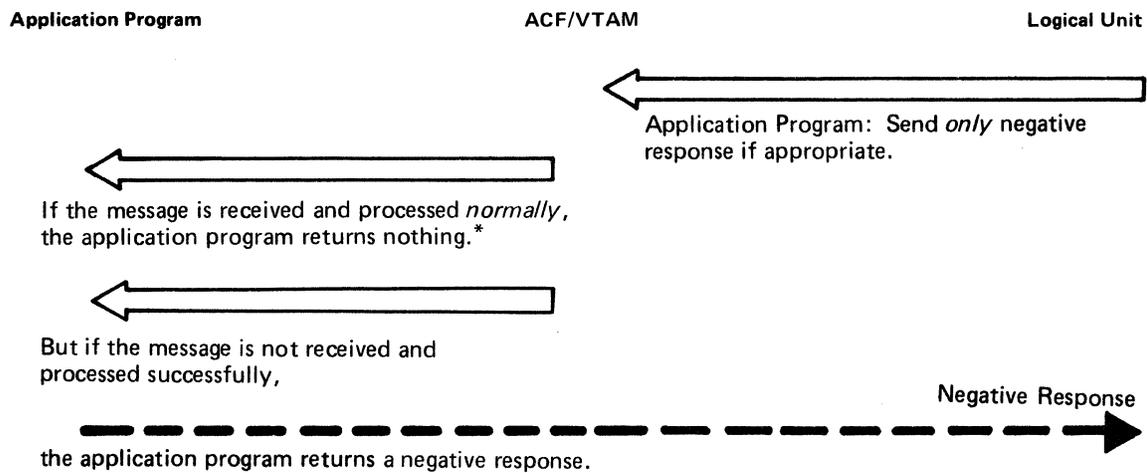


Figure 5-6. An Example of Start Data Traffic and Clear Commands

A Logical Unit Requests a Definite Response



B Logical Unit Requests Only an Exception Response



* Even though a message arrives normally (as far as ACF/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 5-7. An Example of a Logical Unit Requesting (A) a Definite Response and (B) Only an Exception Response

Positive and Negative Responses

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.

An ACF/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 ACF/VTAM as positive or negative when the routine is invoked.

A negative response to a message can be sent even though it arrived successfully. For example, a negative response might indicate that the data in the message was not in a prescribed form. The negative response, in this case, indicates that even though the message arrived successfully, it could not be processed by the receiver.

Definite Response Types 1 and 2

Every response, whether positive or negative, is designated by its sender as a definite response 1 or definite response 2. Some users 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.

Special Handling of a Response

In special circumstances, the sender of a message may want the response to that message to be handled by ACF/VTAM as though the response were an incoming message rather than a response. To do this, the sender of the message specifies a special parameter (PROC=ORDRESP) in the NIB at the time of connection and specifies a special parameter (RESPOND=QRESP) when the message is sent.

Normally, if ACF/VTAM has received a message and a response for an application program, it will give the program the response before the message. When special handling of responses has been specified, however, messages and responses are passed to the program in the exact order they are received. This can be important in special circumstances, such as ensuring proper execution of a Chase command or proper receipt of an End Bracket indicator.

Further information on the effect and uses of the QRESP specification for responses is provided in *ACF/VTAM Macro Language Guide*.

Sequencing and Chaining

ACF/VTAM assigns a sequence number to each normal-flow message sent by an application program using 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 disconnection unless interrupted earlier by the application program. A logical unit assigns sequence numbers to each normal-flow message sent by a logical unit; ACF/VTAM assigns sequence numbers to each normal-flow message sent by a BSC or local 3270 terminal used in record mode.

Should a message arrive out of sequence (that is, its sequence number is not one greater than that of the last one received), ACF/VTAM considers this to be a transmission error and replaces the message with a substitute 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 logging each message received or sent.

Application programs or logical units 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, a message in 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 the Cancel command 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 user. Figure 5-8 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 logical unit 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.

Scheduled and Responded Output

An application program can send a message 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, ACF/VTAM is to consider the output operation completed (by posting an ECB or scheduling an RPL exit routine). Scheduled output is illustrated in Figure 5-9.

Responded output: The application program indicates that ACF/VTAM is not to consider the operation completed until the message has been received and a response has been returned. Responded output is illustrated in Figure 5-10.

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.

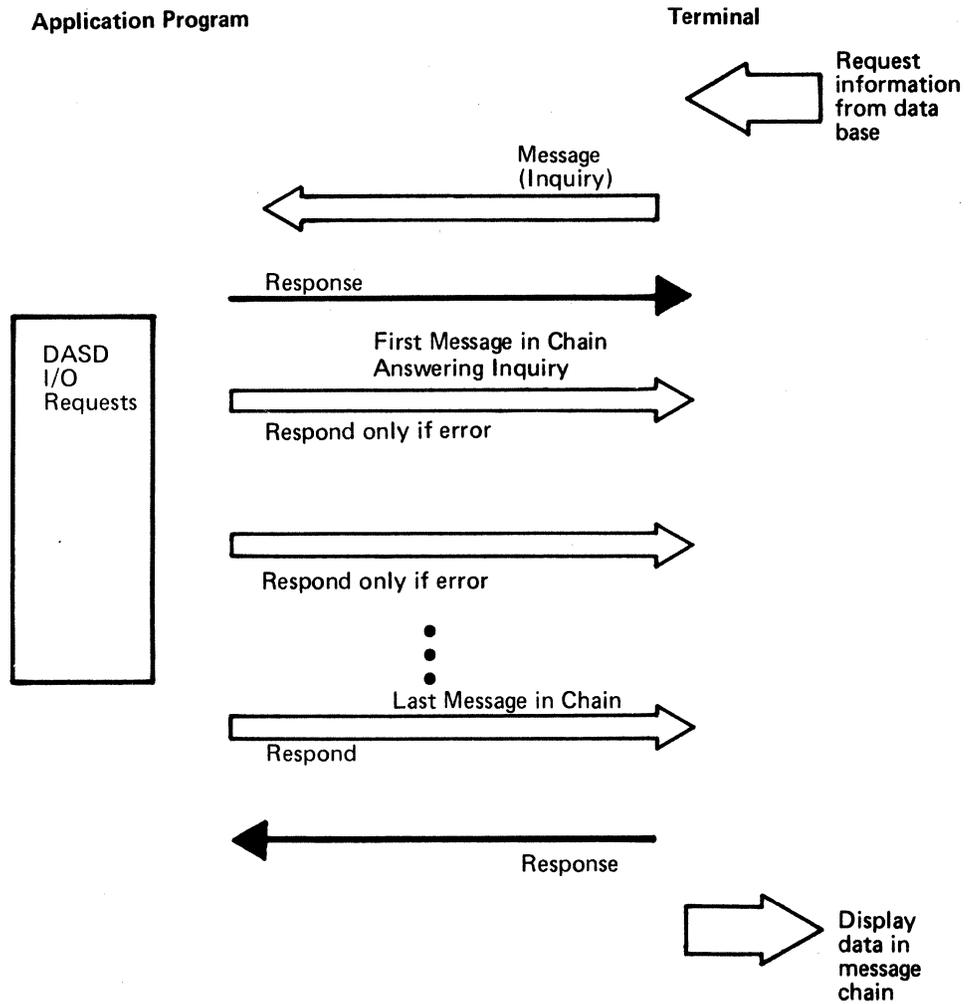


Figure 5-8. An Example of Message Chaining

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 obtain the completion status information. That is why the application program in Figure 5-9 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 sender of the message becomes aware of the error condition when the receiver 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

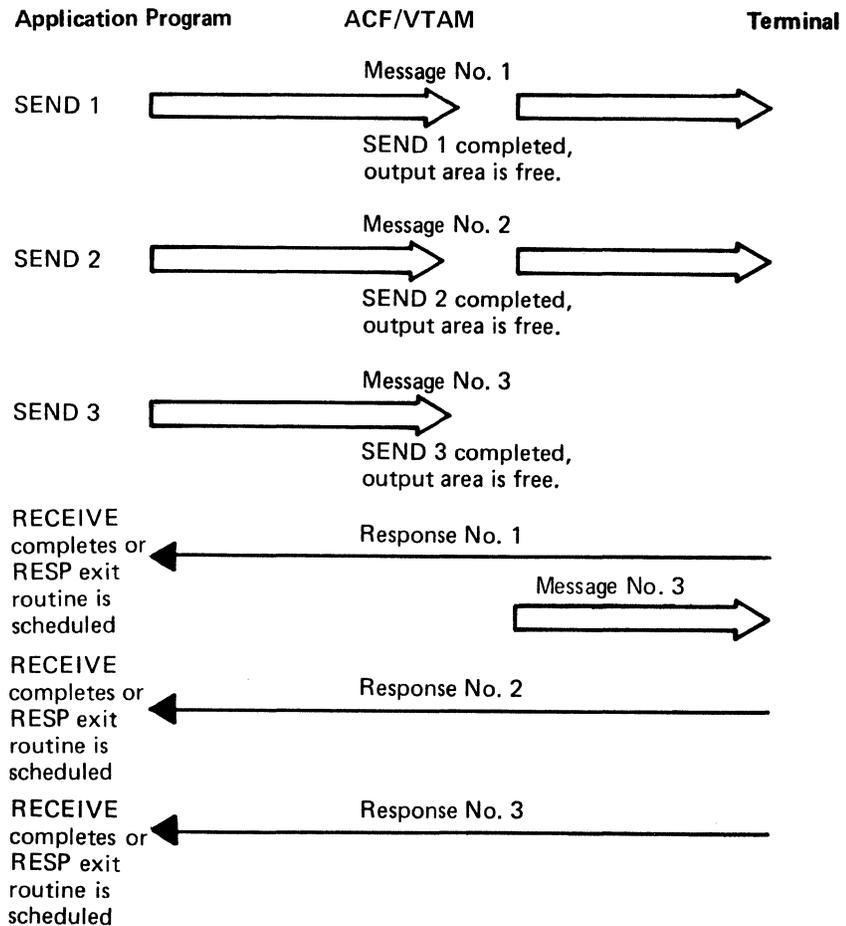


Figure 5-9. Scheduled Output

Normal-flow 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, ACF/VTAM provides two priorities for messages. Data messages and responses to data messages are always treated as normal-flow messages, although responses are presented before data messages. Certain data-flow control information is also treated as messages. 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 data-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. Instead, it must be presented to receiver as soon as possible; consequently, it is expedited by being sent before any normal-flow traffic.

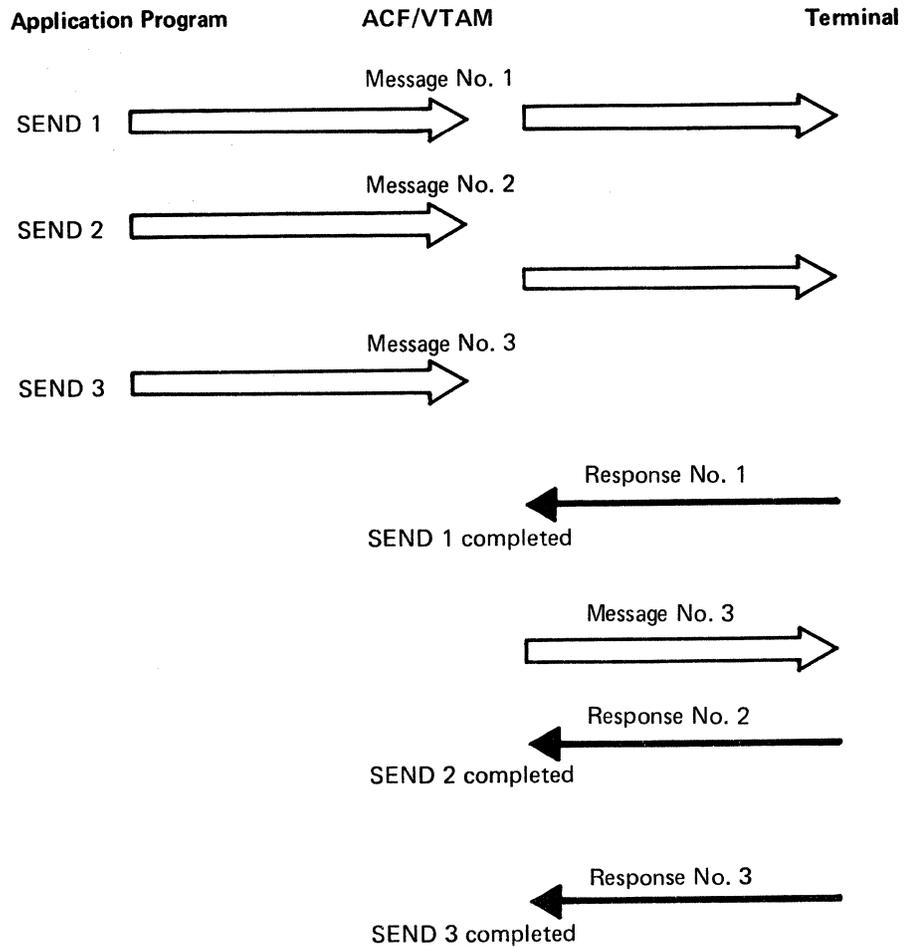


Figure 5-10. Responded Output

Figure 5-11 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.

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

- Expedited-flow messages (DFASY exit routine)
- Responses (RESP exit routine)
- Session control commands (SCIP exit routine)

The application program can have more than one of each of these three types of exit routines. In fact, a program can specify a particular DFASY, RESP, and SCIP exit routine for each connection. This is not required, however; if the program wants, it can use the same DFASY, RESP, and SCIP exit routines for all connections. A program can have only one of each of the other types of EXLST exit routines.

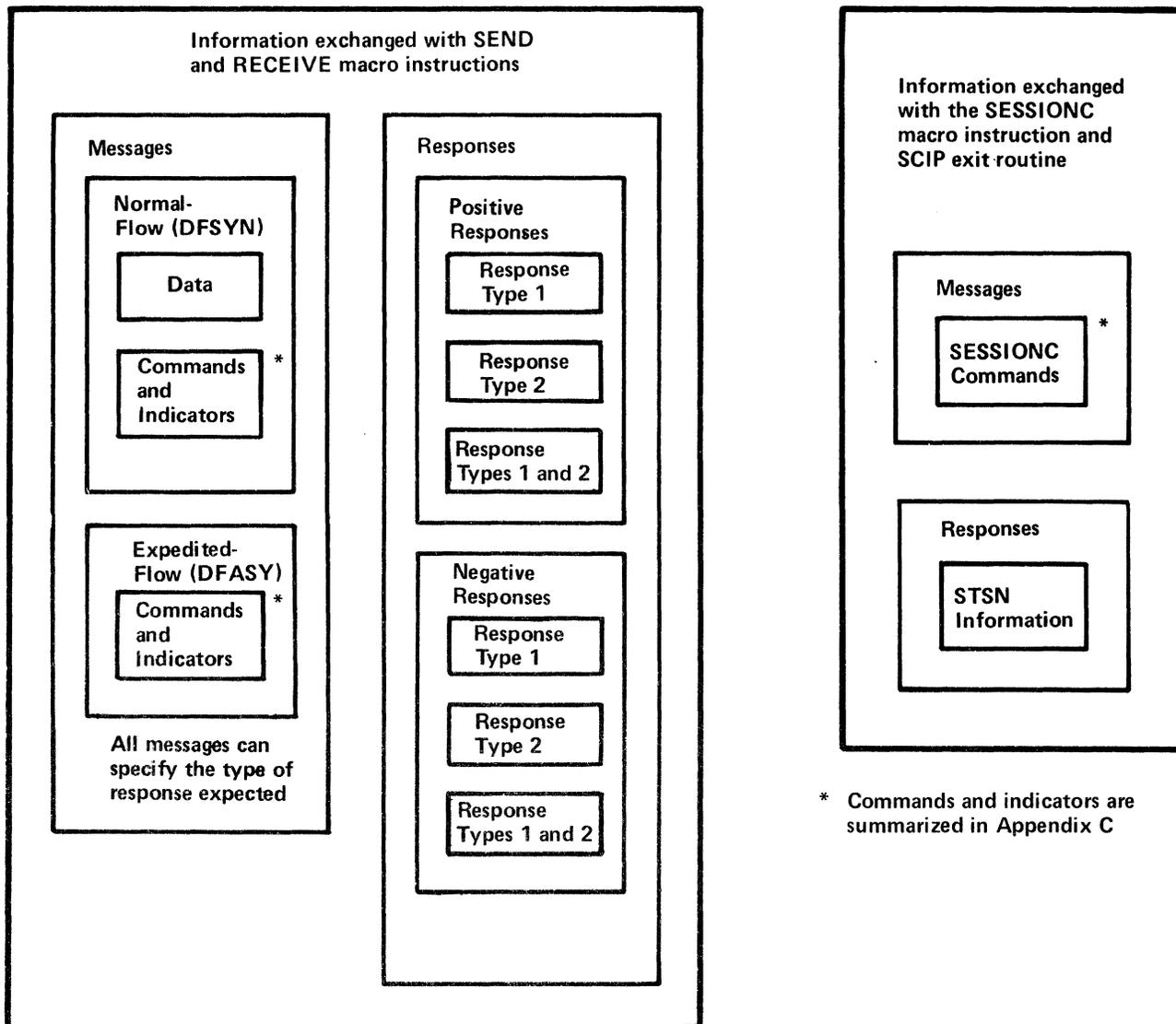


Figure 5-11. Types of Information Exchanged between an Application Program and Terminals

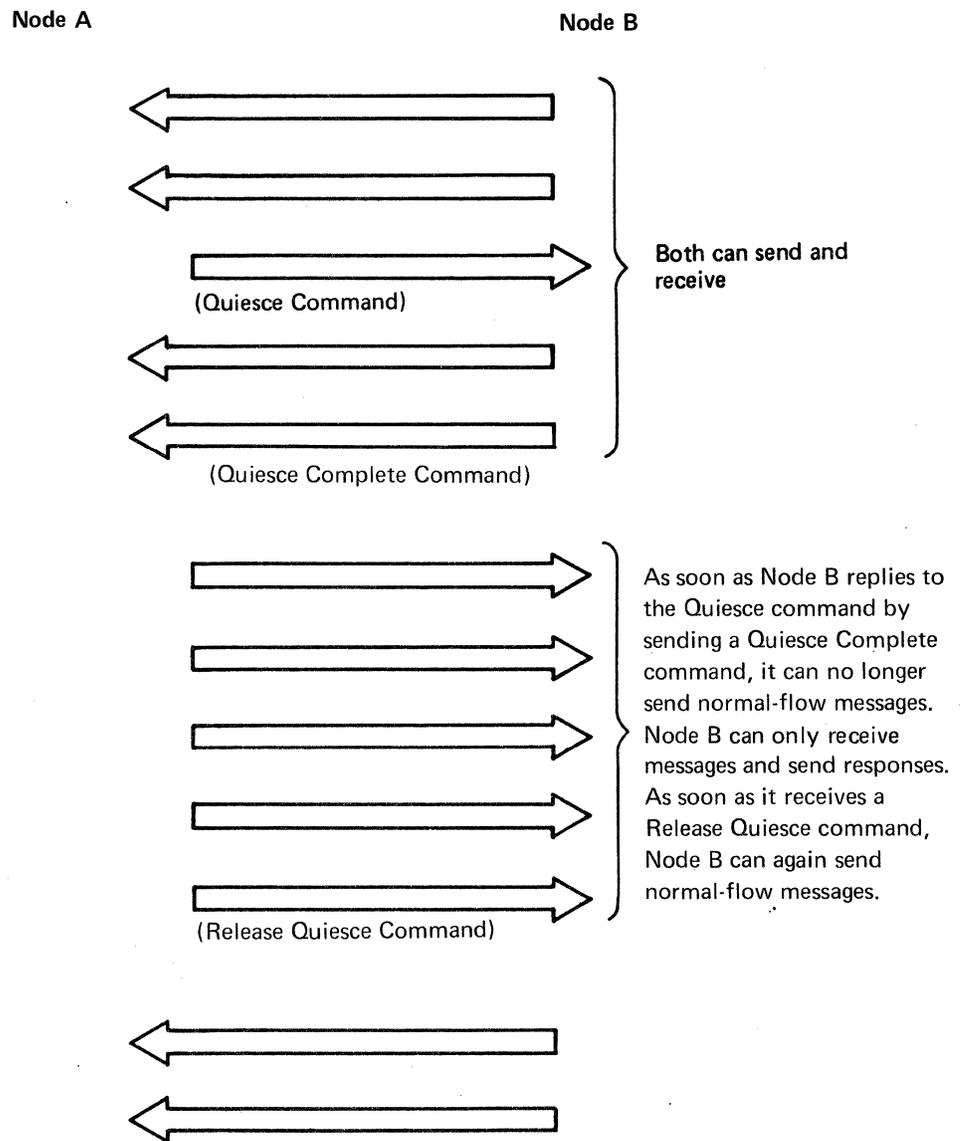
A program does not have to use exit routines to handle expedited-flow messages, responses, and session control commands. Instead, it can use RECEIVE macro instructions to receive these kinds of input, specifying the type of input it wants by including a special parameter in the receive command. Note also that when responded output is specified in a SEND macro instruction, the response information is available in the RPL when the operation is completed.

Quiesce, Change-Direction, and Bracket Protocols

ACF/VTAM allows an application program to send and receive messages simultaneously. 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 sender and receiver of messages 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*. In the following descriptions of these protocols, one end of the session is referred to as *Node A* and the other end of the session is referred to as *Node B*. Either node can be the primary end of the session.

Quiesce Protocol: Node A informs Node B 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 Node B replies, by sending a *Quiesce Complete (QC) command*, it must stop sending and prepare to receive. Node B cannot send normal-flow messages until it receives a *Release Quiesce (RELQ) command* from Node A, as shown in Figure 5-12. While Node B is quiesced, however, it can send responses and expedited-flow commands. ACF/VTAM does not enforce quiesce protocol; it is the user's responsibility to conform to the convention.

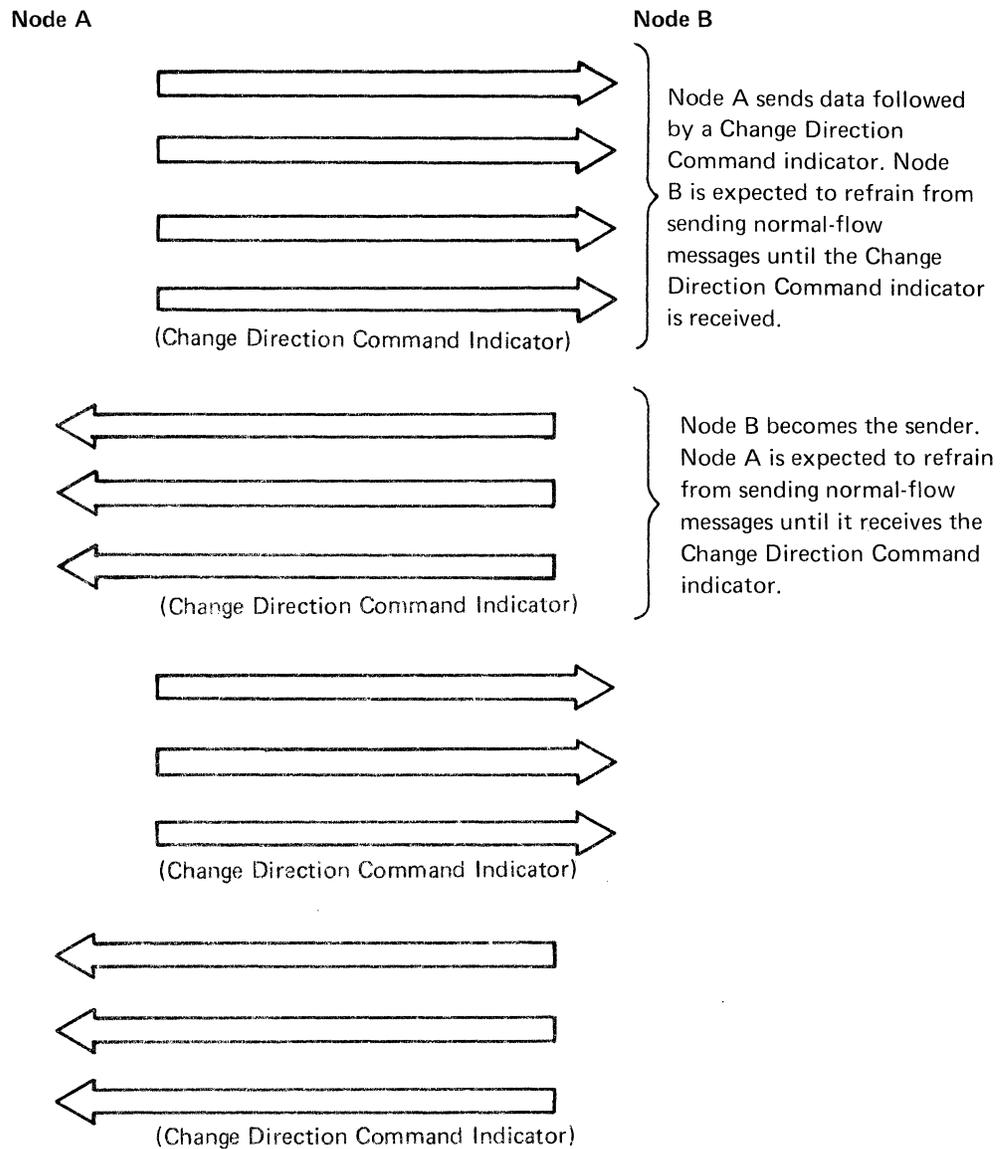


Note: Responses are not shown.

Figure 5-12. An Example of Quiesce Protocol

Change-Direction Protocol: When Node A has finished sending data, it sends a *Change Direction Command indicator* to Node B. Node B 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 5-13.

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



Note: Responses are not shown.

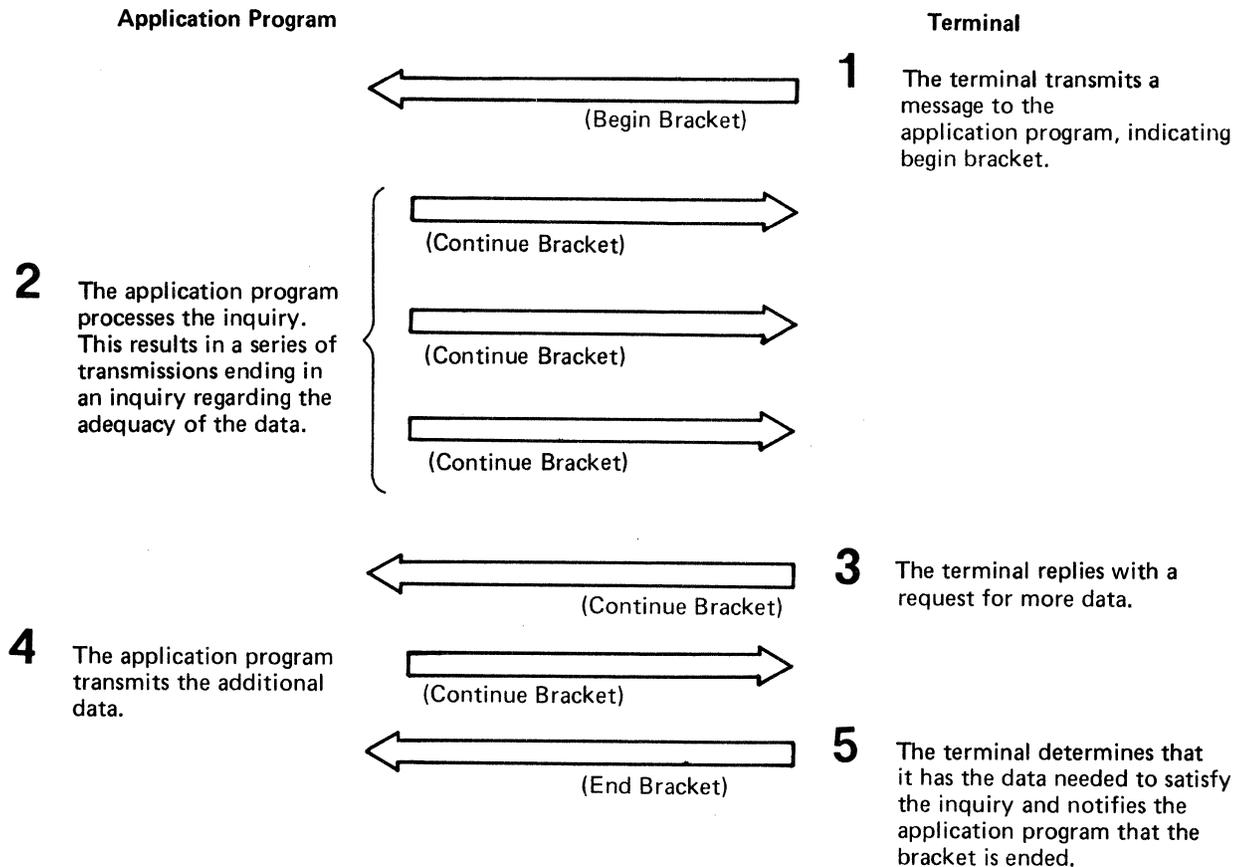
Figure 5-13. An Example of Change-Direction Protocol

Note: The Change Direction Request indicator is a non-SNA indicator supported by ACF/VTAM and certain logical units. It is recommended that this indicator not be used, because it is incompatible with SNA. SNA uses the Signal command to request a Change Direction command indicator.

Change-direction protocol is not enforced by ACF/VTAM. Should the node waiting for a Change Direction Command indicator begin sending data anyway, ACF/VTAM does not prevent the transmission of the data. The successful use of this method of communication rests on the assumption that neither end of the session violates the "rules."

Bracket Protocol: A *bracket* is any unit of work defined by the user and understood by both ends of a session. 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 start 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 5-14 illustrates a use of bracket protocol.



Note: In this example, the terminal determines the beginning and the end of the bracket. In other applications, the application program could determine the beginning and the end of bracket, or one node could determine the beginning and the other node determine the end.

Figure 5-14. An Example 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 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. The session parameters agreed to when connection is established specify which node must use a Bid command before initiating a bracket.

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

Figure 5-15 lists the three sets of indicators and commands discussed above.

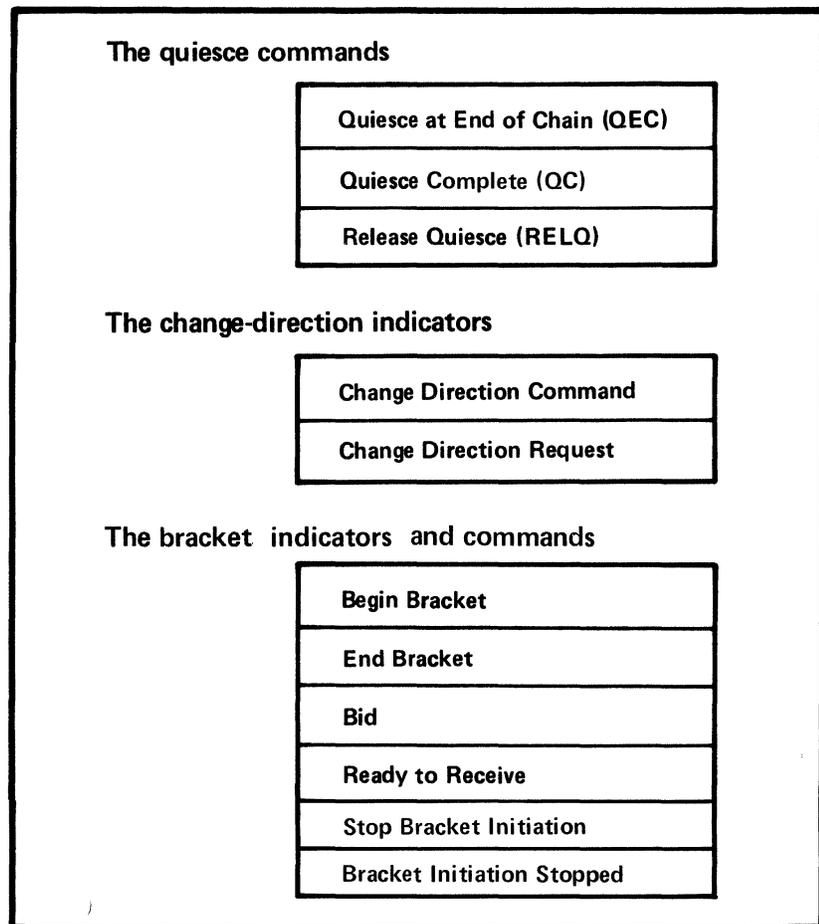


Figure 5-15. Commands and Indicators Used to Direct the Flow of Data

Request and Response Modes

When session parameters are sent to a terminal as part of the connection process, certain combinations of protocol bits establish certain request and response modes. Essentially, the bits indicate whether chaining will be permitted, how often a response will be requested, and in what order the responses must be returned.

There are two modes in which the senders of messages can operate: immediate control mode and delayed control mode.

Immediate Control Mode: When operating in this mode, the sender sends only single-element messages (that is, cannot send a chain), and the sender requests a definite response to each single-element message. After sending each message, the sender must wait for a response before sending the next message.

Delayed Control Mode: When operating in delayed control mode, the sender may send single-element messages and may also send multiple-element messages (chains). There are two forms of delayed control mode:

Immediate request mode: The distinguishing characteristic of this mode is that the sender may send a series of elements constituting one or more complete chains and ask for a definite response only in the last element in the series. In addition, once the sender has requested a definite response, it sends no other element until it receives the definite response.

Thus, if the sender is sending a series of single-element messages, only the last single-element message requests a definite response; the other single-element messages request an exception response only. If the sender is sending a chain, only the last element in the chain requests a definite response. If the sender is sending multiple chains, only the last element in the last chain requests a definite response; all preceding elements ask for an exception response only.

Delayed request mode: The distinguishing characteristic of this mode is that, while the sender may insert requests for definite responses into the series of elements it is sending, it is not required to wait for any of those responses. This mode can be used to send multiple chains, with a definite response requested in the last element of each chain (all other elements in each chain request an exception response only), and the sender can send any number of chains before stopping to wait for responses.

The receiver may be in either of two modes:

Immediate response mode: The receiver sends responses in the same order as the sender requested them. Thus, when the sender receives a response, it can infer that the receiver has received all preceding elements and that no negative responses will be forthcoming for those preceding elements.

Delayed response mode: The receiver need not return responses in the same order as they were requested. A response for one element may be delayed beyond the response for a subsequent element. (There is one restriction, however, on the receiver. The receiver must send responses for elements preceding a Chase command before it sends the response to the Chase command.)

Additional Protocols

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

Whether or not extraneous blank characters can be removed 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

Basic-Mode and Record-Mode Communication

Identification of application programs, logical units, and non-SNA 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 Application Programs, Logical Units, and Non-SNA Terminals

When defining the network to ACF/VTAM, the user assigns a symbolic name (1 to 8 bytes) to each application program, logical unit, or non-SNA terminal. Before a primary application program establishes connection with a terminal, or a secondary application program establishes connection with a primary application program, it uses the symbolic name to refer to the terminal or primary application program. When connection is established, ACF/VTAM assigns a session identification (called a *communication identifier*, or CID) that is used for all subsequent specific-mode I/O requests.

When a RECEIVE or READ macro instruction issued in any-mode is completed, ACF/VTAM provides the CID of the sender. Should the application program need to relate the CID to the sender’s symbolic name it can:

Ask ACF/VTAM to translate the CID into the symbolic name using the INQUIRE macro instruction.

Maintain a table of CIDs and their equivalent symbolic names.

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

Specific-Mode and Any-Mode

An application program can request data from a *specific* logical unit, application program, or non-SNA terminal, or it can request data from *any* one of its connected logical units, application programs, or non-SNA terminals. The application program designates the desired mode—specific or any—with each READ, SOLICIT, or RECEIVE macro instruction.

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

In any-mode, the application program does not know the identity of the source of the data until the data has been moved into its input area and the READ, SOLICIT, or RECEIVE has been completed. Because the source of the data 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 logical units, application programs, or non-SNA terminals rather than using a separate input area for each of them.

With specific-mode, the application program must specify the identity of the logical unit, application program, or non-SNA terminal from which the data is desired. 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 5-16.

Application Program

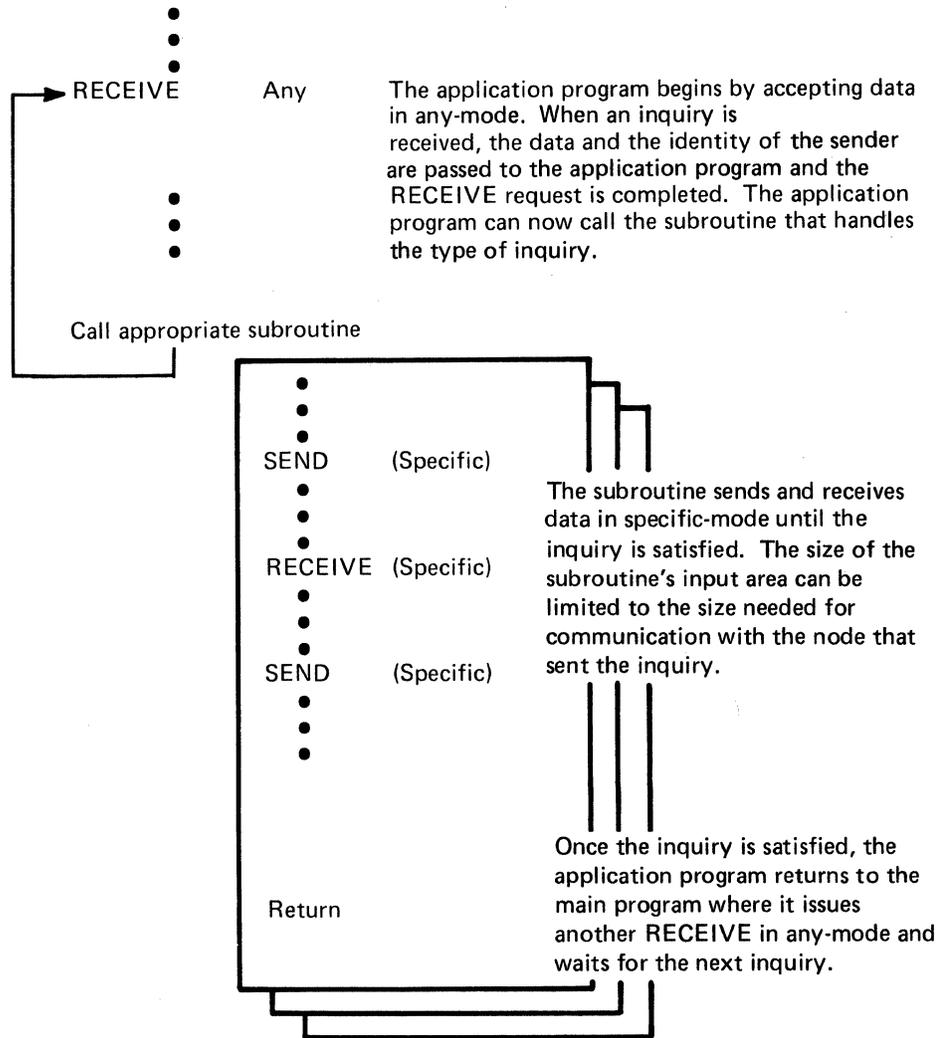


Figure 5-16. Using a Combination of Any-Mode and Specific-Mode

Continue-Any and Continue-Specific Modes

In the example of Figure 5-16, 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 logical units, application programs, or non-SNA 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, ACF/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 logical unit, application program, or non-SNA terminal (called *continue-any* mode) and when a READ, SOLICIT, or RECEIVE macro instruction issued in specific-mode must receive the data (called *continue-specific* mode). These modes are designated when an I/O request is issued, but do not become effective until the I/O operation is completed. Alternatively, a logical unit, application program, or non-SNA terminal can, at connection, be designated as always in either continue-any or continue-specific mode. If a logical unit, application program, or non-SNA 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 logical units, application programs, and non-SNA 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 5-17 illustrates how the modes described above are related. Although Figure 5-17 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. ACF/VTAM provides two ways of handling data that is too large for the input area:

ACF/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.

ACF/VTAM can keep the data. ACF/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 establishing connection, but it can override the option for a given I/O request.

Application Program

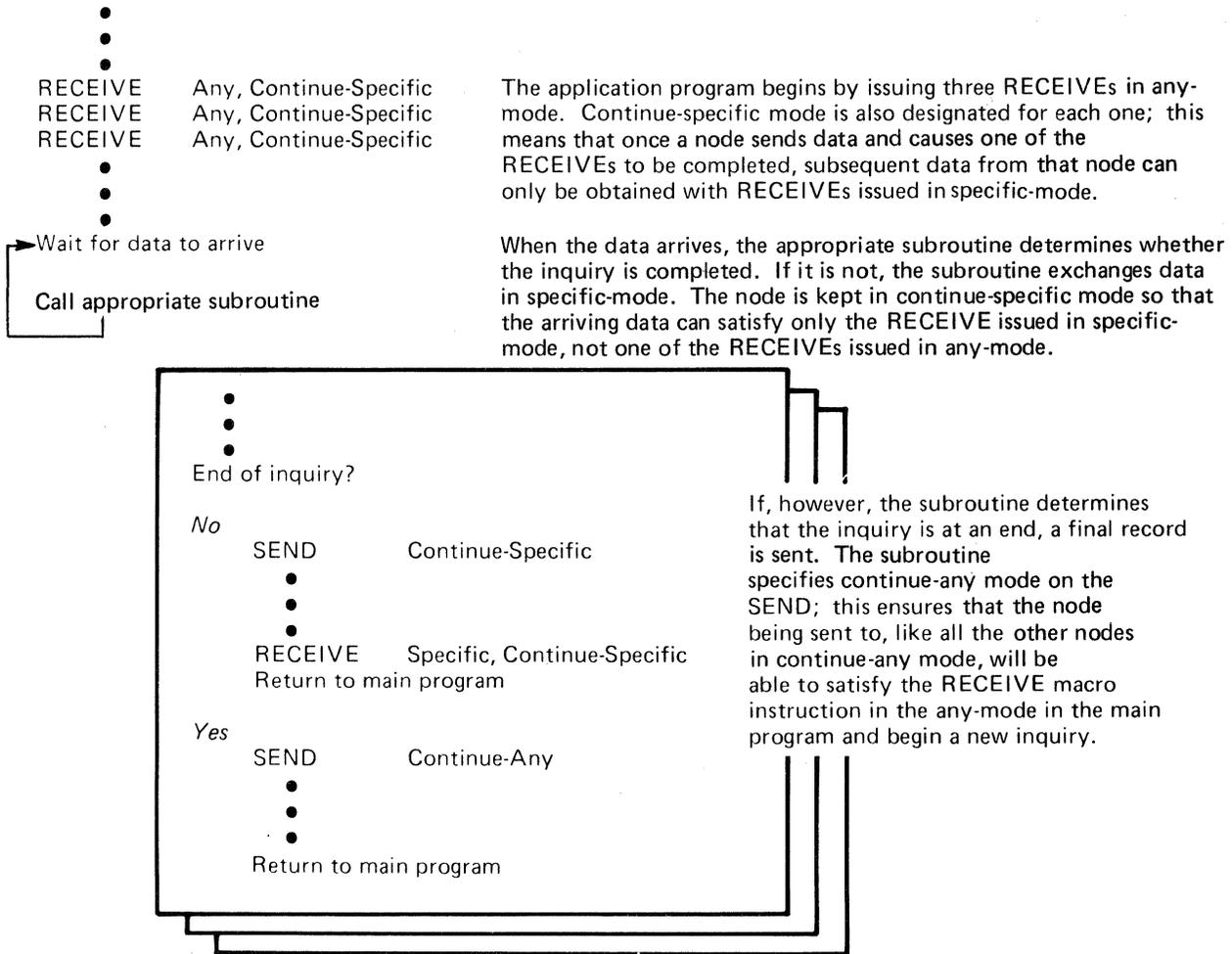


Figure 5-17. Using the Continue-Any and Continue-Specific Modes to Handle Concurrent Inquiries

Chapter 6. Reliability, Availability, and Serviceability

This chapter describes how ACF/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 ACF/VTAM debugging aids, dump programs, and test programs used to correct programming errors and machine malfunctions.

ACF/VTAM's RAS Strategy

ACF/VTAM's reliability, availability, and serviceability (RAS) aids are intended to minimize the effect of programming errors and machine malfunctions on the network. ACF/VTAM attempts to detect problems and handle them before they become serious. If an error is encountered, ACF/VTAM attempts recovery and records the error. If recovery is not possible, ACF/VTAM automatically provides error records and dumps to help identify the problem and its cause. For additional help in debugging, the user can request traces and testing services from ACF/VTAM. If possible, ACF/VTAM avoids terminating application programs or closing down the data communication 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. ACF/VTAM handles both hardware and software errors; software errors include those of users (application programs and the network operator), of the operating system, and of ACF/VTAM itself.

Error-recovery and error-recording operations are invoked automatically. ACF/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:

- Saves all console logs that pertain to the error message. These logs probably reflect all of ACF/VTAM's actions from ACF/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 deactivating part of the data communication 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 ACF/VTAM or its data communication 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: ACF/VTAM's network operator commands and RAS facilities can help re-create the problem and collect pertinent data.

ACF/VTAM's RAS Facilities

ACF/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 describe ACF/VTAM's RAS facilities.

Serviceability Aids

ACF/VTAM's serviceability aids assist in determining the cause of problems in the data communication system. Serviceability aids include:

Error Recording

Hardware Errors: ACF/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): ACF/VTAM uses SYS1.LOGREC to retain records of software errors that result in the abnormal termination of ACF/VTAM.

Traces

DOS/VS: The problem determination and serviceability aids (PDAIDS) can be used to monitor ACF/VTAM's SVCs, I/O operations, and internal storage management.

OS/VS: The generalized trace facility (GTF) can be used to monitor ACF/VTAM's SVCs, I/O operations, and task management and to trace events in application programs using ACF/VTAM.

ACF/VTAM: ACF/VTAM traces can be used to monitor I/O activity, to record the contents of ACF/VTAM buffers, and, in OS/VS, to trace ACF/VTAM storage management. In addition, ACF/VTAM internal traces can be used to record internal processing events related to locking and unlocking, application program interface, process scheduling services, and storage management services, which are all internal components of ACF/VTAM.

Dumps

OS/VS: System dump programs are tailored by ACF/VTAM to provide formatted dumps of some ACF/VTAM control blocks and in some cases to provide additional diagnostic information.

DOS/VS: ACF/VTAM provides no dump facilities for DOS/VS. However, system dump programs can be used to dump some ACF/VTAM control blocks and data areas.

NCP: ACF/VTAM has tailored the NCP dump utility program. This utility program can be invoked using ACF/VTAM's MODIFY command, or, optionally, it can be invoked as part of ACF/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 an ACF/VTAM system.

These aids are discussed in detail below.

Error Recording

ACF/VTAM provides facilities for recording hardware errors and software errors.

Recording Hardware Errors: Recording hardware errors is 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 ACF/VTAM is placed in the error-record data set of the operating system in which the affected device is defined. The information in this data set can be formatted and printed by each system’s environmental recording, editing, and printing (EREP) program.

ACF/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, ACF/VTAM maintains two counters for each local device in its domain. One counter keeps track of the number of Start I/O (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 statistics 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 whenever there is an error from 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 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 ACF/VTAM deactivates a device. Records for 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 ACF/VTAM must terminate processing, ACF/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.

An ACF/VTAM software error record includes a description of the error and an audit trail. An ACF/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 ACF/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

ACF/VTAM under both DOS/VS and OS/VS has operating system traces and ACF/VTAM traces. The operating system traces provide the highest level of information for 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 ACF/VTAM or to an application program using ACF/VTAM, the ACF/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 ACF/VTAM. The data collected by these traces can be used as a chronology of ACF/VTAM activity and reflects the conditions in ACF/VTAM during errors. System traces for ACF/VTAM are started and stopped by using the facilities of the operating system.

DOS/VS trace records for ACF/VTAM are printed with the PDLIST program. OS/VS1 trace records for ACF/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 ACF/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.

ACF/VTAM Activity Traces: The following types of ACF/VTAM activity traces are provided:

I/O traces: To trace I/O activity between ACF/VTAM and local communications controllers and between two application programs in the same domain.

Buffer traces: To record the contents of ACF/VTAM buffers as data enters and leaves ACF/VTAM.

NCP line traces: To record the NCP trace records of specified lines.

Storage management traces: To record the use of ACF/VTAM storage.

ACF/VTAM activity traces are initiated and terminated by ACF/VTAM start options or the MODIFY command.

The ACF/VTAM activity trace records are written on a particular data set in each operating system. In a multiple domain network, the activity trace records are written in the domain from which the trace was started. For DOS/VS, the file name of this data set is TRFILE. For DOS/VS, ACF/VTAM writes the ACF/VTAM activity 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, ACF/VTAM passes the trace data to GTF, which records it in SYS1.TRACE. Because ACF/VTAM activity traces use GTF to record the trace records, GTF must be active before ACF/VTAM data can be recorded. If GTF is not active, ACF/VTAM does not provide a record of the event to be traced.

To print activity trace records, ACF/VTAM provides a utility program for DOS/VS and uses a system utility program for OS/VS. The ACF/VTAM trace-print utility program for DOS/VS selectively edits and prints the data collected by the various ACF/VTAM traces. By using the MODIFY command or by executing a job step, the network operator starts the print utility program and specifies the nodes for which trace records are to be printed. In OS/VS1, ACF/VTAM traces are printed by using the operating system's HMDPRDMP service aid. In OS/VS2, ACF/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.

ACF/VTAM Internal Event Traces: ACF/VTAM also provides a set of traces to record internal processing events. These traces can be used to record events in the following categories:

Application program interface: To record each occurrence of an internal TPIO request, each posting of a user's ECB, each exit to an RPL exit routine, and each exit to an EXLST exit routine.

Locking and unlocking: To record each occurrence of an internal TPLOCK (shared), TPLOCK (exclusive), TPUNLOCK, and TPUNLOCK ALL request.

Process scheduling services: To record each occurrence of an internal TPQUE, TPQUE RESET, TPSCHED, DISPATCH, IRB DISPATCH, TPOST, TPWAIT, and TPEXIT request.

Storage management services: To record each occurrence of an internal REQSTORE, queued REQSTORE, RELSTORE, ABEND RELSTORE, GETSTORE, and FREE-STORE request.

ACF/VTAM internal event traces are started and stopped with the MODIFY command. The internal traces can be started and stopped individually or in combinations, or all traces can be started and stopped at the same time.

In DOS/VS, the trace records are written to a trace table in pageable storage. In OS/VS, the records can be written either to an internal trace table in pageable storage or to a trace table on an auxiliary storage device. When the trace records are written to an internal trace table, the user specifies the number of pages of storage to be allocated to the trace table. When the trace table is filled, ACF/VTAM begins again to record records at the beginning of the trace table, overwriting records that the table already contains.

For more information on the ACF/VTAM internal traces, see the *ACF/VTAM Network Operating Procedures* and *ACF/VTAM Debugging Guide* for the operating system being used at the installation.

Dumps

ACF/VTAM provides no formatted dump facilities in DOS/VS. ACF/VTAM provides routines to augment both the OS/VS dump facilities and the NCP dump facilities. The text below describes these routines.

Operating System Dumps (OS/VS only): ACF/VTAM provides routines that format portions of OS/VS dumps. When the operating system is dumping ACF/VTAM or an application program that is using ACF/VTAM, it invokes these routines to locate and format key ACF/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 ACF/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 ACF/VTAM itself, from the abnormal termination of an application program that is using ACF/VTAM, or from a SNAP macro instruction in an application program. For dumps produced by the PRDMP service aid, however, ACF/VTAM formatting is optional; the option must be specified as part of the input to the service aid.

Dumps of ACF/VTAM include an audit trail and unique identifiers for each ACF/VTAM module and control block. The audit trail is like that recorded for software errors.

NCP Dumps: ACF/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. For shared NCPs, a network operator in any domain that shares the NCP can request a dump of the NCP; however, the dumping process overlays part of the NCP, and the NCP becomes inoperative. To restore operations, a new NCP must be loaded.

Teleprocessing Online Test Executive Program (TOLTEP)

TOLTEP is an ACF/VTAM component that controls the selection, loading, and execution of online tests (OLTs) within the data communication 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 domain.

TOLTEP allows multiple OLTs to be run concurrently with application programs in the ACF/VTAM data communication system. TOLTEP supports testing of devices attached on start-stop or BSC lines, local non-SNA 3270 terminals, and certain SDLC devices. 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 ACF/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 ACF/VTAM is started. If TOLTEP is abnormally terminated prior to ACF/VTAM termination, ACF/VTAM automatically attempts to restart it.

To run TOLTEP, a terminal or the system console 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 recommended 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. 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 connected to an application program, the connection must be broken before the terminal can be made available to TOLTEP.

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 names of the test terminals, the tests to be run, and various options. 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. 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 the *ACF/VTAM TOLTEP* manual, which is identified in full in Figure P-1 in the Preface.

Reliability and Availability Support

The purpose of ACF/VTAM's reliability and availability support is to maintain the operation of the network. This support attempts to prevent problems and, if that is not possible, to minimize the impact of the problems. ACF/VTAM's reliability and availability support includes:

Error Detection and Feedback: Before attempting to act upon any request, ACF/VTAM analyzes it for erroneous information. If an error is detected, ACF/VTAM returns the request along with an indication of the error.

3704/3705 Initial Test: ACF/VTAM allows a communication 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: ACF/VTAM controls the allocation of much of the storage required for its operation. Using this control, ACF/VTAM permits the queuing of requests and attempts to avoid storage interlocking conditions. (A storage interlocking condition is one in which so much of ACF/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, ACF/VTAM attempts to recover from some errors. If recovery is not possible, ACF/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: ACF/VTAM tries to recover from some errors. If recovery is not possible, ACF/VTAM first attempts to isolate the problem and continue processing. If ACF/VTAM cannot continue processing, it attempts an orderly closedown of the data communication system so as not to affect other jobs in the host system.

NCP Slowdown: ACF/VTAM recognizes NCP slowdown and helps the NCP to return to normal operations.

Switched Network Backup: ACF/VTAM allows the user to define a switched line as a backup for a nonswitched line between a communications controller and a physical unit.

Resource Takeover: In a multiple-domain network, if ACF/VTAM or its host computer fails, its resources can be taken over by an adjacent domain.

Configuration Restart: If an error occurs in the data communication system, ACF/VTAM attempts to restore the system to its prefailure status. If it cannot, ACF/VTAM provides facilities that allow the user to restore the network to either its initial or prefailure status. ACF/VTAM also allows manual switched backup for the CPU and communications controllers.

Lost Subarea Notification: When a subarea (a host computer or a communications controller) is lost, the unit or units connected to the lost subarea report the loss to each adjacent unit, which relays the report to other units. This action makes it possible for affected units to send appropriate error indications to all elements in the network that were affected by the loss.

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, ACF/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, ACF/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, ACF/VTAM sends an error message to the operator.

If both the syntax and the function are validated but ACF/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, ACF/VTAM transmits messages to the console describing any unusual or error conditions that affect the operation of the data communication system, such as permanent hardware errors and the abnormal termination by ACF/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, ACF/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. ACF/VTAM has return codes to help isolate the reason for the error. ACF/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.

3704/3705 Initial Test

ACF/VTAM's 3704/3705 initial test program can be used to test a 3704 or 3705 Communications Controller before an NCP is loaded. The program tests the controller's circuitry to determine whether it is operating correctly. This initial testing is performed automatically for remote communications controllers; it is optional for local communications controllers. If testing is specified in the PCCU definition statement, ACF/VTAM automatically invokes the test program before loading an NCP into any controller. If testing is not specified, initial testing is still performed for a remote controller but is not performed for a local controller.

Similarly, when the independent loader utility program of the operating system (DOS/VS or OS/VS) is used to load a local communications controller, the user specifies whether the communications controller is to be tested before the NCP is loaded. If testing is specified, an initial test routine is sent to the controller and executed before the loading is done. If the initial test routine detects no malfunction, loading proceeds; if the routine detects trouble, an error message is issued and no loading is done. For more information on the independent loader utility program and its initial test routine, see the *NCP Generation* manual.

Storage Management

ACF/VTAM uses a number of buffer pools, as noted in "VTAM Buffering" in Chapter 3. The user can specify the size and characteristics of each pool, or the user can allow IBM-provided values contained in ACF/VTAM to determine the size of each pool. ACF/

VTAM uses the buffer pools for various purposes during its internal processing, including use of the pools for control blocks, message buffers, and channel programs.

The parameters supplied when ACF/VTAM is started (which are among the various ACF/VTAM start options) provide for a basic allocation to each ACF/VTAM buffer pool. This basic allocation controls the size of the buffers in each pool, the number of buffers to be allocated initially in each pool, and the point at which slowdown processing for each pool is to occur. Slowdown processing, which occurs when only a specified number of unused buffers remain in pools, means that buffers will be allocated from the pool only in response to priority requests from ACF/VTAM; nonpriority (or normal) requests for buffers are queued or rejected. Slowdown processing ends when the number of available buffers in the pool increases above the slowdown point specified by the user.

In addition to specifying a basic allocation to each pool, the user can specify values for dynamic expansion of each pool. The dynamic expansion parameters indicate that the pool is to be expanded by a certain number of buffers each time the number of available buffers in the pool falls below a certain number. When dynamic expansion has been specified, each time the pool approaches depletion, the pool is enlarged by the specified number of buffers. When use of the pool decreases and all buffers that were allocated in one dynamic expansion are no longer in use, the buffer pool is contracted to a smaller size. If usage of the pool decreases enough, the pool is contracted to the size of its basic allocation, but never any smaller than the basic allocation.

By specifying dynamic expansion of pools, the user can reduce the amount of storage that must be permanently allocated to the pools (only the basic allocation is permanently reserved for each pool). The user does not have to specify a basic allocation that will meet the greatest possible demand on the pool. Instead, by specifying dynamic expansion, the user can allow the pool to be expanded as the demand increases and can allow the pool to contract as demand decreases.

For basic-mode operations, ACF/VTAM provides the user with another means of controlling allocation of buffers. The BUFFACT operand in an APPL definition statement (which defines an application program) and the BUFLIM operand in a terminal's definition statement determine the maximum number of buffers from the pageable data buffer pool (PPBUF) that can be used to hold data that ACF/VTAM has read from the terminal, but has not yet been transferred into the application program's buffers. Judicious use of these operands allows the user to prevent certain program-terminal sessions from monopolizing the pageable buffer pool.

In addition to managing storage by controlling its allocation, ACF/VTAM attempts to protect its storage from unauthorized access. ACF/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

ACF/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 ACF/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 ACF/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 ACF/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 ACF/VTAM itself. If it is a user error, it is handled as explained in "Error Detection and Feedback" earlier in this chapter.

If ACF/VTAM cannot determine whether the problem is a user error, it attempts to recover from the error. If recovery is not possible, ACF/VTAM attempts to isolate the cause of the problem to a single application program and abnormally terminate that program. If the problem cannot be isolated, ACF/VTAM abnormally terminates.

NCP Slowdown

If data communication 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. ACF/VTAM recognizes an NCP slowdown, and, to facilitate NCP recovery, ACF/VTAM stops scheduling input and output requests for that NCP, although ACF/VTAM does continue to read from the NCP. During this slowdown processing, ACF/VTAM continues to accept requests from application programs, but it queues those requests that are directed to nodes in the network accessed through 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.

Switched Network Backup

ACF/VTAM allows the user to define a physical unit and its associated logical units so that the user can use a switched line to back up a nonswitched line. To use this capability, the user defines the physical unit and its logical units as part of an NCP major node and, using the same logical unit names and a different physical unit name, as part of a switched SNA major node. If the physical unit and its logical units are active as part of the NCP major node and the nonswitched line fails, the network operator can release the physical unit and its logical units using the VARY command with the REL (release) option. The network operator can then activate the backup switched line and the switched SNA major node containing the physical unit and its logical units. If the problem with the original nonswitched line is corrected, the physical unit and its logical units can be returned to their original status as a nonswitched part of the NCP major node. To do this, the network operator uses the VARY command to deactivate the logical units, the physical unit, the switched line, and the switched SNA major node, and then uses the VARY command with the ACQ (acquire) option to reassign the physical unit to the NCP major node.

Resource Takeover

In a multiple-domain network, ownership of resources in one domain can be transferred to another domain. This is especially useful when the owning host system fails, but it can be performed when there is no host failure. The user can allow:

- A local communications controller and its resources in one domain to be transferred to an adjacent domain as a local communications controller with attached resources

- A local communications controller and its resources in one domain to be transferred to an adjacent domain as a remote communications controller with attached resources

Note: *The NCP has characteristics that differ depending on whether it is generated for use in a communications controller that is attached by channels to the host computer (local) or that is attached to a local communications controller by an SDLC link (remote). When ownership of a local communications controller is transferred without reloading the NCP, the communications controller retains most of the characteristics of a local communications controller.*

Transferring a Local Communications Controller as a Local Communications Controller: An operating local communications controller and its resources can be acquired by the network operator and the ACF/VTAM in an adjacent domain. (An *adjacent domain* is a domain that is connected to another domain by a cross-domain link or that shares a local communications controller.) In most cases, an adjacent domain acquires a local communications controller from another domain when the host computer in the other domain fails and cannot be restarted in a short period of time. Transfer of the communications controller to another domain allows the controller and its resources to continue operation until the controller's original host computer is restored.

Some preparation is necessary in anticipation of a possible acquisition of a local communications controller. The ACF/VTAM in the domain that is to acquire the controller must have available to it (in the ACF/VTAM definition library) an NCP definition deck that is compatible with, or is the same as, the definition deck for the NCP that is active in the communications controller immediately before acquisition. In addition, the network operators in other domains must be prepared to change cross-domain resource definitions or to modify the assignment of cross-domain resources to cross-domain resource managers in order to reflect the new ownership of the acquired resources after the transfer.

To acquire the communications controller (but not its physical units and logical units), the network operator in the adjacent domain issues a VARY command with the acquire (ACQ) option. This causes the ACF/VTAM in the domain in which the command is issued to take over control of the communications controller. The action of taking over control of the communications controller does not disrupt cross-domain sessions that are in progress. Optionally, the network operator can include the activation (ACT) operand in the VARY command; if included, this operand causes all lines and non-SNA devices attached to the communications controller to be activated on the basis of the ISTATUS operands specified in their definition statements.

The physical units and logical units attached to the acquired controller must be acquired and activated separately. Before acquiring any physical unit, the network operator deactivates cross-domain resource major nodes that define logical units (associated with the physical unit) as cross-domain resources. The operator does this because, after acquisition, the physical unit and logical units will be in the domain that is acquiring them and, therefore, will no longer be cross-domain resources. Then, to acquire each physical unit, the network operator in the acquiring domain issues a separate VARY ACQ command. When a physical unit is acquired, all logical units associated with the physical unit are automatically acquired. After a physical unit is acquired, it must be activated before it can be used. Before activating the physical unit, the operator may want to wait until the logical units associated with the physical unit have ended any cross-domain sessions (activation of an acquired physical unit disrupts cross-domain sessions). The network operator can activate an acquired unit in its new domain either by including the activate (ACT) operand in the acquiring command or by issuing a separate activate command. The network operator follows the above process to acquire and activate each physical unit associated with the acquired communications controller.

More than one adjacent domain can take over part of the controller's resources. To do this, each adjacent domain acquires the communications controller and then acquires only some of the physical units associated with the controller. Two domains cannot acquire the same physical unit; however, different domains can acquire different physical units on a multipoint line.

Figure 6-1 shows the effect of transferring a local communications controller to an adjacent domain as a local controller. In the figure, host computers 1, 2, and 3 are connected by SDLC links to form a three-domain network. The physical and logical units in domain 2 are involved in cross-domain sessions with application programs in host computers 1 and 3. When host computer 2 fails and will remain down for a significant period of time, the acquisition by domain 1 of NCP2 and its resources can follow this pattern, for example:

The network operator at host computer 2 makes a telephone call to the network operator at host computer 1 to tell that operator to take over NCP2 and its resources.

The network operator at host computer 1 issues a VARY ACQ command to acquire NCP2.

The network operators at host computers 1 and 3 then wait until cross-domain sessions involving the physical units at host computer 2 are ended. To do this, the network operators must be in telephone contact, because each operator can determine only by using DISPLAY commands that cross-domain sessions involving elements in his domain have ended.

When all cross-domain sessions involving PU1 have ended, the network operator at host computer 1 deactivates the cross-domain resource major node containing the cross-domain resource definition for LU1 and then issues a VARY ACQ,ACT command for PU1, causing that physical unit to be acquired and activated.

Similarly, when all cross-domain sessions involving PU2 have ended, the network operator at host computer 1 deactivates the cross-domain resource major node containing the cross-domain resource definitions for LU2 and LU3, and then issues a VARY ACQ,ACT command for PU2.

The network operator at host computer 3 issues a MODIFY command to change the assignments of LU1, LU2, and LU3 (as cross-domain resources) from the cross-domain resource manager in host computer 2 to the cross-domain resource manager in host computer 1.

After these actions, the logical units can communicate with application programs in host computer 1 as same-domain resources and with application programs in host computer 3 as cross-domain resources.

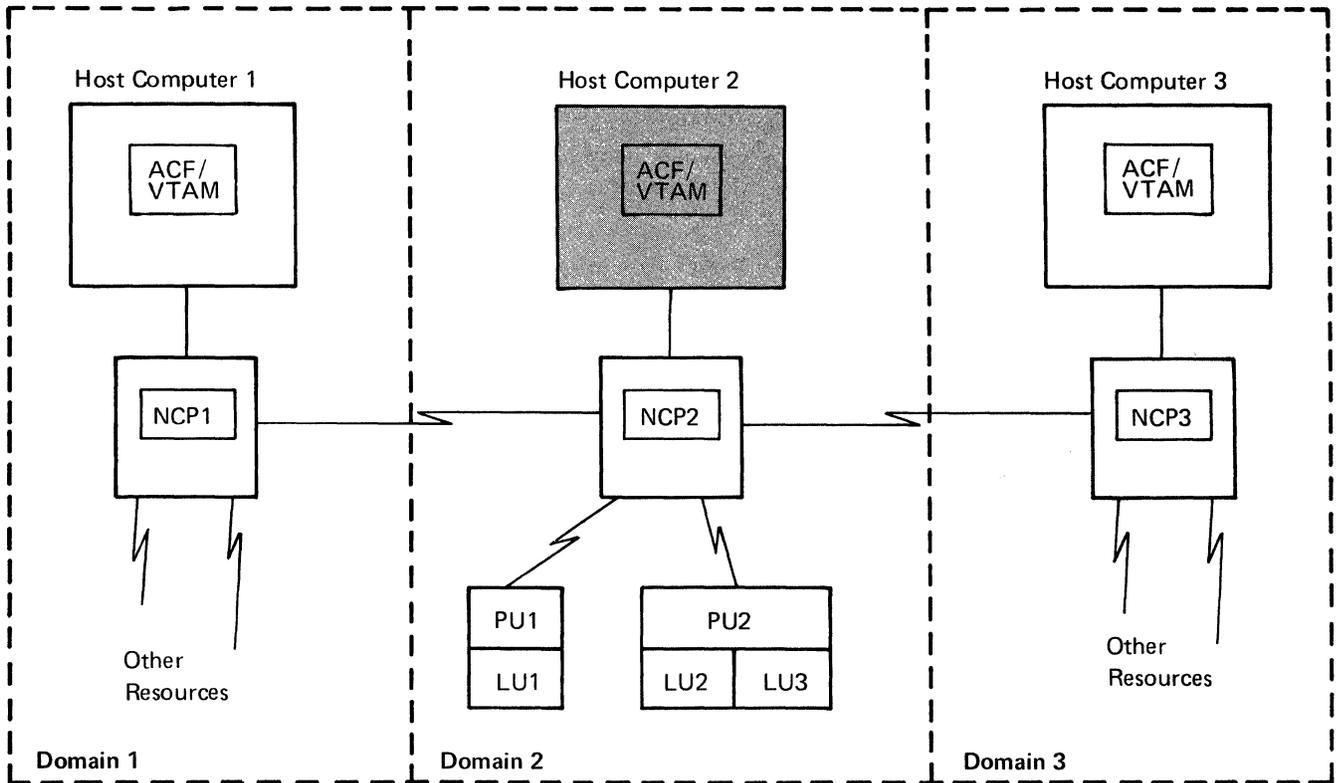
As implied by the above operations, to facilitate the deactivation and takeover of resources, it is advisable during ACF/VTAM definition to make a separate cross-domain resource major node for the cross-domain resource definitions associated with each physical unit in another domain. This enables the network operator to deactivate the definitions and to take over the resources on a physical-unit-by-physical-unit basis.

When operations at host computer 2 are restored, the network operator at that host computer can take back NCP2 and its resources. The sequence could be like this:

The network operator at host computer 2 issues a VARY ACT command to make NCP2 active in his domain. At this point, NCP2 would have control of only those resources that had not been acquired and activated by host computer 1.

The network operator at host computer 2 would then telephone the network operator at host computer 1 and tell him to release the acquired communications controller.

A. The Network before Takeover



B. The Network after Takeover

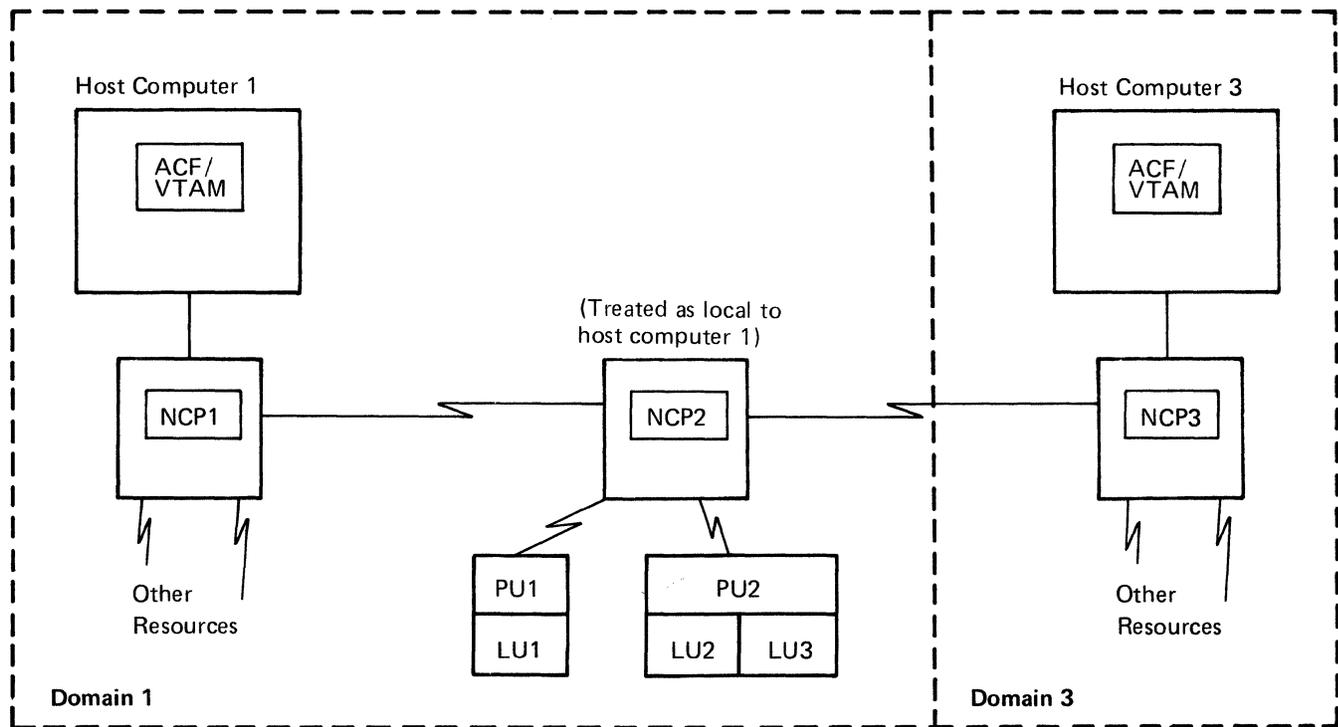


Figure 6-1. Resource Takeover (Local Controller Taken Over as Local Controller)

The network operator at host computer 1 can then release the communications controller by, for example, issuing a VARY command with the release (REL) option for NCP2, which releases not only the NCP but also the physical units and logical units that were acquired. In doing this, the operator specifies whether release of the NCP is to take effect normally or immediately. A normal release allows sessions in progress to be completed before the release takes effect. Immediate release causes sessions in progress to be broken.

The network operator at host computer 2 reactivates the cross-domain manager definitions for the cross-domain managers in domains 1 and 3, causing contact to be established between the cross-domain manager in domain 2 and the managers in domains 1 and 3.

The network operator at host computer 1 also activates the cross-domain resource major nodes containing the cross-domain resource definitions for LU1, LU2, and LU3, because those units will now be in a different domain.

The network operator at host computer 3 issues MODIFY commands to reassign LU1, LU2, and LU3 (as cross-domain resources) from the control of the cross-domain resource manager in host computer 1 to control of the cross-domain resource manager in host computer 2.

The network operator in domain 2 then activates the physical units that were released to him by host computer 1.

After the original acquisition of NCP2 by domain 1, communications would continue normally unless NCP2 or its communications controller failed and required reloading. If such a failure occurred, the acquired communications controller could not be reloaded as a local controller, because a local controller cannot be loaded through an SDLC link. To reload the controller, it would have to be transferred as a remote controller to either domain 1 or domain 3, as described in the next topic.

Transferring a Local Communications Controller as a Remote Communications Controller: A local communications controller in one domain can be transferred to an ACF/VTAM in an adjacent domain as a remote communications controller in the new domain. To do this, the channels of the controller must be turned off manually, and the communications controller must be loaded from the new host computer with an NCP for a remote communications controller. Any sessions with resources attached to the controller and any cross-domain sessions in which the controller is serving as a transit node are disrupted. The communications controller is then activated as a remote controller in the new host computer's domain. The transferred controller can be connected to only one local controller.

Figure 6-2 can be used to describe this process. Host computer 2 fails and NCP2 is to be transferred to host computer 1 as a remote communications controller, operating off of NCP1. The pattern for the transfer is as follows:

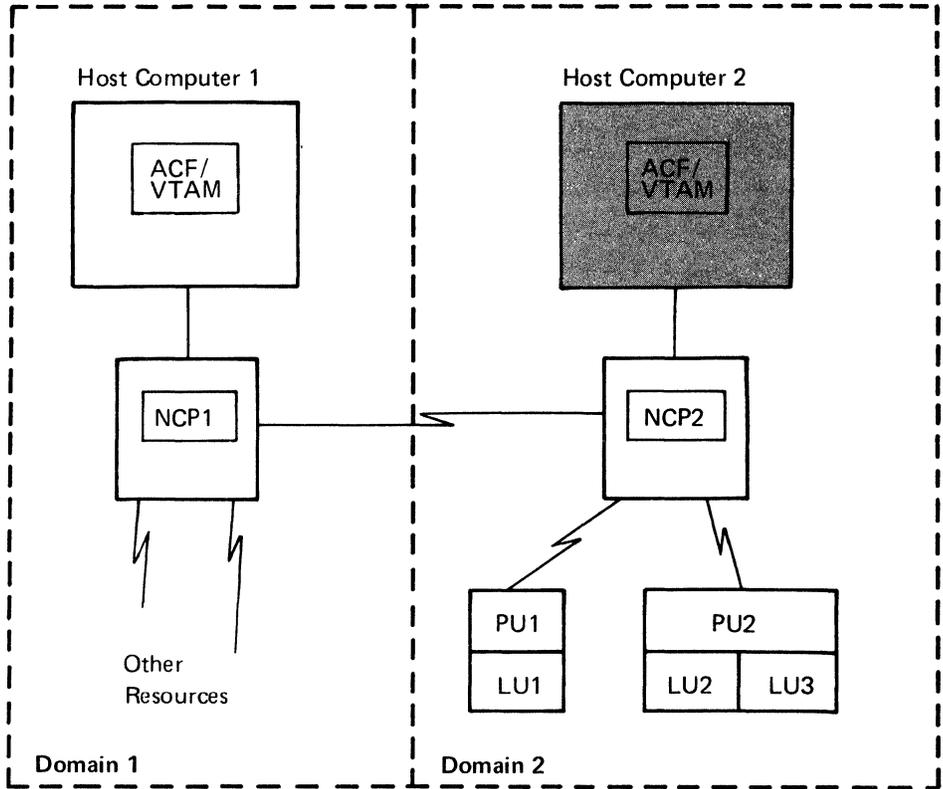
The network operator at host computer 2 manually turns off the channels of NCP2.

The network operator at host computer 2 informs the network operator at host computer 1 that the transfer can be started.

The network operator at host computer 1 deactivates the cross-domain resource major nodes for LU1, LU2, and LU3.

The network operator at host computer 1 issues a VARY ACT command, naming an NCP to be loaded into the controller. (The NCP to be loaded into the controller must be available in the appropriate library for the operating system being used in host computer 1.) This causes ACF/VTAM in host computer 1 to recognize the controller as a remote controller and to load the named NCP into it.

A. The Network before Takeover



B. The Network after Takeover

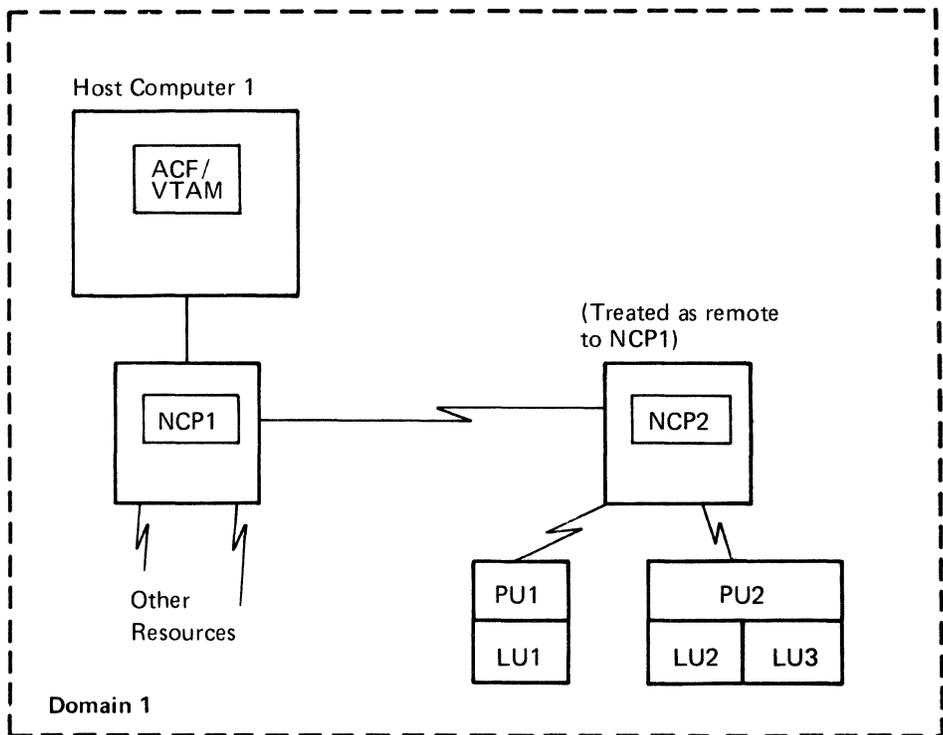


Figure 6-2. Resource Takeover (Local Controller Taken Over as Remote Controller)

The network operator in domain 1 then activates the physical units associated with the new remote controller, unless the units were automatically activated as the result of activating the NCP.

If operations are restored at host computer 2, the network operator at that host computer can take back the communications controller. The pattern is as follows:

The network operator at host computer 1 deactivates the remote communications controller.

The network operator at host computer 2 manually turns on the channels between host computer 2 and the controller.

The network operator at host computer 2 issues a VARY ACT command, causing the communications controller to be loaded with an NCP.

The network operator at host computer 2 reactivates his own cross-domain resource manager and the definition in his domain of the cross-domain resource manager in host computer 1. This reestablishes the cross-domain resource manager session (CDRM-CDRM session) between the managers.

The network operator at host computer 1 activates cross-domain resource major nodes to reflect LU1, LU2, and LU3 as being in domain 2.

Configuration Restart

ACF/VTAM provides the capability to restore the ACF/VTAM domain after certain types of failure occur. The following forms of recovery are provided:

Immediate Configuration Restart is an attempt by ACF/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 ACF/VTAM occurs.

Delayed Configuration Restart is the facility to restore the network to either its initial status or prefailure status after an ACF/VTAM failure, host operating system or host computer failure, or an NCP failure from which ACF/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: ACF/VTAM maintains a record of the configuration of the ACF/VTAM domain. 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, ACF/VTAM reinstates the NCP's minor-node status as it was before the failure. ACF/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 ACF/VTAM, or when an NCP to which the physical unit is attached fails and is restarted, ACF/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.

ACF/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 ACF/VTAM, the user can define a VSAM data set for any major node in the network in which ACF/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 ACF/VTAM is started, the network operator can specify a VSAM data set in which ACF/VTAM maintains a list of active major nodes. When ACF/VTAM is stopped, these data sets are kept for use when ACF/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 (an ACF/VTAM, host computer, or host operating system failure, or an NCP failure from which ACF/VTAM could not recover), the network operator can restore the ACF/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 ACF/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.

Transferring a Remote Communications Controller as a Remote Controller

A remote communications controller can be transferred to a different local communications controller in either the same domain or a different domain. To transfer the remote controller, the network operator activates a link between the remote controller and the new local controller. The NCP in the new local controller must have been generated to include support for a remote controller. If such support was not included, a new NCP including that support must be loaded into the local controller.

If the transfer is made within the same domain, a new NCP must be loaded into the remote controller if the station address of that controller is changed.

If the transfer is to a different domain, the appropriate changes must be made to each domain's cross-domain resource definitions to reflect the changes to the new owning cross-domain resource manager. Changes must also be made to the NCP's PATH statements in both the old local controller and the new local controller. (The remote controller's subarea address must be added to the path information in the old local controller and that subarea address must be deleted from the path information in the new local controller.) To make these changes, new NCPs must be loaded into both local controllers.

Lost Subarea Notification

Each major node in the network is assigned a subarea number either by the user (in definition statements) or by ACF/VTAM (during initialization or at activation). This subarea number is used by ACF/VTAM and other elements in the network to route traffic through the network.

A subarea in the network can be lost (for example, when a host computer or communications controller fails or the link to it fails). When a subarea is lost, the communications controllers and host computers that are directly attached to the lost subarea detect the loss and take action to notify other elements in the network. The detecting node sends to each adjacent node (except the lost node) a list of subareas that have been lost because of the outage. Each node receiving this list modifies the list as necessary before sending it on to nodes adjacent to it. The receiving node checks the list and leaves in the list only those subareas accessed through the adjacent node that notified it of the loss. If after this checking, any subareas remain in the list, the list is sent to all adjacent nodes (except the node from which the original list was received). In this way, notification of the lost subareas is propagated throughout the network to all affected nodes.

The notification enables recipients to send appropriate error information to application programs and logical units and to clean up sessions that cannot continue. For example, if ACF/VTAM is notified of a failure that interrupts a cross-domain session of one of its logical units, ACF/VTAM cleans up the session by deactivating and reactivating the logical unit.

Chapter 7. ACF/VTAM Planning Considerations and Requirements

The first part of the chapter states the machine and program requirements (including NCP requirements) for using ACF/VTAM. The second part of the chapter discusses ways a user can plan for and take advantage of features and facilities in ACF/VTAM.

Machine Requirements

This section discusses the machine requirements for a data communication system with ACF/VTAM. It specifies which units can be used with ACF/VTAM including the central processing units (CPUs) and their required features, communications controller requirements, and terminals and terminal features.

CPU Support

ACF/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 (MVS only), 165II, 168, and 168MP (MVS only).

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 (MVS only), 165II, 168, and 168MP (MVS only).

Local 3270s

ACF/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 ACF/VTAM. Appendix A lists the local 3270 devices that can be used with ACF/VTAM.

Local 3790s

ACF/VTAM provides communication with IBM 3790 Communication Systems that are locally attached. Appendix A lists 3790 devices that can be used with ACF/VTAM.

Requirements for Communications Controllers

For remote devices, ACF/VTAM requires a local IBM 3704 or 3705 Communications Controller.

ACF/VTAM uses only the network control program/VS (NCP) for communications controllers. ACF/VTAM uses only the network control mode of the NCP with or without

partitioned emulation programming (PEP). ACF/VTAM requires NCP/VS Version 5 or ACF/NCP/VS for single-domain communication. ACF/VTAM requires ACF/NCP/VS (NCP Version 6) for cross-domain communication.

ACF/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.

ACF/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

Cross-domain communications can occur only through a local 3705-I or 3705-II controller.

ACF/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 ACF/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 Channel Adapter, ACF/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.

ACF/VTAM does not support the manual Two-Channel Switch (8002) for the 3705. The user is responsible for support of this feature in an ACF/VTAM system.

Shared Communications Controller Support: A 3705-II Communications Controller can be attached to up to four host computers using four Type 4 Channel Adapters or two hosts using two Type 2 or Type 3 Channel Adapters. A 3705-I Communications Controller can be attached to up to two host computers using two Type 4 Channel Adapters.

Backup (Alternate) SDLC Link between Communications Controllers: ACF/VTAM supports use of an alternate SDLC link to back up the main SDLC link between communications controllers. This link can be switched or nonswitched. ACF/VTAM

notifies the network operator of the main link's failure; the network operator can then activate the backup link. The *NCP Generation* manual describes how to specify a backup link. *ACF/VTAM Network Operating Procedures* describes the procedures that the network operator must perform to switch to a backup link.

Other Feature Not Supported: ACF/VTAM does not support speed selection for lines equipped with dual-speed modems.

Remote Terminals

ACF/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 ACF/VTAM.

Because ACF/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 ACF/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 ACF/VTAM data sets can be calculated by using the *ACF/VTAM System Programmer's Guide* for the operating system being used by the installation. 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 ACF/VTAM data set storage in the publication mentioned above, because TOLTEP is included in ACF/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 *Storage Estimates and Performance Planning for the 3704 and 3705 Communications Controllers Network Control Program*.

Operating System Requirements

This section discusses ACF/VTAM's requirements for each operating system (DOS/VS, OS/VS1, and OS/VS2) in which ACF/VTAM can be included.

In a multiple-domain network, a different operating system can be used in each host computer in the network. For example, in a three-domain network with three host computers, DOS/VS can be used in one host, OS/VS1 in another, and OS/VS2 MVS in the third.

ACF/VTAM can be used with DOS/VS Release 34, OS/VS1 Release 6, OS/VS2 Release 1.7 (SVS), and OS/VS2 Release 3.7 (MVS), and with all subsequent releases of those systems unless otherwise indicated. ACF/VTAM will not run with any earlier releases of those operating systems.

Upward Compatibility

The ACF/VTAM macro language is upward compatible from DOS/VS to OS/VS1 to OS/VS2. In addition, procedures for defining the data communication network and network operator commands are similar for all three systems. ACF/VTAM's requirements for each of these systems are detailed in the remainder of this chapter.

Requirements for DOS/VS

ACF/VTAM can be installed in an existing DOS/VS system by using the Maintain System History Program (MSHP) to add the base ACF/VTAM modules to the system and optionally to add the Multisystem Networking Facility. The process, which is outlined in the Program Directory shipped with the product, is analogous to adding a large program temporary fix (PTF) to the system. Adding ACF/VTAM in this way does not require a system generation.

ACF/VTAM can be added with a process that requires system generation. This process is more complex and is not recommended.

When a DOS/VS system that is to include ACF/VTAM is generated, the operand VTAM must be specified for the teleprocessing access method in the SUPVR macro instruction. The system must also be generated to include some DOS/VS facilities that would be optional if ACF/VTAM were not in the system. Of these facilities, 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 PFIx 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 domain must be identified during the generation of the operating system. These devices are local 3270s, local 3790s, and the local communications controllers used by ACF/VTAM. Communication lines and remote devices need not be defined at system generation if they are to be used only through ACF/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 ACF/VTAM through the NCP.

A DOS/VS system with ACF/VTAM must have at least two partitions: one for ACF/VTAM and one for ACF/VTAM application programs. Additional partitions may be needed for other ACF/VTAM application programs or for programs unrelated to ACF/VTAM. The partition containing ACF/VTAM must have a priority higher than any other partition that contains an ACF/VTAM application program. In addition to DOS/VS tasks required for application programs, three DOS/VS tasks are required for ACF/VTAM. A fourth task is required if the network solicitor is used. Certain IBM-supplied programs, for example, Subsystem Support Services (SSS), can be run as subtasks of ACF/VTAM.

ACF/VTAM is started under DOS/VS by entering an EXEC command or job control statement for the partition in which ACF/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 ACF/VTAM network. The assignment must make logical unit SYS000 unassigned for the duration of the ACF/VTAM job step. Any other commands or job control statements required to start ACF/VTAM depend upon factors such as system generation and telecommunication options to be used. See "ACF/VTAM Data Sets" later in this chapter for library and file requirements for ACF/VTAM under DOS/VS.

Requirements for OS/VS

To install ACF/VTAM in an OS/VS1 or OS/VS2 MVS system, the user has a choice between:

Using the System Modification Program (SMP) to add ACF/VTAM to a running system. SMP can be used to add the base ACF/VTAM modules and, optionally, the Multisystem Networking Facility modules.

Using SMP to install the base ACF/VTAM modules and, optionally, the Multisystem Networking Facility modules in the distribution libraries (DLIBs), and then performing a system generation.

For an OS/VS2 SVS system, ACF/VTAM is installed as an independent component release (ICR).

When a system generation is performed, all *local* devices in the ACF/VTAM domain must be identified. These devices are the local 3270s, local SNA devices, and the local communications controllers used by ACF/VTAM. Communication lines and remote devices need not be defined at system generation if they are to be used only through ACF/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 ACF/VTAM through the NCP. Data set requirements for OS/VS are discussed in “ACF/VTAM Data Sets,” later in this chapter.

In OS/VS1, ACF/VTAM is started in a user-numbered partition. This partition must be defined as a problem program partition or system partition before ACF/VTAM is started. If fetch-protection is desired, ACF/VTAM must run in a problem program partition. In OS/VS2 SVS, ACF/VTAM runs in a region that is created when the ACF/VTAM job step is initiated. In OS/VS2 MVS, ACF/VTAM runs as a problem program.

A cataloged procedure must be created for ACF/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 ACF/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, local SNA devices, communications controllers, and remote terminals and devices that are part of the ACF/VTAM data communication network do *not* require DD statements because those devices are defined to ACF/VTAM in definition statements.

Multiple console support (MCS) can be used with ACF/VTAM. Any MCS console that is to be used by the ACF/VTAM network operator must be assigned a command authority of 2 and a routing code of 8.

It is not recommended that ACF/VTAM be run with the Dynamic Support System (DSS). The effects of DSS on ACF/VTAM includes DSS time delays that may result in NCPs in the ACF/VTAM network automatically closing down the network. Refer to the publication *OS/VS Dynamic Support System* for details on DSS.

If an ACF/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

ACF/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 macro instructions are then used as definition statements to define the remote fixed part of the domain to ACF/VTAM. The variable part of the remote domain, the switched major nodes, are not described to the NCP; they need only be described to ACF/VTAM.

The remainder of this chapter contains details on ACF/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 ACF/VTAM

Several types of NCPs may be needed in an ACF/VTAM 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)
- An NCP for cross-domain communication

Whatever the NCP or NCPs used, some functions that are optional for the NCP are required by ACF/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 for 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 ACF/VTAM

To define an NCP and its remote network to ACF/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 ACF/VTAM
- The network configuration

NCP generation macro instructions and the PCCU definition statement supply this information to ACF/VTAM. Two additional ACF/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 ACF/VTAM, identifies the communications controller into which the NCP would normally be loaded when activated. If the NCP is used in a local communications controller that is attached to more than one host computer, the PCCU definition statement is used to define the access method using the NCP.

This statement also defines the functions ACF/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 3704/3705 initial test routine automatically prior to loading the NCP. (Initial testing is optional only 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, ACF/VTAM requires information contained in the NCP generation macro instructions (also called ACF/VTAM definition statements). Most information required by ACF/VTAM is also required by the NCP, although some additional data is required by ACF/VTAM alone.

Pacing

ACF/VTAM enables the user to control the rate of data flow through the path joining an ACF/VTAM application program and a logical unit or secondary application program. 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 or APPL statement by the user, both ACF/VTAM and the NCP can limit the number of messages transmitted to a particular logical unit or application program before a response called a *pacing response* is returned. The same type of control is provided for messages being transmitted from the logical unit to the host computer. For more information on pacing, see the *ACF/VTAM System Programmer's Guide* for the operating system being used at the installation.

Support for SNA Switched Networks

This section describes ACF/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 ACF/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 SNA Major Node

A switched SNA 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 ACF/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 physical 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 ACF/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.)

In a multiple-domain network, the user can define a switched major node either (1) in one domain only and treat it as a cross-domain resource in other domains or (2) in each domain in which the major node will be used. If defined in only one domain, it can dial in to or be dialed out from only that computer, and cross-domain communications will require more definition by system programmers and will require more ACF/VTAM processing. If defined in more than one domain, the switched major node can be dialed in to or dialed out from each of the host computers in which it is defined.

Consideration for ACF/VTAM Switched Network Backup: If the user plans to use a switched line to back up a nonswitched line (see “Switched Network Backup” in Chapter 6), a physical unit and associated logical units must be defined not only as part of an NCP but also as a switched SNA major node. In the definition of the switched SNA major node, the physical unit must have a different name from the name used for the physical unit when it was defined as part of the NCP; the logical units should have the same names. The definition as a switched SNA major node is not activated until the nonswitched line fails and the operator must activate a switched line as a backup line to the physical unit.

Generating Switched Line Groups in the NCP

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.

In regard to switched line groups, for each telecommunication access method to which a communications controller can be connected, one LUPPOOL macro instruction must be included in the NCP generation deck to define the maximum number of logical units on switched SDLC links that the NCP is to handle at any one time for the access method.

ID Verification for Switched SNA Major Nodes

During a dialing operation, the station ID of the physical unit and the SSCP ID of the controlling SSCP are checked. If both IDs are valid, the dial physical connection is established. (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 dialed in. The physical unit sends a response containing its station ID. The station ID is then sent to ACF/VTAM which validates the ID. Finding that the station ID is valid, ACF/VTAM sends an Activate Physical Unit command containing the SSCP 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.

Dial-In Operation

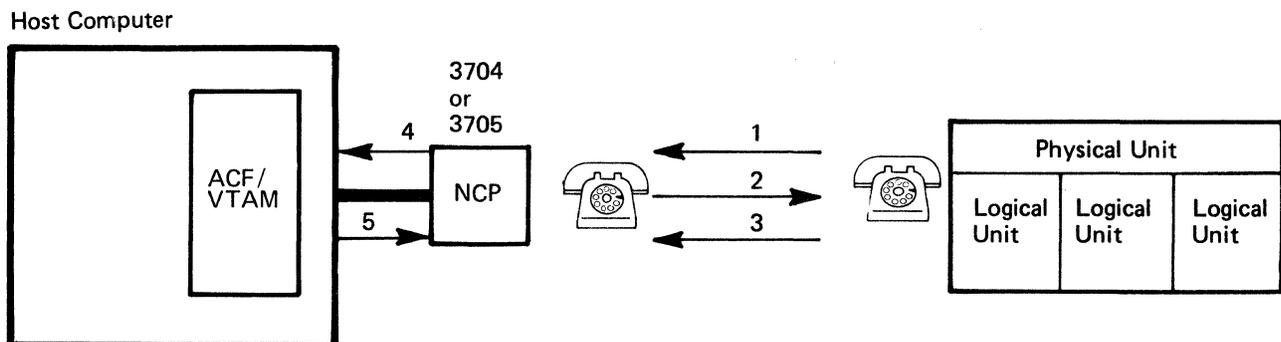
A dial-in operation can be performed from a physical unit defined in a switched major node if the line used in the dial-in operation has been defined as having dial-in capability. Before the dial-in operation can be performed, the following conditions must exist:

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 physical unit (PU) definition within the switched major node must have been activated.

The dial-in operation can be accomplished by manually dialing a dial port of the communications controller or by having a program in a cluster control unit perform the dialing operation (for example, a 3791 Controller can be programmed to perform dial-in operations). When the dial-in call reaches the communications controller, the physical unit's station ID and the host's SSCP ID are exchanged and verified (see Figure 7-1). Once the dial-in operation has been completed, the rest of the connection procedure is the same as for physical units and logical units on nonswitched lines.



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 sends a response containing its station ID.
4. The NCP sends the station ID to ACF/VTAM to be checked.
5. ACF/VTAM checks the identifier and returns the SSCP ID and other information pertaining to the physical unit. The procedure from this point on is the same as that for a physical unit on a nonswitched line.

Figure 7-1. Dial-In Operation

If the network operator previously issued a logon on behalf of the logical unit (with the VARY LOGON command) or if automatic logon was specified (with the LOGAPPL operand) in the definition statement for the logical unit, the logical unit is connected to the application program specified in the command or automatic logon specification. (This occurs if the controlling application program named in the command or automatic logon is available and is accepting logons.) If no controlling application program has been designated for the logical unit or if the program is not active, the dial-in physical connection is completed, and the logical unit remains available for a later session with an application program.

That later session can result from any of the following:

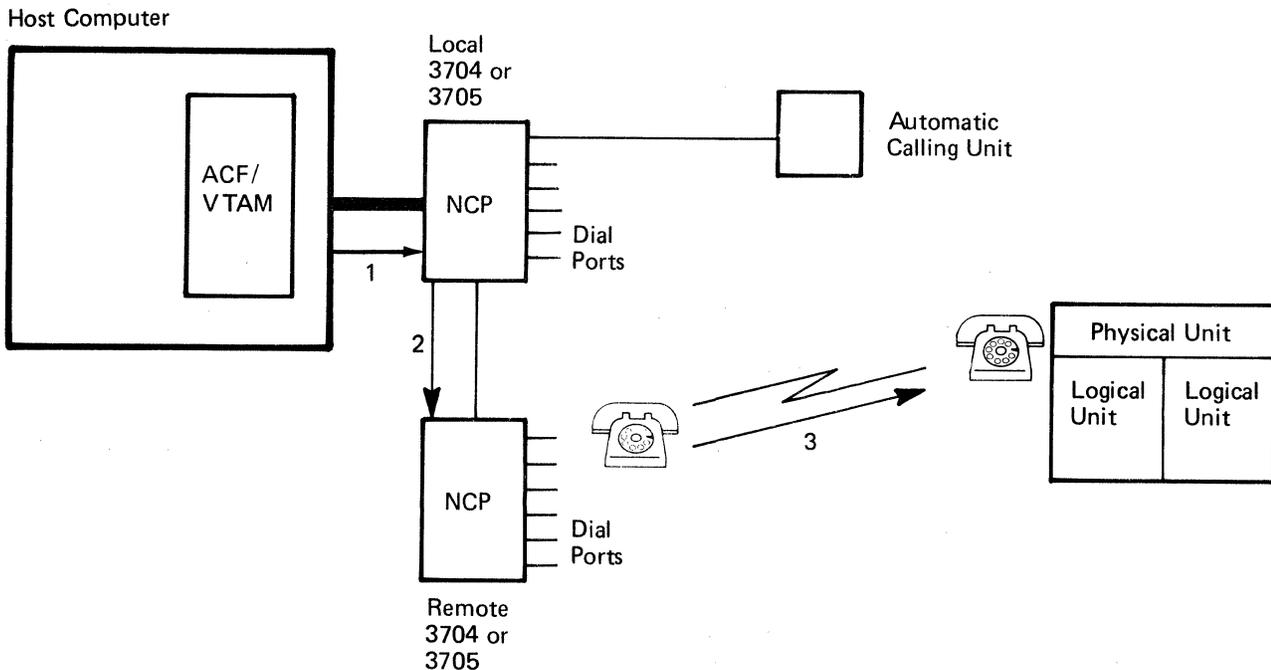
The controlling application program issues a SETLOGON with OPTCD=START, allowing its LOGON exit routine to be scheduled to accept the session establishment request for the logical unit.

A logon is entered at the logical unit and is sent to ACF/VTAM, requesting a session with an application program.

An application program issues a request to acquire the logical unit (either an OPNDST with OPTCD=ACQUIRE or a SIMLOGON). Note that such a request for a session with a logical unit associated with a dial-in-only physical unit is unsuccessful unless the physical unit has dialed in before the request is issued.

Dial-Out Operation

A dial-out operation is performed by ACF/VTAM on behalf of an application program to establish a session with a specific logical unit (see Figure 7-2). Dial-out operations can be performed only on those physical units and logical units that have dial-related PATH statements associated with them in the ACF/VTAM definition.



- 1 and 2. ACF/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, ACF/VTAM uses the path information in the switched major node definition (built from dial-related PATH statements during ACF/VTAM definition) and the switched line groups. When a request is made to dial out to a particular physical unit, ACF/VTAM:

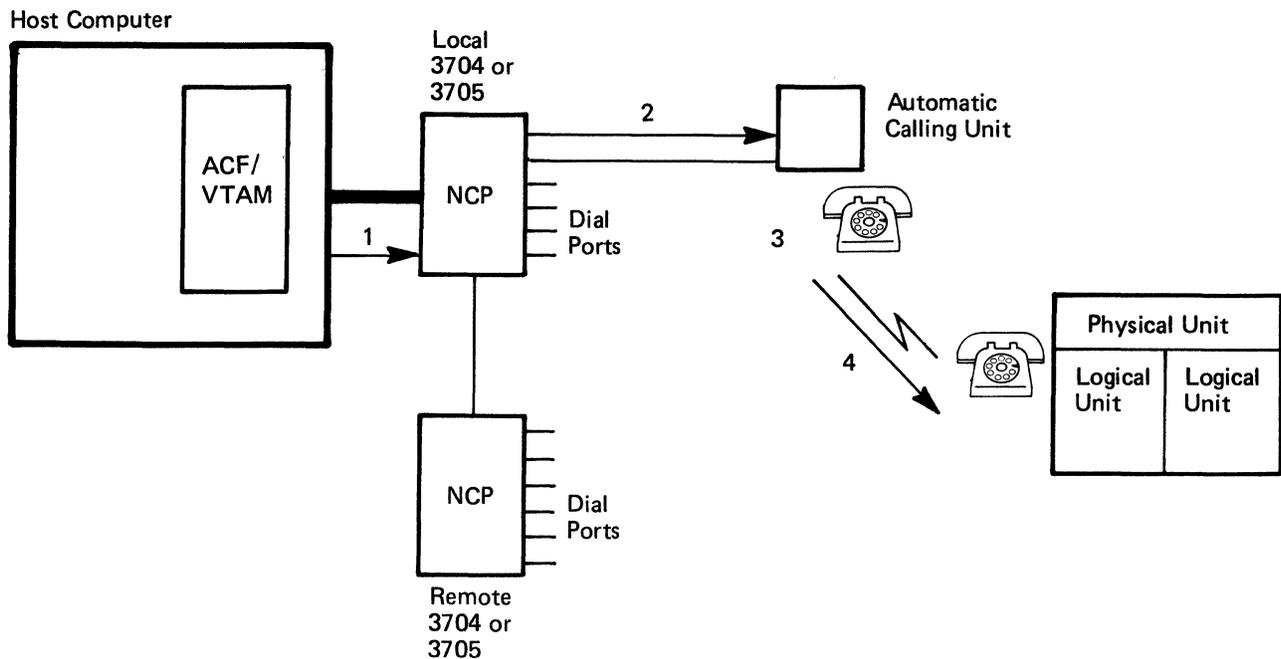
Picks a usable line (dial-out port) within the switched line group identified by the first dial-related PATH statement for the physical unit, and specifies the line and a telephone number to be used.

If that dial-out attempt fails, finds another usable line in the line group and makes another attempt (until physical connection is made or all the lines identified by the first PATH statement are used unsuccessfully).

Uses the next dial-related PATH statement for the physical unit to find another group of lines and attempts the dial-out operation on each usable line (dial-out port) in that group (see Figure 7-3).

Rejects the request for connection if all paths are unusable. The ACF/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.



1. ACF/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 physical connection procedure.

Figure 7-3. An Alternate Path for a Dial-Out Operation with an Automatic Calling 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.

Physical Disconnection of Physical Units and Logical Units in a Switched Major Node

All physical units and logical units in a switched major node have automatic call termination unless the user specifies otherwise. When all sessions between logical units and application programs are terminated, the physical unit is physically disconnected.

Physical disconnection procedures can be changed for each physical unit in the switched major node using the DISCNT operand on the PU statement during ACF/VTAM definition. If DISCNT=YES is specified, automatic call termination is used as described above. If DISCNT=NO is specified, the physical unit remains physically connected to ACF/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's Physical Connection" in Chapter 3 for a description of specifying physical disconnection of the physical unit on a disconnection request.

Partitioned Emulation Programming (PEP)

ACF/VTAM can use the NCP with partitioned emulation programming (PEP). When PEP is in use in the NCP, ACF/VTAM does not provide services for the network serviced by emulation mode. ACF/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 ACF/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.

ACF/VTAM also provides support for the restart capability of the NCP. If an error is encountered in a communications controller and this error causes ACF/VTAM to reload the controller, the reloading disrupts emulation because the entire NCP is reloaded, including the emulation function.

ACF/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. ACF/VTAM's support for such lines is as follows:

If the line is not active for ACF/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 ACF/VTAM, and they are reassigned to emulation mode when they are deactivated by ACF/VTAM.

ACF/VTAM Data Sets

ACF/VTAM uses a number of data sets (files in DOS/VS) to support a data communication system. These data sets contain such items as ACF/VTAM modules, ACF/VTAM definition statements, NCP modules, and RAS records. Data set requirements vary depending upon the operating system in which ACF/VTAM is being used.

Data Sets for ACF/VTAM under OS/VS

In an OS/VS system, ACF/VTAM uses NCP data sets, operating system data sets, and data sets devoted entirely to ACF/VTAM itself. Figure 7-4 is a table of all the data sets used by ACF/VTAM in an OS/VS system. The table columns show the following:

Name: Specifies the name assigned to the data set.

ACF/VTAM Contents: Indicates what information in the data set is used by ACF/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 ACF/VTAM start procedure.

Comments: Provides additional information relevant to the data set. If the data set is specified as “required,” it is required by ACF/VTAM. If it is specified as “optional,” it is not required by ACF/VTAM.

Note: *If an optional data set is not included in an ACF/VTAM system, the contents of that data set, and therefore the associated function, are not available to ACF/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 *NCP Generation* manual for additional information on NCP data sets listed in Figure 7-4.

Data Requirements for ACF/VTAM under DOS/VS

The table in Figure 7-5 shows the file requirements for ACF/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.

File Identification: Specifies the identification of the file on the volume.

Symbolic Device Name: Specifies the name to be used or indicates how the name is supplied.

Content: Indicates what ACF/VTAM uses or puts in this file.

Comments: Provides additional information relevant to the file. If the file is specified as “required,” it is required by ACF/VTAM.

Refer to the publication *Introduction to DOS/VS* for additional information on the DOS/VS logical units and files. Refer to the *NCP Generation* manual for additional information on NCP data requirements.

Name	ACF/VTAM Contents	When Created	JCL	Comments
<i>ACF/VTAM Data Sets</i>				
SYS1.VTAMLIB	Load Modules for ACF/VTAM; see Note 1	System generation	Yes	Only the ACF/VTAM load modules are required
SYS1.VTAMLST	ACF/VTAM definition statements and start options	Prior to starting ACF/VTAM	Yes	Required. See Note 2
SYS1.VTAMOBJ	ACF/VTAM network configuration table	Prior to starting ACF/VTAM	Yes	Required. See Note 3
<i>Operating System Data Sets</i>				
SYS1.DUMP	Records for SVC dumps	IPL	No	Optional
SYS1.LINKLIB	ACF/VTAM initialization module, used when ACF/VTAM is started	System generation	No	Required
	NCP loader utility program	Prior to starting ACF/VTAM	Yes	Required
	NCP dump bootstrap program	Prior to starting ACF/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 ACF/VTAM
SYS1.LOGREC	ACF/VTAM error records	System generation	No	Required
SYS1.LPALIB	ACF/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 in OS/VS2 only
SYS1.MACLIB	ACF/VTAM macro definitions	System generation	No	Required
SYS1.NUCLEUS	ACF/VTAM resident SVCs and abnormal termination modules	System generation	No	Required
SYS1.PARMLIB	ACF/VTAM partition definition	System generation	No	Required
SYS1.PROCLIB	ACF/VTAM cataloged procedure	System generation	No	Required
SYS1.SVCLIB	ACF/VTAM non-resident SVCs and ERPs for local devices	System generation	No	Required in OS/VS1 only
SYS1.TRACE	GTF trace records for ACF/VTAM	System generation	No	Optional
<i>NCP Data Sets</i>				
NCP dump	Dump records for the NCP	When ACF/VTAM is started	Yes	Required if ACF/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	Each NCP is stored as a separate member of the library. See Note 4
Configuration restart data sets	ACF/VTAM status of network nodes	Prior to starting ACF/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 ACF/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:

1. SYS1.VTAMLIB is referred to as the *ACF/VTAM load module library*. For OS/VS1, SYS1.VTAMLIB contains ACF/VTAM load modules for the ACF/VTAM virtual partition; for OS/VS2, it contains ACF/VTAM load modules for the ACF/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 *ACF/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 ACF/VTAM alphanumeric characters specified by the installation). ATCSTRxx members contain lists of major nodes to be activated automatically when ACF/VTAM is started. See "Defining the Network to ACF/VTAM" in Chapter 3 for details on ACF/VTAM definition statements and ACF/VTAM start options.
3. When a major node is activated, ACF/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 ACF/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, ACF/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, ACF/VTAM builds a new table (based on the modified definition) and stores a new copy on SYS1.VTAMOBJ.
4. All NCPs can be in the same library as long as each NCP has a unique name.

Figure 7-4. Data Sets Used by ACF/VTAM under OS/VS

File Name	File Identification	Symbolic Device Name	Contents	Comments
DIAGFLE	Any name chosen by the user	SYS008	Communications controller pro-IPL testing modules	Required if the controller is to be tested automatically prior to the loading of the NCP by ACF/VTAM.
NCP load file	Any name chosen by the user	Name is specified in NCP generation deck	NCP load module after CSERV	
NCPDUMP	Any name chosen by the user	NCPLUB=SYSxxx is specified in the NCP generation decks	Dump records for the NCP	Required if ACF/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	Any name chosen by the user	SYS001	ACF/VTAM trace records	Required only if tracing is to be performed and trace records are to be written to tape or disk. The file is not defined if trace records are to be retained in a trace area in main storage.

Figure 7-5. Files Used by ACF/VTAM under DOS/VS

ACF/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 ACF/VTAM under DOS/VS are described in Figure 7-6.

Sharing Resources

ACF/VTAM permits resources in its data communication 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 ACF/VTAM facilities, application programs can share terminal components, terminal devices, cluster control units, communications controllers, NCP facilities, communication lines, and data channels. Of these elements, terminal devices (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, a logical unit or non-SNA terminal (both end nodes) can be connected to or queued for logon to only one application program at a time. (A primary application program can have multiple connections to other secondary application programs, logical units, and terminals.) If an active terminal device is not connected or queued for logon to an application program, it is available for connection to any application program.

On the other hand, ACF/VTAM does not explicitly connect path elements to end nodes. Instead, ACF/VTAM employs path elements on behalf of end nodes only as long as needed to complete a specific data-transfer operation. 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. These nodes can be in the same or different domains.

DOS/VS Library	Contents	Comments
Core image library	ACF/VTAM modules; NCP tables and block handler routines; configuration data sets and online terminal tests for TOLTEP	Required. See Note 1.
Configuration restart files	ACF/VTAM status of network nodes	Required if a warm restart is to be used.
Procedure library	ACF/VTAM start procedure	Optional.
Source statement library	ACF/VTAM definition statements and start options	Required. See Note 2.

Notes:

1. In this publication, *ACF/VTAM load module library* refers to the ACF/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, *ACF/VTAM definition library* refers to the ACF/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. ACF/VTAM's Use of Libraries under DOS/VS

Sharing and Managing Resources

While ACF/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 ACF/VTAM Definition

The telecommunication network that ACF/VTAM controls is the one presented to it through network definition. ACF/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 ACF/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 ACF/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 ACF/VTAM network operator commands (except START and HALT) using the SENDCMD and RVCMD macro instructions. Such an application program (a program operator) can monitor and control the ACF/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 ACF/VTAM allows any terminal to be connected to any application program, a user can code an authorization exit routine restricting specific connections. ACF/VTAM passes control to this exit routine during the processing of connection and disconnection requests.

Initial Status of Nodes: Using ACF/VTAM definition statements and start options, a user specifies the initial status of each minor node in the ACF/VTAM system. These minor nodes are defined as initially active or initially inactive. If a minor node is not activated when ACF/VTAM is started, the network operator must explicitly activate it before it can be used by ACF/VTAM. On the other hand, if a minor node is activated when ACF/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 ACF/VTAM is started) without network operator intervention.

Managing Resources through NCP Generation

Because ACF/VTAM uses the NCP to communicate with remote devices, options generated in the NCP may affect ACF/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 an ACF/VTAM system has been generated and defined, the availability of resources can be influenced by the application programs' use of ACF/VTAM. An application program can monopolize telecommunication resources that normally would be shared:

As long as a logical unit or non-SNA 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 ACF/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.

ACF/VTAM Security

Using ACF/VTAM facilities, a user can control the use of terminals and application programs. Resources to be controlled are identified during ACF/VTAM definition; control of the access to these resources can be performed at various points in the data communication network and during various stages of 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)

ACF/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 ACF/VTAM
- Restricting an application program's use of ACF/VTAM facilities
- Protecting sensitive data being transmitted by ACF/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. ACF/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 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 station ID, which is sent to ACF/VTAM for verification. If the station ID is valid, ACF/VTAM sends its SSCP ID to the physical unit by the Activate Physical Unit command and gives the NCP information to support a session (such as physical unit type and segment size used by the physical unit). If the station ID is not valid, ACF/VTAM disconnects the physical unit. When the identifications have been exchanged, normal physical connection procedures continue.

Host ID Verification for Physical Units

ACF/VTAM allows a host identification number to be specified when ACF/VTAM is started, either in a start procedure or by the network operator. This host identification number (or SSCP ID) 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 ACF/VTAM in the host computer. (See "Network Control Program Requirements" earlier in this chapter for more information on these definition statements.)

In ACF/VTAM under OS/VS, a user can distribute verification authority between an NCP in a communications controller and ACF/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 ACF/VTAM verify less frequently used terminals. In ACF/VTAM under DOS/VS, verification for dial-in operations must be done by ACF/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 ACF/VTAM for further verification processing or stop communicating with the terminal.

Using the VIDLIST definition statement, the user can specify the following information for ACF/VTAM:

The identification character string for each BSC or TWX terminal on a switched line to be verified by ACF/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 ACF/VTAM is unable to verify a terminal identification, it disconnects the terminal or passes it to the application program designated to handle unidentified terminals. ACF/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 an ACF/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 ACF/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 an ACF/VTAM system under OS/VS, three options are available:

Use the method described for ACF/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 **ACF/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 **ACF/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 **ACF/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 **ACF/VTAM**.

Dividing the verification responsibility between the NCP and **ACF/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 **ACF/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

ACF/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 (with the **PASS** option) 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 **ACF/VTAM**, the authorization exit routine indicates to **ACF/VTAM** whether the request should be permitted. If the request is valid, it is completed by **ACF/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 **ACF/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

ACF/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 ACF/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 ACF/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 ACF/VTAM definition, ACF/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 ACF/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 ACF/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. ACF/VTAM provides a number of tools for verifying logons in the application program. Using these tools effectively requires planning and preparation at ACF/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 (including specially defined remote BSC 3270 terminals), use a USS definition table to define logon formats.

For local non-SNA 3270, BSC, and start-stop terminals, use an interpret table to define logon formats to ACF/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 ACF/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 ACF/VTAM's network solicitor (for local non-SNA 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, ACF/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: ACF/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: ACF/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 ACF/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 ACF/VTAM's logon facilities, the following conditions must be met:

The terminal, the logon, and the application program must be defined at ACF/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 non-SNA terminal (local non-SNA 3270, BSC, or 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.

ACF/VTAM also allows a user to control other types of logons. As noted above, the automatic logon is specified at ACF/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 ACF/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 non-SNA 3270, BSC, or start-stop terminal, that terminal has access to any application program in the ACF/VTAM system. If caution is not exercised when authorizing OS/VS logons, the security of the data communication 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, ACF/VTAM nodes can be addressed in the active system by symbolic names. The symbolic names are assigned at ACF/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 and PU definition statements; 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 ACF/VTAM

An application program must open an access method control block (ACB) before it can use ACF/VTAM resources and facilities. To be opened, this control block must supply an application program identification for verification by ACF/VTAM. The identification is the symbolic name of (or the ACBNAME within) an APPL definition statement specified and filed in a member of the ACF/VTAM definition library during ACF/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 ACF/VTAM. A program's initial use of ACF/VTAM (that is, issuing an OPEN macro instruction for an ACF/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 ACF/VTAM Facilities

Although an application program has been recognized by ACF/VTAM (an ACF/VTAM ACB has been successfully opened), it may still not have access to all of ACF/VTAM's facilities. The user controls access to the following facilities during ACF/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

The use of the SENDCMD and RCVCMD 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, ACF/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 ACF/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 ACF/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 ACF/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 ACF/VTAM's trace facility to define confidential data.

Other Telecommunication Access Methods

ACF/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 can be executed concurrently as long as they have separate networks.

Other Telecommunication Access Methods in DOS/VS

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

ACF/VTAM application programs use ACF/VTAM to communicate with:

- Terminals attached to communications controllers by lines in network control mode
- 3270 terminals attached locally
- 3790 terminals attached locally
- Other application programs

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 7-7 illustrates concurrent use of QTAM, BTAM, and ACF/VTAM in DOS/VS.

With concurrent execution of the access method, a single application program can use both BTAM and ACF/VTAM to communicate with separate networks, provided that all requirements of both access methods are met.

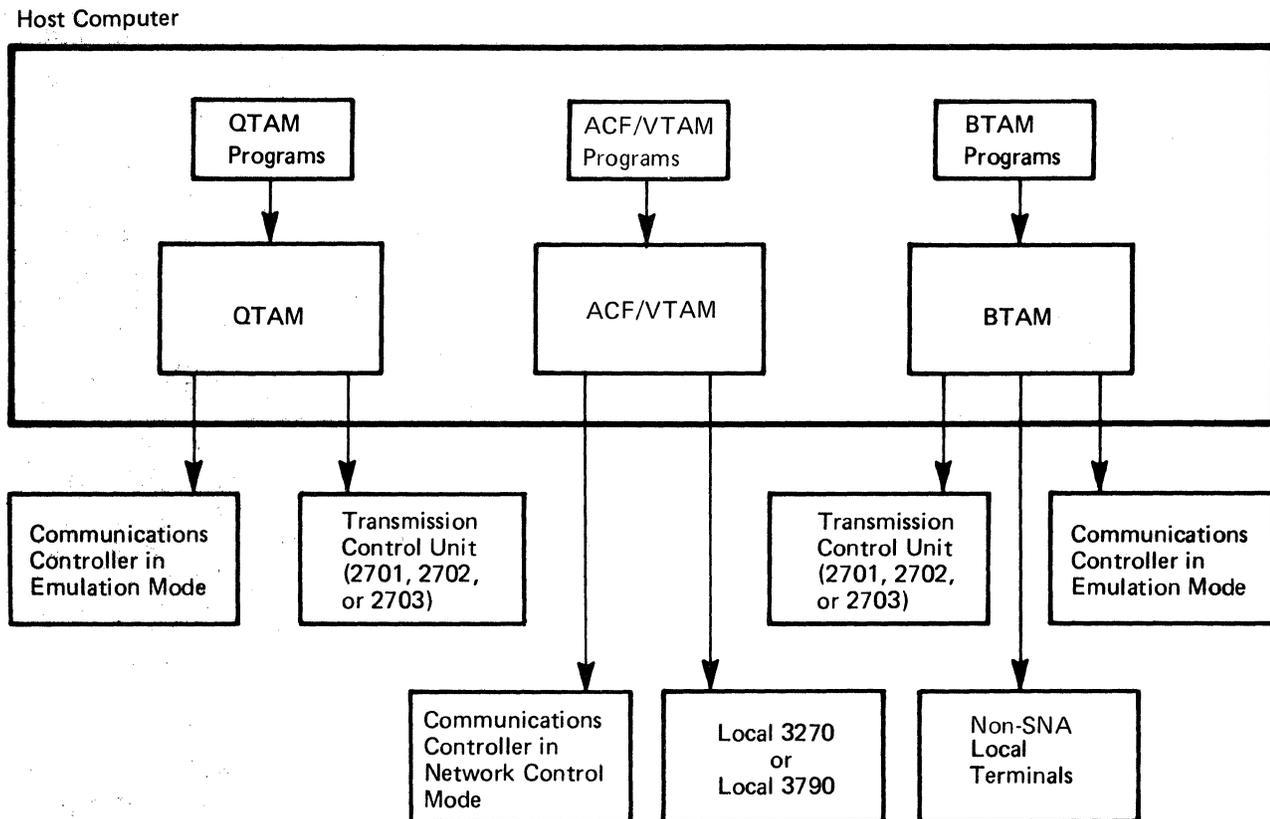


Figure 7-7. Other Telecommunication Access Methods in DOS/VS

Other Telecommunication Access Methods in OS/VS

In an OS/VS system, BTAM, TCAM, and ACF/VTAM can operate concurrently.

BTAM programs use BTAM to communicate with:

- Terminals attached to transmission control units
- Terminals attached to communications controllers by lines in emulation mode
- Locally attached non-SNA terminals

TCAM programs use TCAM to communicate with:

- Terminals attached to communications controllers by lines in network control mode
- Terminals attached to communications controllers by lines in emulation mode
- 3270, 2260, and 7770 terminals locally attached

ACF/VTAM application programs use ACF/VTAM to communicate with:

- Terminals attached to communications controllers by lines in network control mode
- 3270 terminals locally attached
- 3790 terminals locally attached
- Other application programs

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 7-8 illustrates the concurrent use of BTAM, TCAM, and ACF/VTAM in OS/VS.

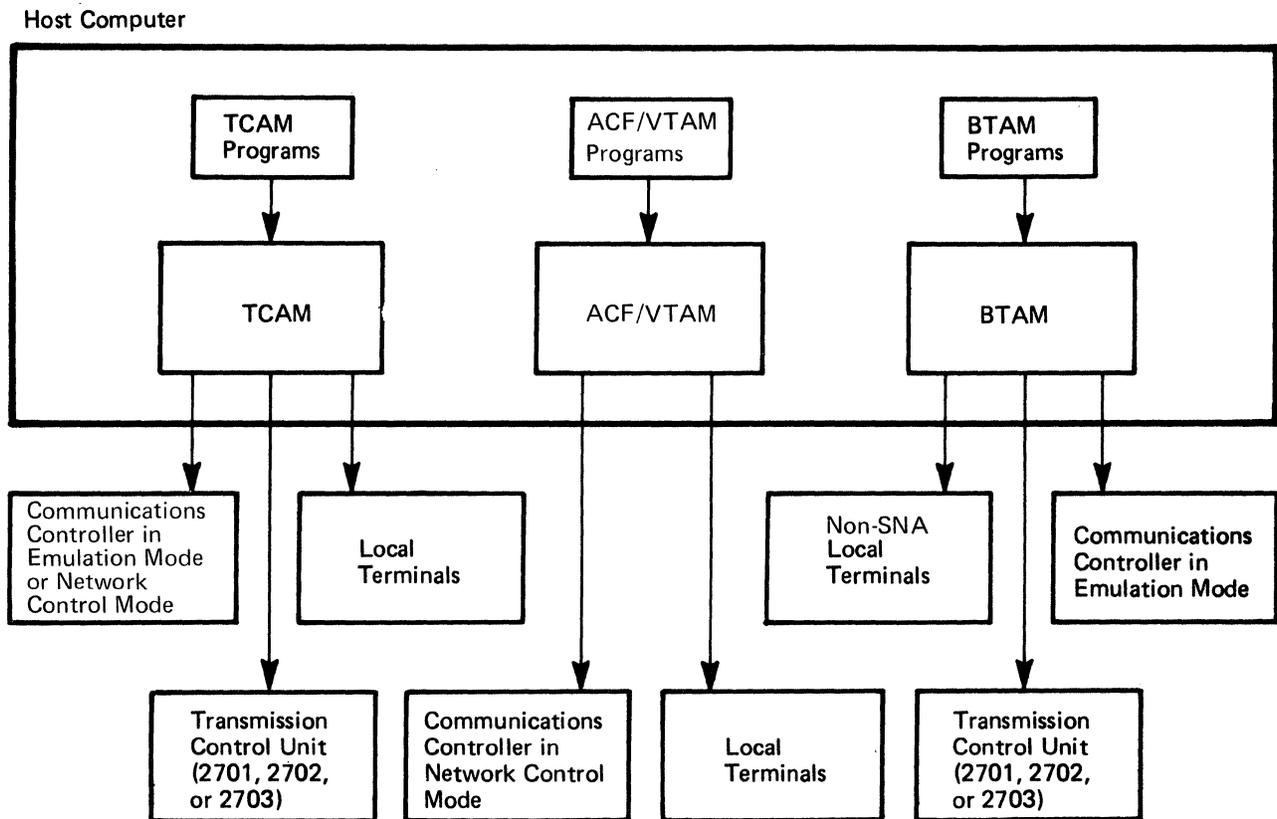


Figure 7-8. Other Telecommunication Access Methods in OS/VS

With the concurrent execution of the access methods, a single application program can use BTAM, TCAM, and ACF/VTAM to communicate with separate networks, provided that all of the requirements of all access methods are met.

Other Telecommunication Access Methods in a Multiple-Domain Network

In a multiple-domain network, SNA resources in an ACF/VTAM domain can communicate with SNA resources in an ACF/TCAM domain if all requirements for cross-domain communication of both access methods are met. (For ACF/TCAM requirements, see *Advanced Communications Function for TCAM: General Information, GC30-2050.*) Figure 7-9 illustrates use of ACF/VTAM and ACF/TCAM in a multiple-domain network. In addition, other access methods can operate in each host computer as described above.

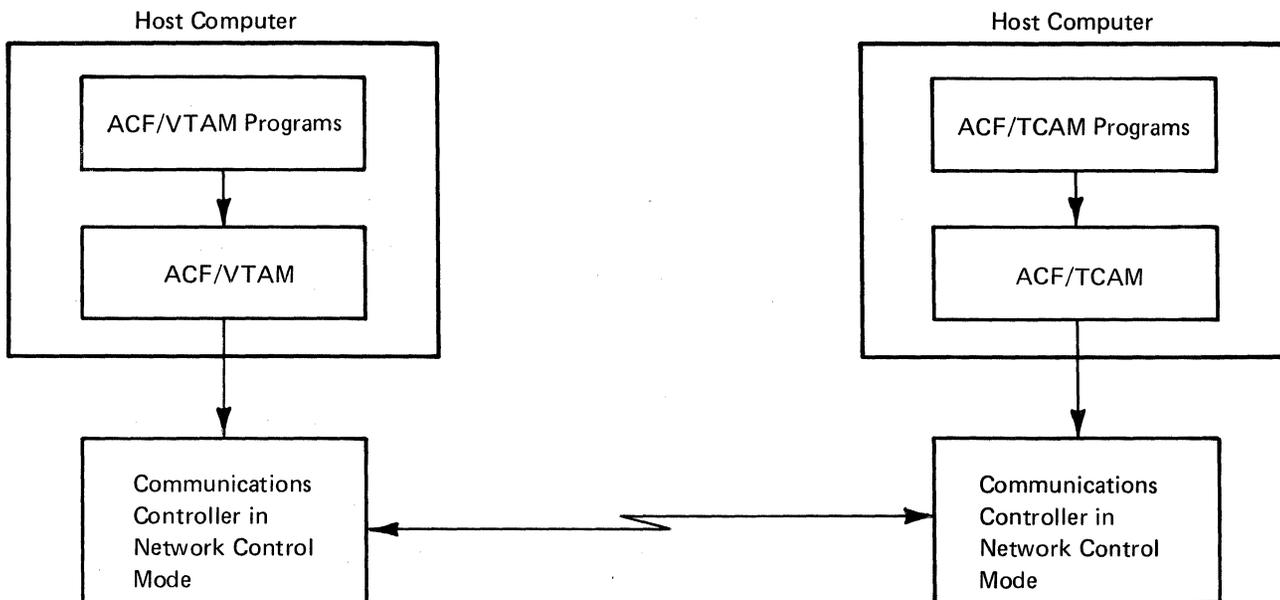


Figure 7-9. Other Telecommunication Access Methods in a Multiple-Domain Network

Chapter 8. Support for Non-SNA Terminals

In addition to its support for terminals in a Systems Network Architecture (SNA) environment, ACF/VTAM supports the non-SNA terminals listed in Appendix A. The information in the previous chapters applies to both SNA terminals and non-SNA terminals, except as noted. This chapter tells what specific facilities or requirements are not applicable to non-SNA terminals and describes the special facilities and requirements for these terminals that are not described elsewhere in this book.

Note: Non-SNA terminals used in basic mode can communicate only within their domain. Specially-defined BSC 3270 terminals used in record mode can communicate across domain boundaries.

Using BTAM

Figure 8-1 shows the major similarities and differences between ACF/VTAM application programs and BTAM application programs. ACF/VTAM application program characteristics shown in Figure 8-1 are discussed in more detail in this chapter and in Chapter 5.

Figure 8-2 shows that communication using BTAM can be combined with communication using ACF/VTAM in a number of ways:

An application program that uses ACF/VTAM record-mode macro instructions to communicate with logical units can also contain BTAM macro instructions to communicate with non-SNA terminals.

An application program that uses ACF/VTAM record-mode macro instructions to communicate with logical units can use ACF/VTAM basic-mode macro instructions to communicate with some non-SNA terminals and can also use BTAM macro instructions to communicate with other non-SNA terminals. A program of this nature is required if it is necessary to communicate with some non-SNA terminals that are supported by ACF/VTAM and some that are not.

One application program can be used for ACF/VTAM communications, and another can be used for BTAM communications.

To continue using terminals not supported by ACF/VTAM requires one of these combinations. Even if all non-SNA terminals in a network are supported by ACF/VTAM, the user might want to use BTAM to communicate with them.

Using ACF/VTAM

Using ACF/VTAM for non-SNA terminals requires:

- Creating an ACF/VTAM network (that includes these terminals)
- Operating the network
- Writing ACF/VTAM application programs

Except as described below, the information in Chapters 3, 4, and 5, on creating a network, operating the network, and writing ACF/VTAM application programs applies for non-SNA terminals. Chapters 6 and 7 also apply, except where noted, to non-SNA terminals. In addition, ACF/VTAM provides special support consisting of:

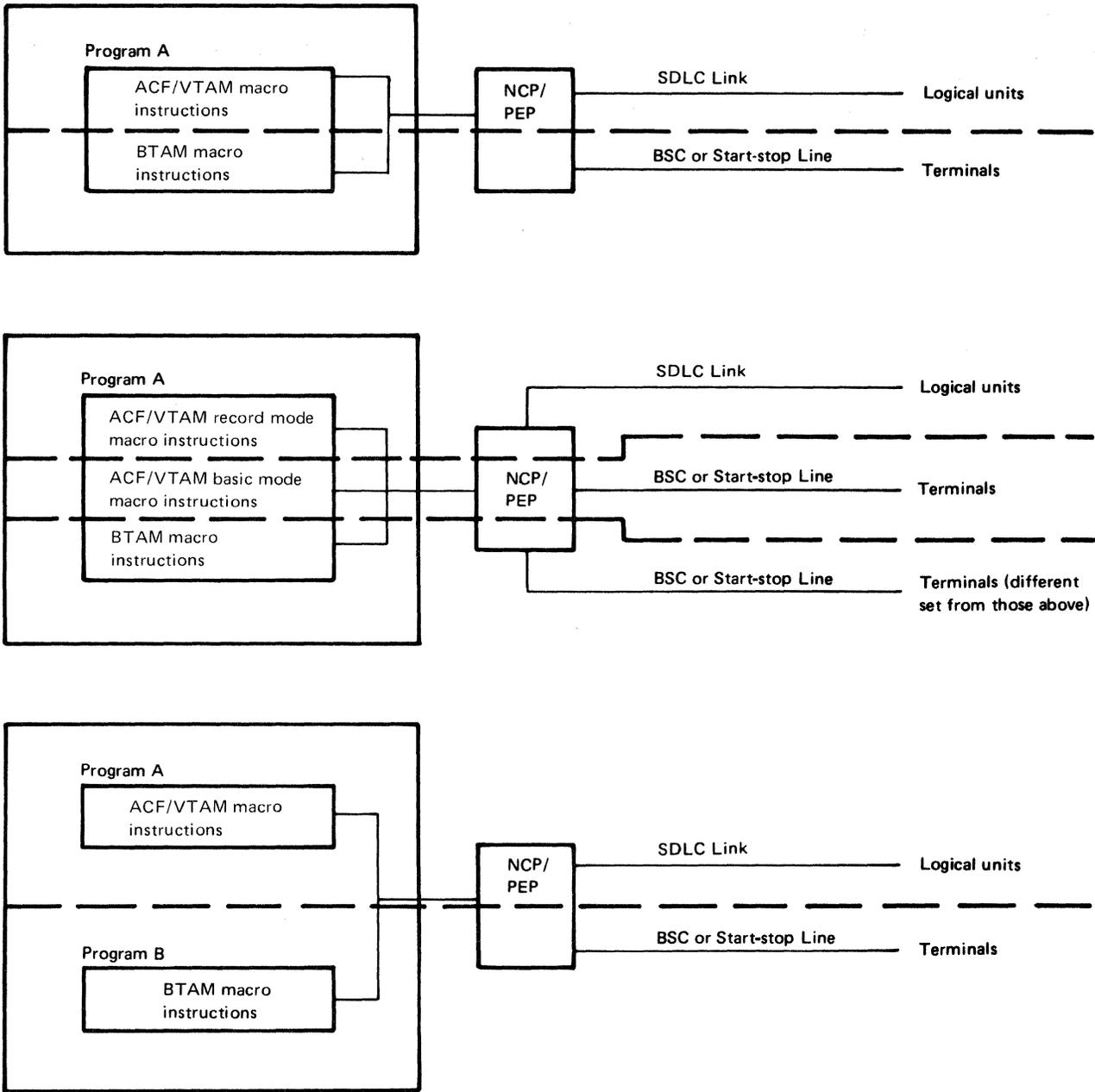
- A network solicitor that controls logons

Characteristic	ACF/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 non-SNA 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>ACF/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 non-SNA terminals</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>Data communication normally separate from processing</p> <p>Can have ACF/VTAM post an ECB or schedule an exit routine when I/O event completes</p>	<p>Data communication 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 ACF/VTAM and BTAM Application Programs

Characteristics	ACF/VTAM Application Program	BTAM Application Program
<i>Functions Provided</i>		
Define line groups	All resources are defined during ACF/VTAM definition	Use DCB or DTFBT macro instruction
Define terminals	All resources are defined during ACF/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 ACF/VTAM	Use error recording macro instructions

Figure 8-1 (Part 2 of 2). Major Similarities and Differences between ACF/VTAM and BTAM Application Programs



Notes: Local and remote 3270 terminals can be communicated with using ACF/VTAM record-mode macro instructions, ACF/VTAM basic-mode macro instructions, or BTAM macro instructions. Using ACF/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 ACF/VTAM to Communicate with Non-SNA Terminals

Interpret tables that can be defined to translate a name in a logon to the name of an ACF/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 an ACF/VTAM application program

Because the BSC and local non-SNA 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 Non-SNA Terminals

In Chapter 3, “Creating an ACF/VTAM System,” the following topics do not apply to non-SNA terminals:

“Defining Terminal-Initiated Connection” (which includes defining USS definition tables and logon mode tables). Instead, see “Defining Terminal-Initiated Logons for Non-SNA 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 “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 non-SNA terminals (unless using record-mode support to communicate with local and BSC 3270 terminals). Instead, see “Communicating with Non-SNA Terminals” in this chapter. See *ACF/VTAM Macro Language Guide*, for examples of program logic using the basic-mode macro instructions. See *ACF/VTAM Macro Language Reference*, for specific terminal considerations when writing an ACF/VTAM application program to communicate with one or more types of non-SNA terminals.

Creating an ACF/VTAM System That Includes Non-SNA 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 ACF/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 ACF/VTAM interpret a logon from a non-SNA terminal.

GROUP, LINE, CLUSTER, TERMINAL, COMP, and VTERM statements that define the network to ACF/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 ACF/VTAM.

The buffer limit for a terminal. (See “Defining ACF/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 ACF/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 non-SNA terminals for logons and passes valid logons to the appropriate application program. Using the network solicitor, a user can permit terminal-initiated logons from non-SNA terminals.

A version of the network solicitor is automatically included in ACF/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 ACF/VTAM’s network operator facilities.

The user can retain this network solicitor, modify it, or replace it. If the ACF/VTAM network solicitor is to be used, the user need only establish logon capabilities for non-SNA terminals as described in “Defining Terminal-Initiated Logons for Non-SNA Terminals” later in this chapter. The network solicitor can be modified through the use of ACF/VTAM’s `NETSOL` macro instruction as explained below.

How the Network Solicitor Works

Figure 8-3 shows what the network solicitor does. 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 Non-SNA Terminals” later in this chapter for a description of defining valid logons.

If the logon is valid, the logon is passed to the appropriate application program, if the application program 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.

If the logon is invalid, or 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.

Terminal operator enters logon from active non-SNA terminal.

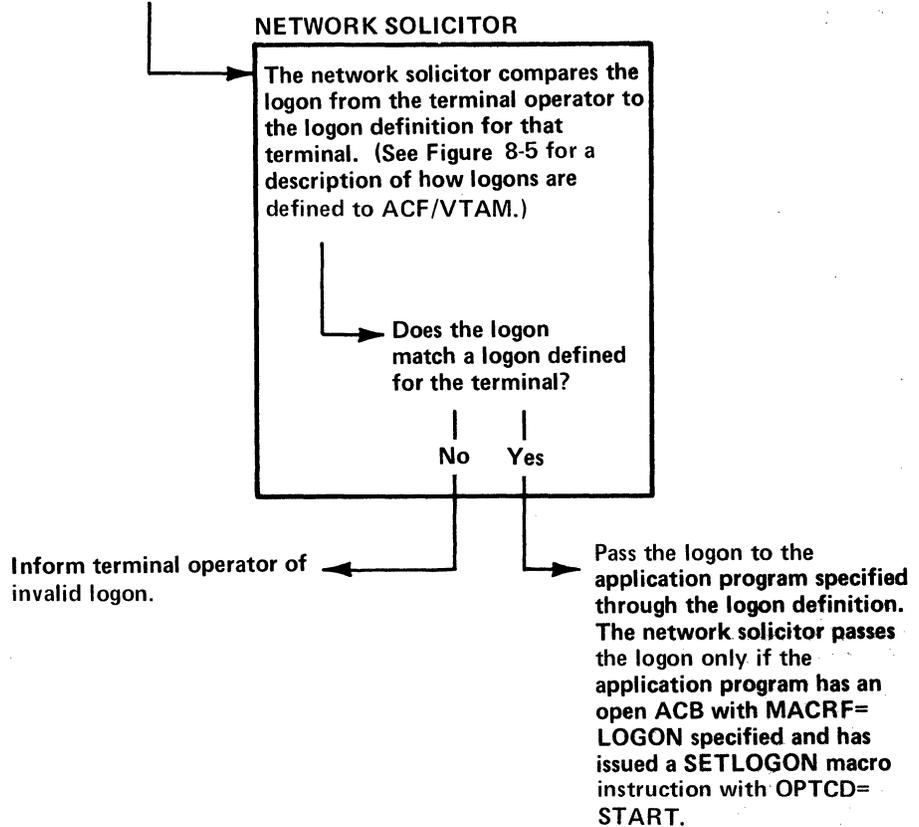


Figure 8-3. Processing a Terminal-Initiated Logon with the Network Solicitor

Modifying the Network Solicitor

The network solicitor can be modified by coding, assembling, and link-editing ACF/VTAM's NETSOL macro instruction. If a NETSOL macro instruction is not used to replace the network solicitor, ACF/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 ACF/VTAM modules in the ACF/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 ACF/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 ACF/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. ACF/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 ACF/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 ACF/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. ACF/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 non-SNA 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 ACF/VTAM; an APPL definition statement must be filed for it, and it must 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

ACF/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 ACF/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

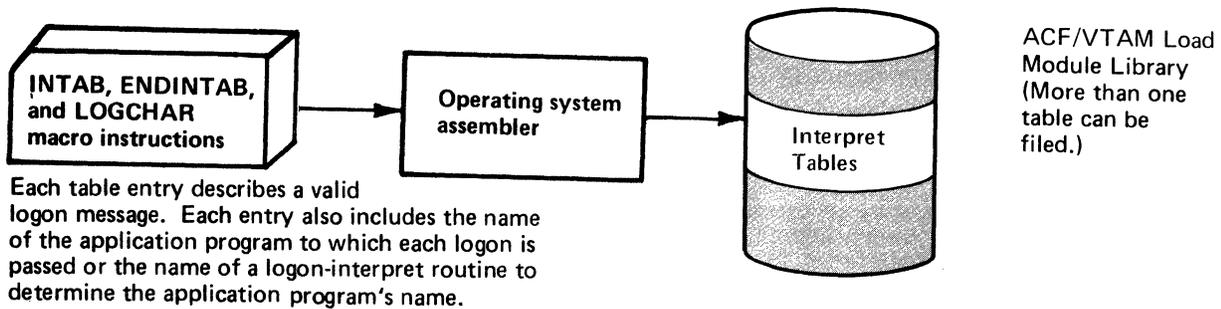


Figure 8-4. Filing Interpret Tables

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 ACF/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 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 ACF/VTAM's INTRPRET macro instruction.

Defining Terminal-Initiated Logons for Non-SNA Terminals

To enable a terminal operator to issue a logon from a non-SNA terminal, the steps are as follows:

1. Modify the network solicitor. (This step is optional.)
2. Define terminal operator logon procedures and sequences to ACF/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 ACF/VTAM definition. Steps 3 and 4 are completed by the network operator, although the degree of network-operator involvement depends upon the ACF/VTAM definition options selected. (See Chapter 4 for details on activating the network solicitor and terminals.) Step 5 is accomplished by the terminal operator.

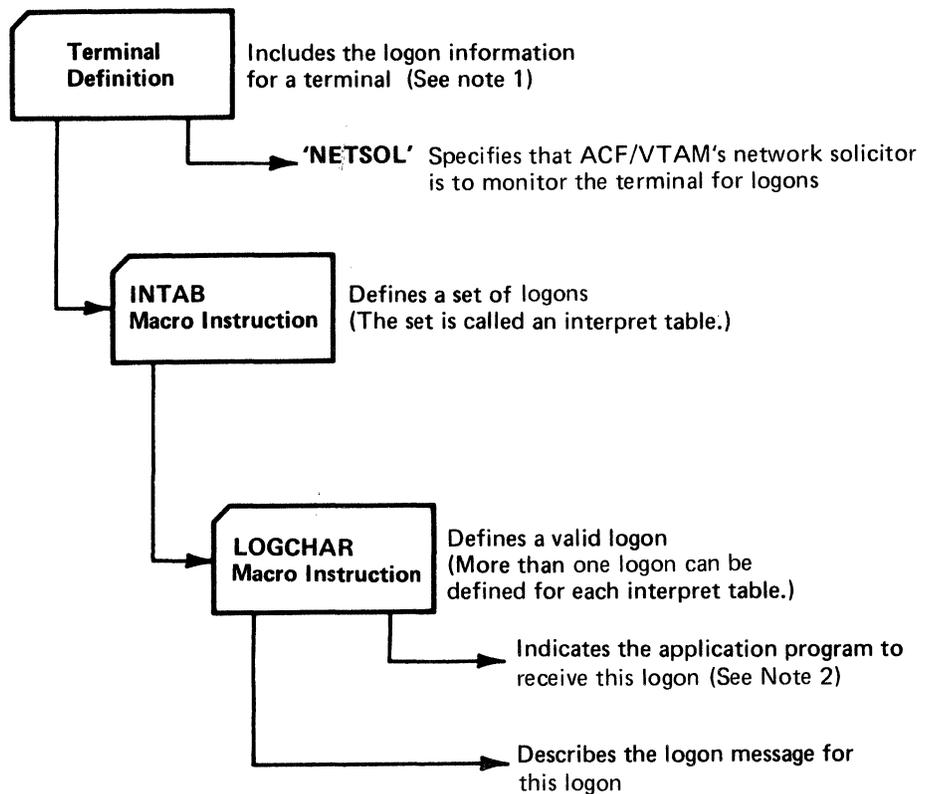
To use the IBM-supplied network solicitor, the user must define:

1. Which terminals are to be handled by ACF/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 ACF/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 ACF/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 ACF/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 ACF/VTAM for non-SNA terminals.



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 logon.

Figure 8-5. Providing Control Information for Processing Logons from Non-SNA Terminals

Defining a Switched Network for Start-Stop and BSC Terminals

This section describes ACF/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.

Call-In Terminals

Call-in (dial-in) terminals have ACF/VTAM-definition requirements that affect some network operator and application program activities. Understanding these effects requires an understanding of ACF/VTAM's support of call-in terminals.

ACF/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 ACF/VTAM.

For ACF/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. 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 ACF/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 ACF/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, ACF/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.

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 ACF/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 ACF/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 ACF/VTAM is to connect unidentified terminals calling in over the line serviced by the port.

Call-Out Terminals

ACF/VTAM has no special requirements for supporting call-out terminals. Using ACF/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, ACF/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 ACF/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 ACF/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 an ACF/VTAM System with Non-SNA Terminals

The operation of an ACF/VTAM network described in Chapter 4 includes the facilities that apply to non-SNA 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 neither 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 handing 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 an ACF/VTAM Application Program

The concepts and facilities described in Chapter 5, "Writing an ACF/VTAM Application Program," except where noted, apply for programs that communicate with non-SNA terminals. A program that communicates with both logical units and non-SNA terminals might be organized so that as much coding as possible is shared. Separate communication routines might be written for individual types of non-SNA terminals. In a program with logon acceptance, ACF/VTAM, as the result of an OPNDST macro instruction, identifies the type of terminal (logical unit or local non-SNA terminal) by setting a value in the DEVCHAR field of the NB 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 ACF/VTAM-supported non-SNA 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.

Basic-Mode Macro Instructions

Here is a brief description of the basic-mode communication macro instructions:

SOLICIT: Requests that ACF/VTAM solicit input from a specific non-SNA terminal or from a group of terminals. Input is read into an ACF/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 ACF/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 an ACF/VTAM buffer as the result of a previous solicit or read operation, data is transferred immediately.

WRITE: Requests that ACF/VTAM transfer data from an application program to a specific terminal. The application program can also request that ACF/VTAM send certain control information to a terminal or write conversationally (write and then read from a terminal).

DO: Requests that ACF/VTAM perform the special I/O action associated with a specific LDO control block in the application program.

RESET: Requests that ACF/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 ACF/VTAM change information contained in a NIB. Using CHANGE quiescens 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," ACF/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 Non-SNA 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, ACF/VTAM initiates whatever actions (such as polling or line preparation) are required to obtain data from the terminal and put it into ACF/VTAM buffers.

Read requests issued in specific-mode cause solicitation if no previously solicited data is in ACF/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 ACF/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.

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, ACF/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.

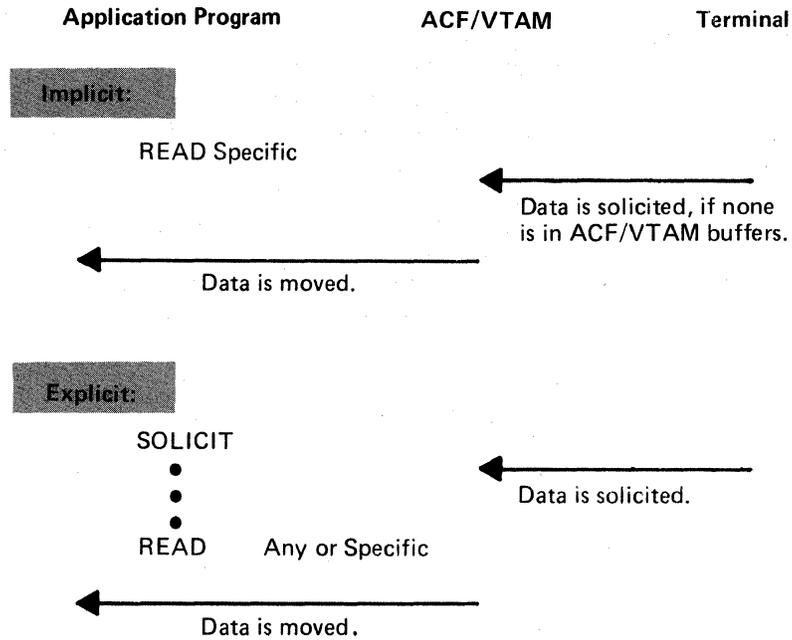


Figure 8-6. Implicit and Explicit Solicitation Using Basic Mode

Solicits a *transmission* from the terminal. For both start-stop and BSC terminals, a transmission ends with an EOT character. Transmissions comprise 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 ACF/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)

- Read the entire contents of any 3270 display station buffer (READBUF 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 certain processing options that ACF/VTAM is to use during subsequent communication with the terminal. These options are specified in the PROC operand of the NIB used to make the connection (the names of the options are shown in parentheses below). The extent of solicitation described above—block, message, transmission, or continuous—is one option. 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):

- ACF/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 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 notify the application program when the attention interruption is detected (MONITOR option). ACF/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 ACF/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 BSC 3270 or Local Non-SNA 3270 Terminals in Record Mode

A BSC 3270 or local non-SNA 3270 is not defined as a logical unit; however, the application program can communicate with it as though it were a logical unit. The restrictions listed in Chapter 5 that apply to the SNA 3270 also apply to the BSC 3270 or local non-SNA 3270 in record mode. For information on special programming considerations when communicating with 3270 terminals in record mode, see Appendix I of *ACF/VTAM Macro Language Reference*.

Appendix A. Supported Terminals

This appendix lists the terminals supported by ACF/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 ACF/VTAM and uses ACF/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, ACF/VTAM receives and transmits data only in extended binary coded-decimal interchange code (EBCDIC). The NCP translates EBCDIC messages from ACF/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 ACF/VTAM. For terminals on SDLC lines, ACF/VTAM receives and transmits data in EBCDIC or any other code. The NCP does no translation. The ACF/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 ACF/VTAM.

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 non-SNA 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.

Start-Stop Terminals

Start-stop terminals can be attached to either a local or a remote communications controller in network control mode. Application programs can communicate with start-stop terminals only through the basic mode of ACF/VTAM.

Binary Synchronous Communications (BSC) Terminals

BSC terminals can be attached to either a local or a remote communications controller in network control mode. Application programs can communicate with BSC terminals, except for 3270s, only through the basic mode of ACF/VTAM. Either the basic mode or the record mode of ACF/VTAM can be used to communicate with 3270s.

Local Non-SNA 3270 Terminals

Application programs can use either the basic mode or the record mode of ACF/VTAM to communicate with local non-SNA 3270 Information Display Systems.

Terminal Support List

Figure A-1 lists the terminals supported by ACF/VTAM. The figure also indicates whether the terminal can be used on switched or nonswitched lines (or both) and indicates the major node in which the terminal is defined.

Device Name	Controlling Device-Model	PU Type	Switched	Non-switched	Major Node in Which Device Is Defined
SNA -- Record Mode Only					
Local					
3270 Information Display System	3274-1A ¹	2			Local SNA
3790 Communication System	3791	2			Local SNA
Remote (SDLC)					
3270 Information Display System	3271-11, 12	1		X	NCP
	3274-1C	2	X	X	NCP or Switched SNA
	3275-11, 12	1		X	NCP
	3276-11, 12, 13, 14	2	X	X	NCP or Switched SNA
3600 Finance Communication System	3601	2		X	NCP
	3614	2		X	NCP
3650 Retail Store System	3651-A50, B50	2	X	X	NCP or Switched SNA
3660 Supermarket System	3651-60	2	X		Switched SNA
3767 Communication Terminal	3767	1	X	X	NCP or Switched SNA
3770 Data Communication System	3771	2	X	X	NCP or Switched SNA
	3773	2	X	X	NCP or Switched SNA
	3774	2	X	X	NCP or Switched SNA
	3775	2	X	X	NCP or Switched SNA
	3776	2	X	X	NCP or Switched SNA
	3777	2	X	X	NCP or Switched SNA
3790 Communication System	3791	2	X	X	NCP or Switched SNA
5937-S01 Industrial Terminal System/32 Batch Work Station	5937 ²	1		X	NCP
	System/32 ³	2	X	X	NCP or Switched SNA
Non-SNA					
Local -- Record or Basic Mode					
3270 Information Display System	3272-1, 2				Local Non-SNA
3270 Information Display System	3274-1B ⁴				Local Non-SNA
Remote Start-Stop -- Basic Mode Only					
1050 Data Communication System	1051-1, 2		X	X	NCP
2740 Communication Terminal	2740-1		X	X	NCP
	2740-2			X	NCP
2741 Communication Terminal	2741-1		X	X	NCP
3767 Communication Terminal	3767-1, 2 ⁵		X	X	NCP
5100 Portable Computer	5100 ⁶		X	X	NCP
Communicating Magnetic Card SELECTRIC® Typewriter	Comm Mag Card SELECTRIC® Type ⁶		X		NCP
System/7 Processor Station	System/7 ⁷				
World Trade Telegraph	W. T. Tel.			X	NCP
AT&T 83B3 Selective Calling Station	AT&T 83B3			X	NCP
Western Union Plan 115A	W. U. Plan 115A			X	NCP
CPT-TWX Terminal	CPT-TWX-33, 35		X		NCP
Remote BSC -- Record or Basic Mode					
3270 Information Display System	3271-1, 2			X	NCP
	3274-1C ⁸			X	NCP
	3275-1, 2			X	NCP
	3276-1, 2, 3, 4 ⁸			X	NCP

Figure A-1 (Part 1 of 2). Devices Supported by ACF/VTAM

Device Name	Controlling Device-Model	PU Type	Switched	Non-switched	Major Node in Which Device Is Defined
Non-SNA (Cont.)					
Remote BSC -- Basic Mode Only					
2770 Data Communication System	2772		X	X	NCP
2780 Data Transmission Terminal	2780		X	X	NCP
2980 General Banking Terminal System (US only)	2972			X	NCP
3735 Programmable Buffered Terminal	3735		X	X	NCP
3740 Data Entry System	3741-1 ⁹		X	X	NCP
3750 Switching System	3751			X	NCP
3770 Data Communication System	3771-1, 2 ¹⁰		X	X	NCP
	3773-1, 2 ¹⁰		X	X	NCP
	3774-1 ¹⁰		X	X	NCP
	3775-1 ¹⁰		X	X	NCP
	3776-1 ¹¹		X	X	NCP
3780 Data Communications Terminal	3780 ¹²		X	X	NCP
5275 Direct Numerical Control Station (US only)	5275-1 ¹³			X	NCP
5937-S01 Industrial Terminal	5937 ¹³			X	NCP
System/3 Central Processing Unit	System/3		X	X	NCP
System/7 Processor Station	System/7 ¹⁴		X	X	NCP
System/32 Batch Work Station	System/32 ¹⁴		X	X	NCP
System/370	System/370 (115-168)		X	X	NCP
<p>1 Supported as a 3791</p> <p>2 Supported as a 3275-11 or 12</p> <p>3 Supported as a 3770</p> <p>4 Supported as a 3272-1, 2</p> <p>5 Supported as a 2740-1, 2, or 2741</p> <p>6 Supported as a 2741</p> <p>7 Supported as a 2740-1</p> <p>8 Supported as a 3271-1, 2</p> <p>9 Supported as a System/3</p> <p>10 Supported as a 2772</p> <p>11 Supported as a 2772 or 3780</p> <p>12 Supported as a 2770</p> <p>13 Supported as a 3275-1, 2</p> <p>14 Supported as a System/3</p>					

Figure A-1 (Part 2 of 2). Devices Supported by ACF/VTAM

Appendix B. Remote Station Versus Remote Controller

Because ACF/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 an ACF/VTAM network, there are three 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. Another way is to connect two independent, local communication controllers. The connection of local communications controllers is depicted in Figure B-2. A third way is to connect two local communications controllers to form a multiple-domain network. This Appendix does not describe the third method.

In Figure B-1, both ACF/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 ACF/VTAMs) is viewed by the other as a single terminal. In fact, there are two 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, ACF/VTAM in CPU A can directly address any terminal attached to communications controller B. Figure B-2 depicts a network in which the ACF/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 ACF/VTAM. The application program must request connection, using ACF/VTAM facilities, to terminal C. After the connection is completed, the application program must request that ACF/VTAM transmit the message to the terminal.

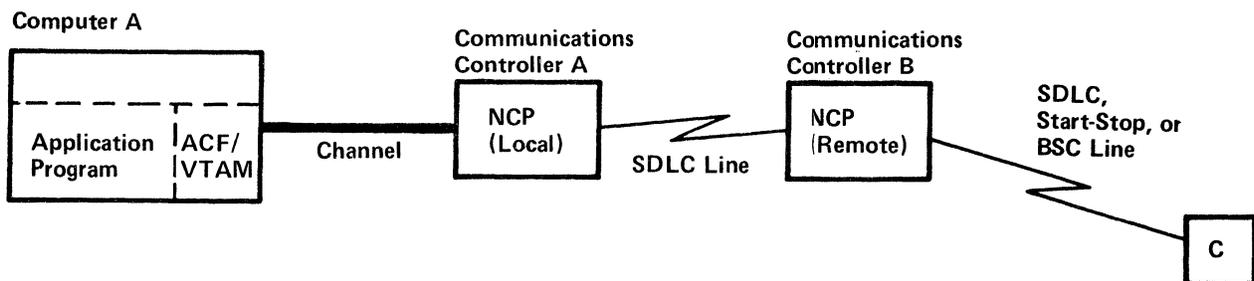
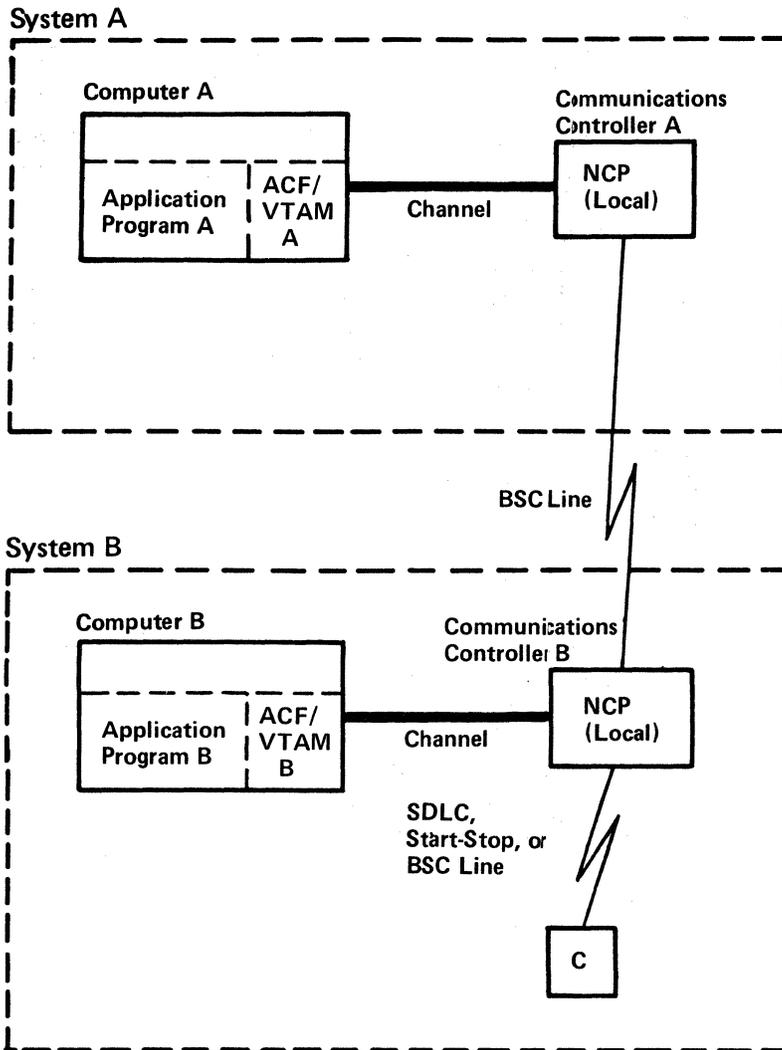


Figure B-1. A Remote Communications Controller



Note: ACF/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 ACF/VTAM in A.

Figure B-2. Communications Controllers Attached as Part of a Remote Station

Upon receiving the request for data transmission, ACF/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.

Application program A first requests ACF/VTAM for connection with the terminal that is system B (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 ACF/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 ACF/VTAM A transmit the message to the connected terminal (system B). ACF/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 ACF/VTAM B for input from the terminal defined as system A. Upon receiving the message from controller A, communications controller B transmits it to ACF/VTAM B. ACF/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 ACF/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 ACF/VTAM and a logical unit or between two application programs. 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 or commands and indicators or both. A response can include:

The type of response, that is, a definite response 1 or a definite response 2 or both, and also a positive or negative response indicator.

An explanation of the response if it is negative.

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 Command or Indicator: Specifies the general category of the commands or indicator.

Command or Indicator: 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.

This table contains the following types of indicators and commands:

- Normal-flow
- Expedited-flow
- Change-direction
- Bracket
- SESSIONC

Type of Command or Indicator	Command or Indicator	Function
Change-Direction Indicators	Change Direction Command (CMD)	Tell the receiver that it is now its turn to send
	Change Direction Request (REQ)	Request the receiver to send a Change Direction Command indicator
Bracket Indicators	Begin Bracket (BB)	Begin a bracket
	End Bracket (EB)	End a bracket
SESSIONC Commands (Session Control Commands)	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
	Request Recovery (RQR)	Tell the primary end of the session that a recovery procedure is required
	Start Data Traffic (SDT)	Allow the sending of normal-flow or expedited-flow messages and responses
	Set and Test Sequence-Number (STSN)	Exchange information with the logical unit so that sequence numbers can be determined and/or reset
	Bind (Negative response)	Tell the primary end of the session that the request for connection is rejected
Normal-Flow Commands	Bracket Initiation Stopped (BIS)	Tell the receiver that the sender will not begin any new bracket
	Bid	Bid to begin a bracket
	Cancel	Tell the receiver to purge chain elements already received
	Chase	Determine that the receiver has no more responses to send
	Logical Unit Status (LUS)	Inform the receiver of an unexpected condition
	Quiesce Complete (QC)	Tell the receiver that the sender is now quiesced and will not send until released
	Ready to Receive (RTR)	Tell the receiver that a bracket has ended and a request to begin one can be sent
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
	Release Quiesce (RELO)	Permit the receiver to send now
	Request Shutdown (RSHUTD)	Request the primary end of the session to disconnect
	Shutdown Complete (SHUTC)	Tell the receiver that preparation for shutdown is complete
	Shutdown (SHUTD)	Tell the secondary end of the session to prepare to shut down
	Signal	Pass a 4-byte message with an agreed-upon meaning
	Stop Bracket Initiation (SBI)	Request the receiver to refrain from starting any new bracket

Appendix D. Considerations in Moving from VTAM Level 2 to ACF/VTAM

This appendix lists limitations, restrictions, and other factors to be taken into account by users who are moving from VTAM Level 2 to ACF/VTAM.

Use of the System Management Program to Install ACF/VTAM

When the System Management Program (SMP) is used to install a programming feature (such as ACF/VTAM or the Multisystem Networking Facility), the feature cannot be easily removed. To delete the feature, the user must perform another system generation and reapply features.

Configuration Considerations

Connection of Domains: One domain can be connected to another domain only (1) by a nonswitched link (or switched backup link) between two local 3705 Communications Controllers or (2) by sharing a communications controller that is channel-attached to a host computer in each domain.

Unsupported Connections: Within a single domain, ACF/VTAM does not support physical connections from a remote communications controller to another remote communications controller.

ACF/VTAM does not support the following types of cross-domain physical connections:

- Channel-to-channel connection between two host computers

- A local communications controller in one domain to a remote communications controller in another domain

- A remote communications controller in one domain to a remote communications controller in another domain

- Multiple active links between a local communications controller in one domain and a local communications controller in another domain.

Units Eligible for Cross-Domain Sessions: With the Multisystem Networking Facility in both domains, ACF/VTAM allows cross-domain sessions between application programs and remote SNA terminals and specially defined remote BSC 3270 terminals. Channel-attached (local), start-stop, and BSC terminals (except for specially defined remote BSC 3270 terminals) cannot participate in cross-domain sessions. ACF/VTAM application programs can also communicate with each other across domain boundaries.

Relationship between ACF/VTAM and ACF/TCAM: A TCAM application program formerly could communicate with some terminals by having messages passed to VTAM through an interface.

Starting with TCAM 10 and continuing with ACF/VTAM, TCAM application programs must communicate with all terminals directly through TCAM or ACF/TCAM.

Definition Considerations

NCP Generation: An NCP generated for VTAM Level 2 must be respecified and regenerated for ACF/VTAM. The new NCP generation decks must be filed in the ACF/VTAM definition library.

Automatic Logon Specifications: The application program named in the LOGAPPL operand of a definition statement can be in the same domain as the terminal (or terminals) being defined, or it can be in another domain. Automatic logon can be specified only for SNA logical units or specially defined remote BSC 3270 devices.

Bidirectional Pacing: In ACF/VTAM, pacing controls have been extended to include:

- Pacing in a session between two application programs either in the same domain or in different domains

- Pacing in a session between an application program in one domain and a terminal in another domain

In addition, bidirectional pacing is introduced, providing for control over the rate at which the secondary end of a session can send data to the primary end. (Pacing from the primary end to the secondary end existed in VTAM Level 2.)

PACING and VPACING Parameters: In VTAM Level 2, two parameters are specified for PACING and VPACING:

- n*-value, which indicates the number of elements to be sent before the transmitting node waits for a pacing response

- m*-value, which indicates in which element the request for a pacing response is to be placed

In ACF/VTAM, only the *n*-value is specified. ACF/VTAM assumes an *m*-value of 1. If the *m*-value is specified, it is ignored, and the value 1 is used. Consequently, any tuning in VTAM Level 2 that is based on the *m*-value is no longer valid in ACF/VTAM.

Use of the BFRPAD Parameter on the Host Statement: In moving from VTAM Level 2 to ACF/VTAM, the value specified in the BFRPAD parameter of the HOST statement must be changed. For VTAM Level 2, the proper BFRPAD values are 15 for DOS/VS and 28 for OS/VS. For ACF/VTAM, the value must be changed to 0. If the BFRPAD value is not 0, activation of an NCP by ACF/VTAM will fail, and an error message will be issued to indicate that the BFRPAD value is not correct.

Use of the BUFLIM and BUFFACT Parameters: For ACF/VTAM, the BUFLIM and BUFFACT parameters in definition statements apply only to basic-mode terminals. The values have no meaning for record-mode terminals.

Use of the CLUSTER Statement: With ACF/VTAM, the CLUSTER statement cannot be used to define an SDLC cluster control unit; it can be used only for a non-SDLC cluster control unit. For ACF/VTAM, an SDLC cluster control unit is defined with the PU statement.

Use of the INNODE Statement: In defining a network control program for use with ACF/VTAM, the NCP's INNODE statement cannot be used. The information is included in the PU statement instead. If the INNODE statement is coded, it is rejected during NCP generation.

Use of Interpret Tables and the INTRPRET Macro Instruction: For ACF/VTAM, an interpret table must be defined and stored in the domain in which the terminal (or terminals) related to the table is located. When the terminal enters a logon for an application program in another domain, the interpretation of the logon is performed by the ACF/VTAM in the domain in which the terminal is located, and the results are passed to the ACF/VTAM in the other domain.

An ACF/VTAM application program in one domain cannot issue an INTRPRET macro instruction that calls for use of an interpret table in another domain.

PATH Statements: In working with ACF/NCP/VS and ACF/VTAM, the user is working with three different PATH statements with different purposes, as follows:

An NCP PATH statement that provides routing information to the NCP

An ACF/VTAM PATH statement that defined a dial-out path to a physical unit in a switched SNA major node

A different ACF/VTAM PATH statement that provides routing information to ACF/VTAM for cross-domain communications

Differentiating between the Old *bth* Parameter and the New *slowpt* Parameter in Defining ACF/VTAM Buffer Pools: The old *bth* parameter and the new *slowpt* parameter serve similar functions (they define the point at which the pool is to enter slowdown processing), but the parameters represent different values. For VTAM Level 2, the *bth* parameter specifies the slowdown threshold in terms of the number of buffers that *are in use*. For ACF/VTAM, the *slowpt* parameter specifies the threshold in terms of the number of buffers remaining available in the pool (that is, the number of buffers *not in use*).

ISTATUS Operand on a LINE Statement: In VTAM Level 2, the ISTATUS operand in a LINE statement does not apply to the line being defined by the statement; it is used for sifting purposes and applies only to the network elements defined by statements subordinate to and following the LINE statement. In ACF/VTAM, the ISTATUS operand in the LINE statement applies to the line itself as well as to the subordinate network elements.

Operator Considerations

Control of Domains: A network operator can only control resources in his own domain; the operator has no control over resources in other domains.

Use of VARY LOGON: The following restrictions apply to use of the VARY LOGON command:

The network operator *can* specify that an automatic logon relationship is to be established between one or more SNA terminals in the operator's domain and an application program either in the operator's domain or in another domain.

The network operator *cannot* specify automatic logon for a terminal or major node that is in another domain.

No automatic logon relationship can be established between one application program and another application program.

Application Program Considerations

ACF/VTAM-ACF/TCAM Considerations: When ACF/VTAM and ACF/TCAM are being used in the same network, the following factors should be considered:

The VARY LOGON command cannot be used to create an automatic logon relationship between an ACF/VTAM-owned device-type terminal and an ACF/TCAM application program.

A CLSDST operation with OPTCD=PASS will be successful only if (1) the old application program (the one issuing the CLSDST PASS), the new application program, and the terminal are all in ACF/VTAM domains, or (2) the old application program and the new application program are both in the same ACF/VTAM domain and the terminal is in an ACF/TCAM domain. The operation will fail if (1) ACF/TCAM owns the new application program, or (2) ACF/TCAM owns the terminal and the old and new application programs are in different ACF/VTAM domains.

The only non-SNA terminals owned by ACF/TCAM that can log on to an ACF/VTAM application program are BSC 3275s, 3277s, 3284s, and 3286s.

A SIMLOGON request with the Q option will fail (that is, will not be queued) when the desired terminal is owned by ACF/TCAM and is already in session.

INQUIRE APPSTAT cannot be used in an ACF/VTAM application program when the named application program is owned by ACF/TCAM.

Use of the INQUIRE Macro Instruction for a Cross-Domain Resource: For a resource in another domain, the following three forms of the INQUIRE macro instruction can be issued only when there is a pending logon from the resource:

INQUIRE OPTCD=DEVCHAR

INQUIRE OPTCD=TERMS

INQUIRE OPTCD=SESSPARM

That is, one of those forms can be used only after the LOGON exit routine has been scheduled and before the OPNDST or CLSDST macro instruction is issued to accept or reject the logon.

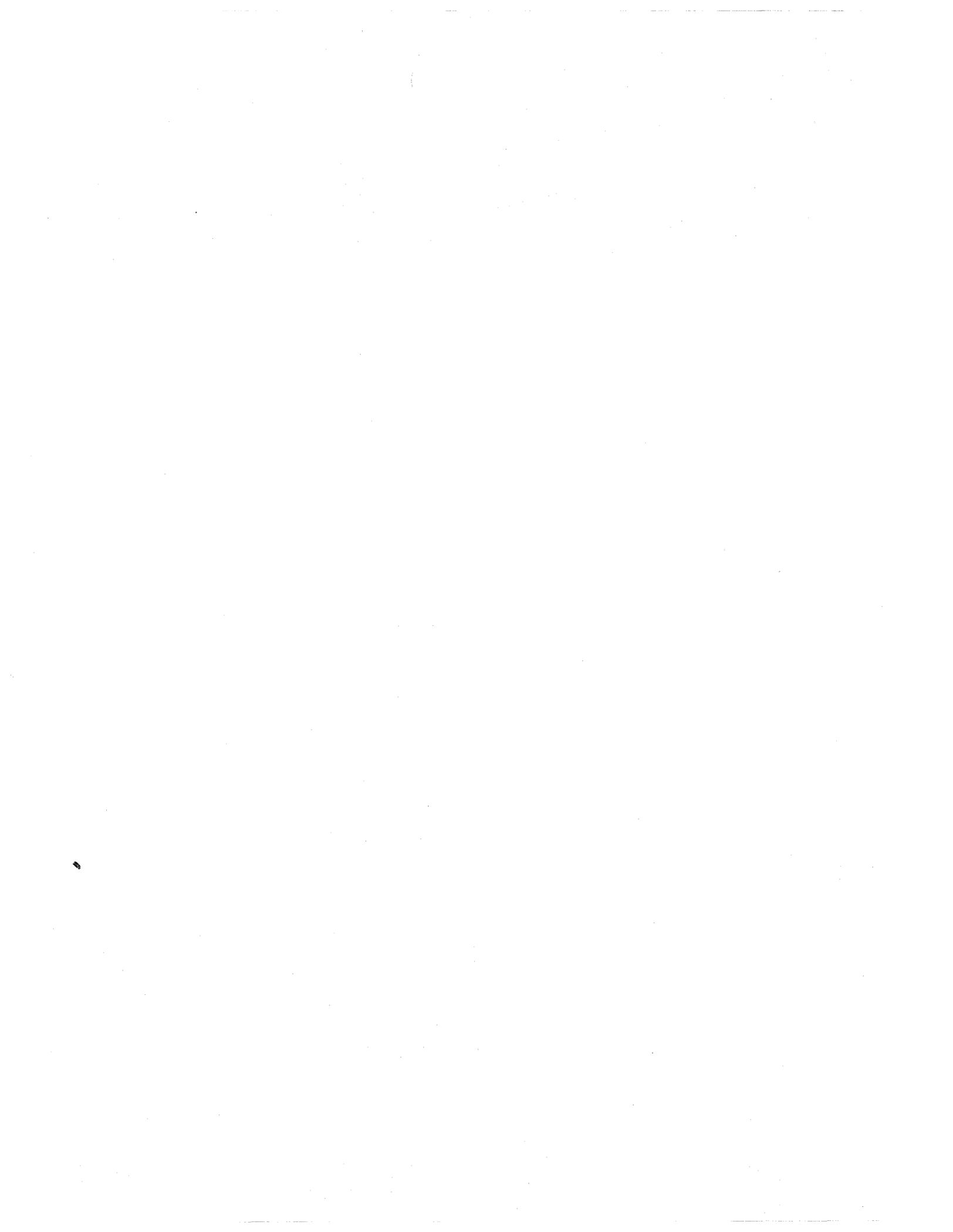
Also, as mentioned above, INQUIRE APPSTAT cannot be used in an ACF/VTAM application program when the named application program is owned by ACF/VTAM.

Use of Logon Mode Names with Session Initiation Requests: When a session initiation request involves a terminal or application program in another domain, logon mode names are specified as follows:

When an OPNDST macro instruction with OPTCD=ACCEPT is issued to accept a connection from a terminal in another domain, no logon mode name can be present in the LOGMODE field of the NIB (pointed to by the RPL specified in the request). For an OPNDST ACCEPT for a cross-domain resource, the LOGMODE field of the NIB can contain 0 (to specify that the Bind command is to contain the session parameters that accompanied the logon), or the application program can put an address in the BNDAREA field of the NIB (to specify that the session parameters starting at that address are to be used in the Bind command).

When an OPNDST macro instruction with OPTCD=ACQUIRE, a SIMLOGON macro instruction, or a CLSDST macro instruction with OPTCD=PASS is issued for a terminal in another domain, the LOGMODE field of the NIB *can* contain a logon mode name. The name must be the name of a logon mode table entry in the logon mode table that is associated with the terminal in the terminal's domain.

When a REQSESS macro instruction is issued to ask for a connection with a primary application program in another domain, the LOGMODE field of the NIB *can* contain a logon mode name. The name must be the name of a logon mode table entry in the logon mode table associated with the application program that is issuing the REQSESS macro instruction. The session parameters designated by the logon mode name are sent with the logon to the domain of the primary application program.



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 ACF/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 ACF/VTAM, to connect a terminal to a primary application program as the result of a logon. The logon may be originated by the terminal, the network operator, another primary application program, or ACF/VTAM. Contrast with *acquire* (1).

access method control block (ACB). A control block that links an application program to VSAM or ACF/VTAM.

accounting exit routine. In ACF/VTAM, an optional, user-written routine that collects statistics about connections and disconnections in the communication network.

ACF. Advanced Communications Function.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method.

ACF/VTAM application program. A program that has opened an ACB to identify itself to ACF/VTAM. It can now issue ACF/VTAM macro instructions.

ACF/VTAM definition. The process of defining the communication network to ACF/VTAM (which is called "network definition") and modifying IBM-defined characteristics to suit the needs of the user.

ACF/VTAM definition library. The DOS/VS files or OS/VS data sets that contain the definition statements and start options filed during ACF/VTAM definition.

ACF/VTAM system. The resources defined to and controlled by ACF/VTAM.

acquire. (1) In relation to an ACF/VTAM application program, to connect a terminal to the application program in the absence of a logon. The connection occurs at the primary application program's initiative. Contrast with *accept*. (2) In relation to ACF/VTAM resource control, to take over resources (communications controllers or physical units) that were formerly controlled by a data communication access method in another domain, or to assume control of resources that were controlled by this domain but released. Contrast with *release*. See also *resource takeover*.

active. Pertaining to a major node that has been made known to ACF/VTAM by operator command and is available for use or pertaining to a minor node that is connected to, or available for connection to, an ACF/VTAM application program. Contrast with *inactive*.

adjacent domain. A domain that is physically connected to another domain by a single cross-domain link or by a shared local communications controller.

adjacent node. A node that is physically connected to another node by a single data link.

Advanced Communications Function (ACF). A group of program products for users of DOS/VS and OS/VS that can provide improved single-domain and, optionally, multidomain data communication capability.

Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM). A program product that provides improved single-domain data communication capability and, optionally, multidomain capability.

any-mode. In ACF/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 an application program is identified to ACF/VTAM. It is specified in the APPLID parameter of the ACTS macro instruction. It corresponds to the ACBNAME parameter in the APPL statement or, if the ACBNAME is defaulted, to the name of the APPL statement.

application program major node. In ACF/VTAM, a member (OS/VS) or book (DOS/VS) of the ACF/VTAM definition library that contains one or more APPL statements, each representing an application program.

APPLID routine. Synonym for *logon-interpret routine*.

asynchronous operation. In ACF/VTAM, an operation such as a connection or data transfer in which the application program is allowed to continue execution while ACF/VTAM performs the operation. ACF/VTAM interrupts the program as soon as the operation is completed.

asynchronous request. In ACF/VTAM, a request for an asynchronous operation.

authorization exit routine. In ACF/VTAM, an optional, user-written routine that approves or disapproves requests for connection and disconnection.

authorized path. In ACF/VTAM for OS/VS2 MVS, a facility that enables an authorized application program to specify that a data transfer or related operation be carried out in a faster manner than usual.

automatic logon. A process by which ACF/VTAM creates a logon for a terminal or logical unit to a designated application program whenever the terminal or logical unit is not connected to another program. Specifications for the automatic logon can be made when the terminal or logical unit is defined or can be made by the network operator in the VARY LOGON command. See also *controlling application program*.

available. In ACF/VTAM: (1) Pertaining to a terminal that supports only one session, 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 ACF/VTAM, a mode of data transfer in which the application program can communicate with non-SNA terminals. Contrast with *record mode*.

block. In the basic mode of ACF/VTAM, a unit of data that is transmitted between an ACF/VTAM application program and a terminal.

bracket. In ACF/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 inquiries/replies, 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 logical unit designated at session initiation as the first speaker, and the other logical unit as the bidder. The bracket protocol involves bracket initiation and termination rules.

C

cancel closedown. A closedown in which ACF/VTAM is abnormally terminated as the result of an operator command.

CDRSC. Cross-domain resource.

CDRM. Cross-domain resource manager.

change-direction protocol. In ACF/VTAM, a method of communication in which the sender stops sending on its own initiative, having signaled this fact to the receiver on the last request sent, and prepares to receive.

character-coded. In ACF/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 *cancel closedown*, *orderly closedown*, and *quick 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 is attached to a host computer only through a communications controller. A local cluster control unit is attached through a channel. 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), is a command, while Quiesce Complete is the reply. (3) In SNA, a data flow control or session control request that may be sent or received by an application program using record mode.

communication control character. *A control character intended to control or facilitate transmission of data over communication networks.

communication identifier (CID). In ACF/VTAM, a key for locating the control blocks that represent a session. The key is created during the session establishment procedure 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 ACF/VTAM, the facility for immediate recovery after a failure in the NCP or communications controller or after a loss of contact with a physical unit or logical unit, or for delayed recovery after a failure or deactivation of a major node, ACF/VTAM, or the host computer. 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 one or more VSAM data sets in which ACF/VTAM keeps a record of changes to initial configuration data.

connection. (1) In ACF/VTAM, the linking of control blocks in such a way that an application program is in session with a terminal. Connection includes establishing and preparing the network path between the program and the terminal. (2) A physical capability of communicating between two end points. Also called *physical connection*. See also *queued for connection*.

continue-any mode. In ACF/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 ACF/VTAM, a state into which a terminal is placed that allows its input to satisfy only input requests issued in specific-mode.

controlling application program. An application program to which a terminal (other than a secondary application program) is automatically logged on whenever the terminal is active and available. See also *automatic logon*.

conversational write operation. In the basic mode of ACF/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 ACF/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*.

cross-domain. Pertaining to control or resources involving more than one domain.

cross-domain link. A data communication line physically connecting two domains. See also *local-to-local link*.

cross-domain resource (CDRSC). A resource owned by another domain but known in this domain by name and associated cross-domain resource manager.

cross-domain resource manager (CDRM). The portion of the system services control point (SSCP) that controls cross-domain sessions.

cross-domain session. A session between network addressable units in different domains.

D

data communication. The transmission, reception, and validation of data.

data flow. In SNA, any of four flows in a given session, characterized as either primary-to-secondary or secondary-to-primary, each of which may be normal or expedited.

data flow control. In SNA, a set of protocols and control functions used by the network addressable unit to assist in controlling the flow of requests and responses within a session. Contrast with *session control*.

data flow control protocol. In SNA, the sequencing rules for requests and responses by which network addressable units in a communication network coordinate and control data transfer and other operations. For example, see *bracket protocol*.

data link control protocol. A set of rules used by two nodes on a data link to accomplish an orderly exchange of information. Synonymous with *line discipline*.

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.

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* and *no response*.

definition statement. In ACF/VTAM, the means of describing an element of the communication network.

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.

device-type logical unit. A logical unit residing in a physical unit other than a host computer.

disconnection. (1) In ACF/VTAM, the dissociation of control blocks in such a way as to end a session between an application program and a connected terminal. The disconnection process includes suspending the use of the network path between the program and the terminal. **(2)** A physical dissociation between two end points.

domain. In a data communication system, the portion of the total network that is controlled by the SSCP in one telecommunication access method.

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. See *exception request*.

exception request. In communicating with a logical unit, a message that indicates an unusual condition such as a sequence number being skipped. When ACF/VTAM detects such a condition, it notifies the application program. ACF/VTAM or the 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 ACF/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 ACF/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 ACF/VTAM, a type of 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 flow (normal or expedited) are usually 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*.

external domain. A domain controlled by a different system services control point (SSCP).

F

field-formatted. In ACF/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) In a data processing system that includes ACF/VTAM or ACF/TCAM, the computer in which ACF/VTAM or ACF/TCAM 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 ACF/VTAM, pertaining to a major node that has not been made known to ACF/VTAM and is unavailable for use, or pertaining to a minor node that is not connected to nor available for connection to an application program. Contrast with *active*.

intermediate node. In SNA, a physical unit that is capable of routing path information units to another subarea.

interpret table. In ACF/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.

leading graphics. From one to seven graphic characters that may accompany an acknowledgment sent to or from a BSC terminal in response to the receipt of a block of data.

line. See *communication line*.

line control. The scheme of operating procedures and control signals by which a communication network is controlled.

line discipline. Synonym for *data link control protocol*.

line group. A collection of one or more communication lines of the same type.

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 NCP. An NCP that is channel-attached to a host computer. Contrast with *remote NCP*.

local non-SNA major node. In ACF/VTAM, a major node whose minor nodes are locally attached non-SNA terminals.

local SNA major node. In ACF/VTAM, a major node whose minor nodes are locally attached physical and logical units.

local-to-local link. A data communication link between two local communications controllers. The link can be either a cross-domain link (communications controllers in different domains) or it can exist within a domain between local communications controllers controlled by the same system services control point.

local 3270 major node. See *local non-SNA major node*.

logical device order (LDO). In ACF/VTAM, a set of parameters that specify a data-transfer or data-control operation to local non-SNA 3270 Information Display Systems and certain kinds of start/stop or BSC terminals.

logical error. In ACF/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. ACF/VTAM application programs must communicate with logical units in record mode. See also *physical unit, system services control point*.

log off. In ACF/VTAM, to request that a terminal be disconnected from an application program.

logoff. In ACF/VTAM, a request that a terminal be disconnected from an application program.

log on. In ACF/VTAM, to request that a terminal be connected to an application program.

logon. In ACF/VTAM, a request that a terminal be connected to an application program. See also *automatic logon* and *simulated logon*.

logon data. In ACF/VTAM: (1) The data portion of a field-formatted or character-coded logon from an SNA terminal or from a non-SNA 3270 terminal for which PU=YES has been specified. (2) The entire logon sequence or message from a non-SNA terminal.

logon-interpret routine. In ACF/VTAM, a user-written exit routine associated with a logon-interpret table entry that translates logon data. It may also verify the logon. Synonymous with *APPLID routine*.

logon message. Synonym for *logon data*.

logon mode. In ACF/VTAM, the communication protocols that govern a session between a logical unit and an ACF/VTAM application program or between two application programs. Synonymous with *session parameters*.

logon mode name. In ACF/VTAM, the symbolic representation of a logon mode.

logon mode table. In ACF/VTAM, a set of macro-generated constants making up one or more logon modes. Each logon mode is associated with a logon mode name.

LU-LU session. In SNA, a session between two logical units in the network. It allows communication between two end users, each associated with one of the logical units.

M

major node. In ACF/VTAM, a set of minor nodes that is filed as a member or book of a definition data set and 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) A combination of characters and symbols transmitted from one point to another. (5) In SNA, a request/response header and its associated request/response unit. In some ACF/VTAM publications, a distinction is made between messages, responses, and commands, where "message" is used to mean a data request.

minor node. In ACF/VTAM, a uniquely-defined resource within a major node that can be activated or deactivated by the VARY command. Synonymous with *specific node*. See also *major node*.

MTA. Multiple terminal access.

multiple-channel-attached communications controller. A communications controller that can be channel-attached to more than one host computer.

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.

Multisystem Networking Facility. In ACF/VTAM, a feature that supports communication among multiple host computers operating with DOS/VS, OS/VS1, and OS/VS2 (SVS and MVS).

multithread application program. An ACF/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 ACF/VTAM, a major node defined through NCP generation.

negative response. A response indicating that a request did not arrive successfully or was not processed successfully by the receiver in a session. Synonymous with *exception response*. Contrast with *positive response*.

network. (1) (SC1) The assembly of equipment through which physical connections are made between terminal installations. (2) In data communication, a configuration in which two or more locations are physically connected for the purpose of exchanging data.

network control mode. The functions of a network control program that enable it to direct a communications controller to perform 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 a 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 ACF/VTAM, the process of defining the identities and characteristics of each node in the network and the arrangement of the nodes. Network definition is part of ACF/VTAM definition.

network operator. (1) A person responsible for controlling the operation of a communication network. (2) An ACF/VTAM application program authorized to issue network operator commands.

network operator command. A command used to monitor or control the communication network.

network operator console. A system console or terminal in the network from which a network operator controls a communication 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.

no response. In SNA, an indication in the RH for a request unit that no response is to be returned to the request, whether or not it is received and processed successfully. Contrast with *definite response* and *exception response*.

node. (1) An addressable point in a data communication network. (2) In ACF/VTAM, a point in a communication network defined by a symbolic name. See also *major node* and *minor node*.

node initialization block (NIB). In ACF/VTAM, a control block associated with a particular terminal that contains information used by the application program to identify the terminal and indicate how communication requests directed at the terminal are to be processed.

node name. In ACF/VTAM, the symbolic name assigned to a specific major or minor node during network definition.

non-SNA terminal. A terminal supported by ACF/VTAM that uses start-stop or BSC protocol or that is part of a local non-SNA 3270 Information Display System.

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 flow (normal or expedited) are usually 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 ACF/VTAM and the communication network. An orderly shutdown does not take effect until all application programs have been disconnected from ACF/VTAM. Until then, all data transfer operations continue. Contrast with *cancel shutdown* and *quick shutdown*.

P

padding. In data communication, a technique by which a receiving connection point manager or boundary controls the rate of transmission of a sending function connection point manager to prevent overrun.

partitioned emulation programming (PEP). A feature of the network control program, versions 2 and later, 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 communication network.

path. (1) In ACF/VTAM, the intervening nodes and data links connecting a terminal and an application program in the host computer. (2) In defining a switched SNA major node, a potential dial-out port that can be used to reach a physical unit. (3) In defining ACF/VTAM or ACF/NCP routing tables, a route through an adjacent subarea to one or more destination subareas. (4) 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.

peer NCP. An NCP that is attached to another NCP through a local-to-local link. Contrast with *remote NCP*. See also *local-to-local link*.

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 request was received and processed successfully. Contrast with *negative response*.

primary application program. In a session, an application program that adheres to predefined primary protocols. Contrast with *secondary application program*.

primary end of a session. A network addressable unit (for example, a primary application program) that adheres to predefined primary protocols.

program operator. An ACF/VTAM application program that is authorized to issue network operator commands and receive ACF/VTAM network operator awareness messages. See also *solicited messages* and *unsolicited messages*.

protocol. A set of rules used by the network entities to accomplish an orderly exchange of information and control. See also *data flow control protocol*.

Q

queued for connection. In ACF/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 ACF/VTAM, a closedown in which current data-transfer operations are completed, while new connection and data-transfer requests are canceled. Contrast with *cancel closedown* and *orderly closedown*.

quiesce protocol. In ACF/VTAM, a method of communicating in one direction at a time. Either the application program or the logical unit assumes the exclusive right to send normal-flow requests, and the other node refrains from sending such requests. When the sender wants to receive, it releases the other node from its quiesced state.

R

record mode. In ACF/VTAM, a mode of data transfer in which the application program can communicate with logical units or with local non-SNA or remote 3270 Information Display Systems. Contrast with *basic mode*.

release. In ACF/VTAM resource control, to relinquish control of resources (communications controllers or physical units). See also *resource takeover*. Contrast with *acquire (2)*.

remote. In ACF/VTAM, pertaining to devices that are physically connected through a communications controller.

remote NCP. An NCP that is not attached directly through a channel, but is attached through a data link to a local NCP that is channel-attached. Contrast with *local NCP* and *peer NCP*.

reply. In SNA, a request unit sent in reaction to a previously received request unit (command). See also *command (2)*.

request. (1) A directive that causes a data transfer or related operation to be performed. Contrast with *response*. (2) In SNA, synonym for *request unit*.

request header. In SNA, a request/response header that indicates a request.

request parameter list (RPL). In ACF/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*. See also *request/response unit*.

resource takeover. In ACF/VTAM, the action of a network operator to transfer control of resources from one domain to another. See also *acquire (2)* and *release*.

responded output. In ACF/VTAM, a type of output request that is completed when a response is returned. Contrast with *scheduled output*.

response. (1) An answer to an inquiry. (2) The unit of information that is exchanged between ACF/VTAM or an ACF/VTAM application program and a SNA terminal to describe how a request arrived. (3) In SNA, synonym for *response unit*. (4) Contrast with *request*.

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*. See also *request/response unit*.

RH. Request/response header.

RPL. Request parameter list.

RPL-based macro instruction. In ACF/VTAM, a macro instruction whose parameters are specified by the user in a request parameter list.

RPL exit routine. In ACF/VTAM, a user-written routine whose address has been placed in the EXIT field of a request parameter list. ACF/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 ACF/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.

secondary application program. In a session, an application program that adheres to secondary session protocols. Contrast with *primary application program*.

secondary end of a session. A logical unit, secondary application program, or non-SNA terminal.

sequence number. A numerical identifier assigned by ACF/VTAM to each message exchanged between two nodes.

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 the user 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. (4) In the NCP, a line-scheduling period. See *LU-LU session*, *SSCP-LU session*, *SSCP-PU session*.

session control. In SNA, one of the components of transmission control. It is responsible for allocating resources necessary for a session, for purging data flowing in a session if an unrecoverable error occurs, and for resynchronizing the data flow after such an error.

session limit. (1) In the network control program, the maximum number of concurrent line-scheduling sessions on a non-SDLC, multipoint line. (2) In SNA, the maximum number of simultaneous sessions a particular network addressable unit can support.

session parameters. Synonym for *logon mode*.

share limit. The limit of the number of SSCPs that can simultaneously share a resource.

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 ACF/VTAM at the primary application program's request. The primary application program accepts or rejects the terminal as if it had logged on.

single-channel-attached communications controller. A communications controller that is channel-attached to only one host computer.

single-thread application program. An ACF/VTAM application program that processes requests from terminals one at a time. Such a program usually requests synchronous operations from ACF/VTAM, waiting until each operation is completed before proceeding. Contrast with *multithread application program*.

SNA. Systems network architecture.

SNA terminal. In ACF/VTAM: (1) A physical unit, logical unit, or secondary application program. (2) A terminal that is compatible with systems network architecture.

SNBU. Switched network backup.

solicit. In ACF/VTAM, to obtain data from a BSC or start-stop terminal or from a local non-SNA 3270 terminal and move the data into ACF/VTAM buffers.

solicited message. A response from ACF/VTAM to a network operator command entered by a program operator. Contrast with *unsolicited message*.

specific-mode. In ACF/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.

SSCP-LU session. A session during which ACF/VTAM (the system services control point, SSCP) and a logical unit (LU) can communicate.

SSCP-PU session. A session during which ACF/VTAM (the system services control point, SSCP) and a physical unit (PU) can communicate.

start options. In ACF/VTAM, the user-specified or IBM-supplied options that determine certain conditions that are to exist during the time an ACF/VTAM system is operating. For example: the size of ACF/VTAM buffer pools, which major and minor nodes are to be traced by the ACF/VTAM trace facility, and which major nodes are to be initially active. Start options can be predefined or specified by the network operator when ACF/VTAM is started.

subarea. A group of addressable elements in the network that have the same subarea ID.

subarea ID. A subfield of network address.

switched network backup (SNBU). An optional facility that allows a user to specify, for certain types of stations, a switched line to be used as an alternate path (backup) if the primary line becomes unavailable or unusable.

switched SNA major node. In ACF/VTAM, a major node whose minor nodes are physical and logical units attached by switched SDLC links.

synchronous operation. In ACF/VTAM, a connection, communication, or other operation in which ACF/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 ACF/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

teleprocessing subsystem. In ACF/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 ACF/VTAM, the secondary end of a session; that is, a logical unit, a start-stop or BSC device, a local non-SNA 3270 device, or an application program.

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 or a printer component of a keyboard-printer 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*.

U

unsolicited message. A network operator message, from ACF/VTAM to a program operator, that is unrelated to any command entered by the program operator. Contrast with *solicited message*.

Bibliography

This bibliography lists some of the non-ACF/VTAM IBM publications that are referred to in this publication or that relate to ACF/VTAM. (ACF/VTAM publications are shown in Figure P-1.) The listed publications (when available) contain a more detailed description of facilities that are used with ACF/VTAM.

General Data Communications

Introduction to Data Communications Systems, SR20-4461

Systems Network Architecture (SNA)

Systems Network Architecture General Information, GA27-3102

Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic, SC30-3112

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 ACF/VTAM, GC28-0663

OS/VS

OS/VS Dynamic Support System, GC28-0640

DOS/VS and OS/VS TOLTEP for ACF/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 ACF/VTAM Component Release Guide, GC27-0053

OS/VS2 SVS ACF/VTAM Component Release Logic, SY27-7279

OS/VS2 MVS

OS/VS2 System Programming Library:

Service Aids, GC28-0674

Supervisor, GC28-0628

Storage Estimates, GC28-0604

IBM 3704 and 3705 Communications Controllers

Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051

IBM 3705 Advanced Communications Function for Network Control Program/VS: Generation and Utilities Reference Manual, SC30-3116 (called the NCP Generation manual throughout this publication)

Index

- abnormal termination (ABEND) 129
- ACB
 - control block
 - description 89
 - relationship to executable macro instructions 92
 - use of 31
 - macro instruction 89
- acceptance
 - any mode 101
 - general description 100
 - specific mode 100
- accepting connection
 - description 31, 100
 - to a specific terminal 100
 - to any terminal 101
 - to terminals 100
 - with a LOGON exit routine 101
- accounting exit routine
 - coding and installing 61
 - defining 61
 - executed inline 62
 - input to 61
 - run in supervisor state 62
 - task 62
- ACF/VTAM
 - application programs, writing 87, 183
 - compared to BTAM 171
 - CPU support 143
 - definition, sharing resources through 158
 - functions, summary of 1
 - in operation 30
 - language 89
 - machine requirements 143
 - macro instructions 87
 - operating system requirements 145
 - operator commands
 - entering from a program operator 67
 - entering from the system console 67
 - summary 67
 - planning considerations 143
 - security 160
 - storage pools 63
 - system, controlling 67
 - traces
 - buffer traces 128
 - I/O traces 128
 - NCP line traces 128
 - storage management traces 128
 - using with non-SNA terminals 175
- acquiring connection
 - CONALL option 102
 - CONANY option 102
 - general description 31
 - security 163
 - SIMLOGON 53
 - using the OPNDST macro instruction 53, 102
- acquiring resources 82, 135
- acquiring terminals that are connected to another program 102
- activating and deactivating
 - cross-domain resource managers 80
 - cross-domain resources 80
 - effect on emulation mode 77
 - general description 74
 - local non-SNA terminals 76
 - local SNA terminals 77
 - non-shared local NCPs 77
 - path tables 80
 - PEP lines 77
 - remote attachments 78
 - remote NCPs 78
 - shared local NCPs 78
 - switched SNA major nodes 79
- activating major and minor nodes as START options 63
- activation, special considerations 81
- active
 - application program 81
 - terminal 81
- allocation of resources 11
- any mode 101, 121
- APPL statement 40
- application program
 - active 81
 - as a network operator 84
 - as a terminal 17
 - as an ACB 15
 - authorization
 - acquiring terminals 159
 - block processing 159
 - effect on sharing resources 158
 - passing connection 159
 - program operator 159
 - authorized facilities 40
 - defined to ACF/VTAM 15, 40
 - definition of 15
 - displaying status of 70
 - node structure 40
 - opening 100
 - organization 94
 - primary 87
 - secondary 87
 - starting and stopping 76
 - view of the network 28
 - writing 87, 183
- application program-initiated connection 52
- application program-initiated disconnection 58, 59
- asynchronous request handling 95
- AT&T 83B3 Selective Calling Station 191
- ATCSTRxx 63
- ATCSTR00 63
- audit trail 126, 129
- authorization exit routine
 - coding and installing 60
 - defining 60
 - executed inline with ACF/VTAM 61
 - run in supervisor state 61
 - security 163
 - task for 61
 - use in managing resources 159
- authorized facility, specified on APPL statements 40
- authorized path 100, 147
- automatic dialing for SNA terminals 154
- automatic logon
 - description 52
 - effect of disconnection on 60
- auxiliary storage used by ACF/VTAM 19
- backup
 - communications controller 140
 - manual switched CPU 140
 - switched network backup
 - considerations 150
 - general 134
- backup link 134
- basic mode
 - communication 94, 184
 - macro instructions
 - CHANGE 184

- description 89
- DO 184
- READ 184
- RESET 184
- SOLICIT 184
- WRITE 184
- restrictions in a multiple domain network 171
- special I/O operations 187
- special processing options 187
- Begin Bracket indicator 198
- Bid command 198
- Binary Synchronous Communication (BSC)
 - terminal supported 191, 192
- Bind command 9, 198
- block processing
 - authorizing use of 40
 - effect on resource sharing 186
- bracket indicators, summary 198
- bracket protocol
 - Begin Bracket indicator 119
 - Bid command 119
 - description 118
 - enforcement 119
 - Ready to Receive command 119
- BSC 3270 terminals, supported 191, 192
- BTAM
 - compared to ACF/VTAM 171
 - using with non-SNA terminals 171
- buffering 64, 133

- call-in terminals 181
- call-in/call-out terminals 182
- call-out terminals 182
- cancel closedown 70
- Cancel command 111, 198
- chain, message, definition of 111
- chaining 110
- Change Direction Command (CMD) indicator 117, 198
- change direction indicators, summary 198
- change direction protocol
 - description 117
 - enforcement 118
 - indicators, summary 198
- Change Direction Request (REQ) indicator 118, 198
- CHANGE macro instruction, description 184
- changing automatic logon 54
- channel adapter support 144
- character-coded (USS) definition table 49
- character-coded logoff 56
- character-coded logon
 - defining 47
 - description 46
 - IBM-defined 47
- Chase command 198
- CHECK macro instruction, description 90
- CID (*see* communication identifier)
- Clear command 198
- CLOSE macro instruction
 - description 88
 - use 92
 - using for disconnecting terminals 92
- closedown
 - cancel 70
 - orderly 69
 - quick 69
- closing an application program 91
- CLSDST macro instruction
 - description 88
 - PASS option 105
 - RELEASE option 105
- cluster minor node, defining 41
- cold restart 77

- commands
 - and indicators, summary 197
 - expedited-flow 198
 - network operator 67
 - normal-flow 198
 - communicating in basic or record mode 93
 - Communicating Magnetic Card SELECTRIC® Typewriter 191
 - communicating with non-SNA terminals 185
 - communication, common for basic and record mode 106
 - communication, record mode 121
 - communication identifier (CID) 93, 121
 - communication macro instructions
 - basic mode 184
 - READ 184
 - RESET 184
 - SOLICIT 184
 - WRITE 184
 - record mode 89
 - RECEIVE 89
 - RESETSR 89
 - SEND 89
 - SESSIONC 89
 - communication protocols
 - bracket protocol 119
 - change direction protocol 117
 - quiesce protocol 116
 - communications controller
 - as a remote station 193
 - backup SDLC link 144
 - features not supported 144
 - features supported 144
 - functions 16
 - general description 15
 - local and remote 143
 - models not supported 144
 - models supported 144
 - multiple channel attachment 144
 - requirements for 143
 - 3705 two-channel support 144
 - CONALL option 102
 - CONANY option 102
 - conditional disconnection 57, 105
 - confidential data, protecting 167
 - configuration restart
 - delayed 141
 - immediate
 - of an NCP 118
 - of physical units 118
 - connecting terminals 92
 - connection
 - and terminal sharing 160
 - controlling 163
 - description 100
 - primary 92, 100
 - secondary 93, 105
 - connection macro instructions
 - CLOSE 88
 - CLSDST 88
 - OPNDST 88
 - OPNSEC 88
 - REQSESS 88
 - SESSIONC 88
 - SIMLOGON 88
 - TERMSESS 88
 - considerations in moving from VTAM Level 2 to ACF/VTAM
 - ACF/VTAM-ACF/TCAM considerations 202
 - configuration considerations 199
 - definition considerations 200
 - operator considerations 201
 - continue-any mode 123
 - continue-specific mode 123

- control block macro instructions
 - declarative 89
 - manipulative 90
- control blocks
 - ACB 89
 - EXLST 89
 - handling 94
 - NIB 89
 - relating to executable macro instructions 91
 - RPL 89
- controlling
 - access to confidential data 167
 - access to ACF/VTAM 166
 - an ACF/VTAM system 67
 - buffering 64
 - connections 164
 - non-SNA terminals 183
 - the ACF/VTAM system 29
 - use of ACF/VTAM facilities 167
- conversion (*see* considerations in moving from VTAM Level 2 to ACF/VTAM)
- coordinating I/O 118
- counter overflow 127
- CPT-TWX Terminal 191
- CPU support 143
- creating an ACF/VTAM system
 - general description 29
 - multiple-domain 33
 - single-domain 33
- cross-domain functions of the Multisystem Networking Facility 1
- cross-domain link 9

- data flow control 107
- data flow through an ACF/VTAM system 11
- data sets
 - for ACF/VTAM under DOS/VS 155
 - for ACF/VTAM under OS/VS 155
- DATAMGT macro instruction 33
- deactivation
 - forced 76
 - immediate 75
 - normal 75
 - restart 76
- declarative macro instructions 89
- defining
 - a character-coded (USS) definition table 49
 - ACF/VTAM buffering 64
 - ACF/VTAM initiated connection 52
 - ACF/VTAM start options 62
 - ACF/VTAM to the operating system 33
 - an authorization exit routine 60
 - application program major nodes 40
 - application program-initiated connection 52
 - automatic logons 52
 - BSC and start-stop terminals 42
 - BSC clusters 41
 - character-coded connection 47
 - connection procedures 44
 - control blocks, different methods 94
 - cross-domain path tables 44
 - cross-domain resource managers 43
 - cross-domain resources 44
 - disconnection procedures 56
 - field-formatted connection 46
 - groups of lines 41
 - interpret tables 178
 - lines 41
 - local devices to the operating system 33
 - local non-SNA major nodes 41
 - local SNA major nodes 41
 - logical units 41
 - logon modes 54
 - logon-interpret exit routines 179
 - logons and logoffs 46
 - major nodes 36
 - minor nodes 37
 - NCP major nodes 41
 - network operator-initiated connection 53
 - other domains to ACF/VTAM 43
 - physical units 41
 - ports 41
 - SNA terminals 41
 - start options, filing 62
 - start options, predefined 63
 - switched BSC and start-stop terminals 181
 - switched line group 150
 - switched SNA major nodes 43, 150
 - terminal components 42
 - terminal-initiated connection 46
 - terminal-initiated logons for non-SNA terminals 180
 - terminals 42
 - the network to ACF/VTAM 36
- definite response
 - requesting 107
 - type 1 110
 - type 2 110
- definition macro instruction 42
- definition statement 42
- delayed configuration restart 141
- delayed control mode 120
- delayed request mode 120
- delayed response mode 120
- designing ACF/VTAM application programs 30
- DFASY exit routine 114
- dial-in (*see also* call-in) 151
- dial-out (*see also* call-out) 152
- DISCNT parameter 57
- disconnecting terminals 93
- disconnection
 - conditional 57
 - description
 - primary 93, 105
 - secondary 93, 105
 - effect on automatic logon 60
 - general description 56
 - of logical units 93
 - of switched SNA terminals 154
 - pass 105
 - release 105
 - requested by a secondary application program 59
 - requested by an application program 58
 - requested by the network operator 59
 - requested by the terminal 56
 - requested by ACF/VTAM 59
 - unconditional 57
- DISPLAY command 70
- displaying
 - ACF/VTAM buffer pools 73
 - application programs 71
 - buffer pools 73
 - communication lines 71
 - cross-domain resource managers 73
 - cross-domain resources 72
 - lines 73
 - major nodes 73
 - NCPs 72
 - path tables 73
 - pending session I/O 74
 - physical units 72
 - ports 72
 - sets of cross-domain resources 73
 - terminals 71, 72
- distributed function 13
- DO macro instruction, description 184

- domain
 - definition 1
 - multiple-domain system 28
 - single-domain system 19
 - SNA concept as used by ACF/VTAM 9
- DOS/VS
 - CPU support 143
 - other telecommunication access methods in requirements 168
- DSS (*see* Dynamic Support System)
- dump utility for the NCP 130
- dumps
 - effect on emulation mode 154
 - NCP 126, 130
 - operating system
 - DOS/VS 126
 - OS/VS 126, 129
 - TOLTEP 126
- duplicate names 39
- Dynamic Support System (DSS) 147

- ECB compared to RPL exit routine 95
- elements of an ACF/VTAM system 13
- emulation mode
 - effect of activation and deactivation not supported by ACF/VTAM 15
- End Bracket indicator 198
- end-of-day records 127
- enforcement
 - of bracket protocol 119
 - of change direction protocol 118
 - of quiesce protocol 116
 - of session parameters 54
- error
 - detection and feedback 132
 - notification 98
 - recording 126
 - recovery 133
 - return codes 98
- establishing procedures for using the system 30
- establishing sets of session parameters 104
- event control block (*see* ECB)
- exception message 112
- exception response, requesting 107
- excess data, handling 123
- EXECPPL macro instruction, description 90
- EXLST
 - control block
 - description 89
 - specified by NIB 92
 - exit routines
 - accounting 61
 - ATTN 96
 - authorization 60
 - DFASY 96
 - LERAD 96
 - LOGON 96
 - logon-interpret 62
 - LOSTERM 96
 - NSEXIT 96
 - RELREQ 96
 - RESP 96
 - SCIP 96
 - SYNAD 96
 - TPEND 96
 - macro instruction, description 89
- expansion number 64
- expansion point 64
- expedited-flow commands 198
- expedited message flow 107

- field-formatted logoff 57
- field-formatted logon 46
- filing ACF/VTAM definition statements 36
- filing NCP definition statements 42
- fixed storage pools 63
- formatted system services (FSS) 46

- GENCB macro instruction, description 90
- generating ACF/VTAM 33
- generating an NCP 36

- HALT command 69
- halting ACF/VTAM
 - cancel 70
 - general description 69
 - orderly 69
 - quick 69
- handling excess data 123
- hardware error recording 127
- hardware error recovery procedures 133
- holding a physical unit connection 57

- I/O areas handling 94
- ID verification (*see* identification verification)
- identification verification
 - for BSC and TWX terminals 161
 - for physical units 161
 - for switched SNA terminals 161
 - for the host computer 161
- identifying terminals 121
- immediate configuration restart
 - of an NCP 140
 - of physical units 140
- immediate control mode 120
- immediate deactivation 59
- immediate request mode 120
- immediate response mode 120
- initial status of nodes 159
- Initiate command 8, 46
- initiating requests for connection 82
- input, receiving 112
- INQUIRE macro instruction, description 91
- installing an ACF/VTAM system 29
- interpret tables, specifying 178
- INTRPRET macro instruction, description 91

- KEEP option 123

- LDO control block 184
- LERAD exit routines 99
- levels of network control 67
- line scheduling specifications, changing 83
- line sharing for start-stop and BSC terminals 160
- loading when activating an NCP 77
- local communications controller
 - transferring as a local communications controller 135
 - transferring as a remote communications controller 138
- local non-SNA terminal, defining 41
- local SNA terminal, defining 41
- local 3270 device support 143
- local 3790 device support 143
- logical unit (*see also* terminal)
 - connection 44
 - definition 6, 16
 - local, defining to ACF/VTAM 41
 - nonswitched, defining to ACF/VTAM 41
 - switched, defining to ACF/VTAM 43

- Logical Unit Status (LUS) command 198
- logoff 56
- logon (*see also* connection)
 - general description 8
 - preventing 101
 - types of 31, 46
- LOGON exit routine 8
- logon-interpret routine 62, 179
- logon mode, definition of 46
- logon mode name 54
- logon mode table 42
- logons, controlling 164
- LOSTERM exit routine 96
- LU-LU session
 - completing 9
 - opening 8

- machine requirements 143
- macro instructions
 - basic mode 89
 - communication 88
 - connection 88
 - control block 89
 - manipulative 90
 - network control 89
 - record mode 88
 - relating to control blocks 91
 - summary 88
 - ACB 89
 - CHECK 90
 - CLOSE 88
 - CLSDST 88
 - DO 184
 - EXECRPL 90
 - EXLST 89
 - GENCB 90
 - INQUIRE 91
 - INTRPRET 91
 - MODCB 90
 - NIB 89
 - OPEN 88
 - OPNDST 8, 88
 - OPNSEC 88
 - RCVCM 89
 - READ 184
 - RECEIVE 89
 - REQSESS 88
 - RESET 184
 - RESETSR 89
 - RPL 90
 - SEND 89
 - SENDCMD 89
 - SESSIONC 88, 89
 - SETLOGON 91
 - SHOWCB 90
 - SIMLOGON 88
 - SOLICIT 184
 - TERMESS 88
 - TESTCB 90
 - WRITE 184
 - types of 87
- major and minor node structure 37
- major node 36
- managing resources
 - through ACF/VTAM definition 158
 - through application programs 160
 - through NCP generation 159
- manipulative macro instructions 90
- manual switched CPU and communication controller
 - backup 140
- message
 - definition of 106
 - exception 95, 112
- message chain, definition of 111
- message control information, summary 197
- message flow
 - expedited 107
 - normal 107
- minor node 37
- MODCB macro instruction, description 90
- monitoring
 - ACF/VTAM status 70
 - application programs 71
 - buffer pools 73
 - communication lines 71
 - cross-domain resources 72
 - NCPs 72
 - path tables 73
 - pending session I/O 74
 - physical units 72
 - ports 72
 - sets of cross-domain resource managers 72
 - sets of cross-domain resources 72
 - terminals 71
- multiple console support 147
- multiple-domain system 11, 28
- Multisystem Networking Facility, functions, summary of 1

- names
 - duplicate 39
 - reserved 40
- naming nodes 40
- NCP
 - defining to ACF/VTAM 148
 - functions required by ACF/VTAM 148
 - general description 15
 - initial test 132
 - requirements 148
 - slowdown 134
 - steps in generating 42
- negative response 110
- network addressable unit, definition of 4
- network control macro instructions 89
- network control mode 15
- network control program (*see* NCP)
- network operator
 - an application program as 84
 - considerations for control 86
 - general description 67
 - in a multiple-domain network 67
- network solicitor
 - how it works 176
 - messages 178
 - modifying 177
 - naming 177
 - password 178
 - release request 178
 - replacing 178
- NIB control block, use in connection 92
- NIB macro instruction, description 89
- node structure major and minor 37
- non-SNA terminal
 - ACF/VTAM system that includes 175
 - communicating with 185
 - defining terminal-initiated logons for 180
 - definition of 17
 - support for 171
 - topics not applicable to 175
- normal deactivation 58
- normal flow commands 198
- normal message flow 107

OPEN macro instruction
 description 88
 use 91
 opening an application program 91, 100
 operating system requirements 145
 operating system traces 128
 OPNDST macro instruction
 description 88
 use 8, 93
 OPNSEC macro instruction
 description 88
 use 105
 orderly closedown 69
 organization of an application program 94
 OS/VS
 CPU support 143
 other telecommunication access methods 169
 requirements 147
 OS/VS logon, security considerations 166
 other telecommunication access methods
 in a multiple-domain network 170
 under DOS/VS 168
 under OS/VS 169
 output, responded 111
 output, scheduled 111

pacing 149
 pageable storage pools 63
 partitioned emulation programming extension (PEP)
 general description 15
 planning considerations 154
 passing connection
 general description 53
 security 163
 PEP (*see* partitioned emulation programming extension)
 permanent hardware errors 127
 physical unit 16
 positive response 110
 primary application program
 connection 92, 100
 definition of 6
 disconnection 93, 105
 procedures, establishing, for using the system 30
 program operator 84
 protecting confidential data 167
 protocols
 additional 120
 bracket 118
 change-direction 116
 quiesce 116

queuing connection requests
 effect on availability 104
 general description 102
 quick closedown 69
 Quiesce at End of Chain (QEC) command 116, 198
 Quiesce Complete (QC) command 116, 198
 quiesce protocol 116

RAS (*see* reliability, availability, serviceability)
 RVCMD macro instruction, description 89
 READ macro instruction, description 184
 Ready to Receive (RTR) command 119, 198
 RECEIVE macro instruction, description 89
 receiving input 112
 record mode communication 93, 106
 recording hardware errors 127
 recording software errors 127
 reenterability, requirements for EXLST exit routines 99

Release Quiesce (RELO) command 116, 198
 releasing resources 82
 reliability and availability support, summary 131
 reliability, availability, serviceability 125
 RELREQ exit routine 102
 remote communications controller, transferring as a remote
 communications controller 141
 remote devices not defined to the operating system 36
 remote station versus remote controller 193
 remote terminals 145
 REQSESS macro instruction
 description 88
 use 105
 request mode selection 120
 Request Recovery (RQR) command 198
 Request Shutdown (RSHUTD) command 198
 request shutdown protocol 58
 requesting a response 107
 requesting another application program's terminal 102
 reserved names 40
 RESET macro instruction, description 184
 RESETSR macro instruction, description 89
 resource takeover 134
 resources
 allocation 1
 sharing 11
 RESP exit routine 114
 responded output 111
 response
 definition of 106
 negative 110
 positive 110
 requesting
 definite response 107
 exception response 107
 restart
 cold 77
 warm 77
 RPL control block use in connection 92
 RPL macro instruction, description 90

scheduled output 111
 SCIP exit routine 114
 secondary application program
 connection 93, 105
 definition of 6
 disconnection 93, 106
 security 160
 SEND macro instruction, description 89
 SENDCMD macro instruction, description 89
 sending messages
 responded 111
 scheduled 111
 sequence numbers 111
 sequencing 110
 serviceability aids
 error recording 126
 summary 126
 session
 definition of 6, 31
 LU-LU 9
 SSCP-LU 8
 SSCP-PU 7
 session control 107
 session parameters
 establishing
 for local and BSC 3270 104
 for non-SNA terminals 105
 general description 31, 54
 SESSIONC commands 198

SESSIONC macro instruction
description 88, 89, 107
use in connection 105

Set and Test Sequence Numbers (STSN) command 198

SETLOGON macro instruction, description 91, 101

sharing resources
general description 11
managing 158
resources that can be shared 157
through logon 158

SHOWCB macro instruction, description 90

Shutdown (SHUTD) command 198

Shutdown Complete (SHUTC) command 198

Signal command 198

SIMLOGON macro instruction, description 88

simulating a logon 53

single-domain system 19

slowdown point 64

SNA concepts in ACF/VTAM
network addressable unit 4
primary logical unit 6
secondary logical unit 6
session (*see* session)

SNA terminals supported 189

software error recording 127

software errors, processing 134

SOLICIT macro instruction, description 184

solicitation
block 186
continuous 187
message 186
transmission 186

special handling of a response 110

specific mode 101, 121

SSCP-LU session 8

SSCP-PU session 7

Start Data Traffic (SDT) command 198

start options, ACF/VTAM 62, 68

start-stop terminals supported 191

starting and stopping ACF/VTAM facilities 83

starting ACF/VTAM
in DOS/VS 69
in OS/VS 69

storage management 132

storage pools
fixed 63
pageable 63

storage requirements 145

subarea
adjacent subarea 44
destination subarea 44
number 37

supervisor state
accounting exit routine in 62
authorization exit routine in 61

support macro instructions 90

suppressing network operator messages 83

SUPVR macro instruction 33

switched line group, generating 149

switched network 149

switched network backup
considerations 150
general 134

symbolic names use in controlling connection 166

SYNAD exit routines 99

synchronous request handling 95

system services control point (SSCP) 4, 46

System/3 192

System/32 (BSC) 192

System/32 (SNA) 191

System/370, as a supported terminal 192

System/7 (BSC) 192

System/7 (start-stop) 191

takeover
of a local communications controller as a local communications controller 135
of a local communications controller as a remote communications controller 138

task for accounting exit routine 62

task for authorization exit routine 61

teleprocessing online test executive program (TOLTEP) 130

terminal
active 81
application program as 17
connecting 92
cross-domain communication ability 16
disconnecting 93
non-SNA
general 17
support for 171
sharing 160
SNA 16
supported 189

terminal-initiated connection for SNA terminals 46

terminal-initiated connection types 46

TERMSESS macro instruction
description 88
use 105

TESTCB macro instruction, description 90

TOLTEP (*see* teleprocessing online test executive program)

traces
ACF/VTAM
buffer 128
general 126
I/O 128
NCP line 128
storage management 128
DOS/VS 126, 128
OS/VS 126, 128

transfer of data 1, 32

transferring a local communications controller
as a local communications controller 135
as a remote communications controller 138

transferring a remote communications controller as remote 141

TRUNC option 123

unconditional disconnection 57, 106

unformatted system services (USS) 46

upward compatibility 145

user logon message 46

using an interpret table with non-SNA devices 52

views of an ACF/VTAM system
multiple-domain
as viewed by ACF/VTAM 29
as viewed by application programs 29
as viewed by the operating system 28
physical configuration 28

single-domain
as viewed by ACF/VTAM 28
as viewed by an application program 28
as viewed by the operating system 19
physical configuration 19

VTAM Level 2, considerations in moving from, to ACF/VTAM 199

warm restart 77

work areas, handling 94

World Trade Telegraph 191

WRITE macro instruction, description 184

Western Union Plan 115A 191

1050 Data Communication System 191
2740 Communication Terminal 191
2741 Communication Terminal 191
2770 Data Communication System 192
2780 Data Transmission Terminal 192
2980 General Banking Terminal System 192
3270 Information Display System
 (BSC) 191
 (local) 191
 (SNA) 191
3600 Finance Communication System 191
3650 Retail Store System 191
3660 Supermarket System 191
3735 Programmable Buffered Terminal 192
3740 Data Entry System 192
3750 Switching System 192

3767 Communication Terminal
 (SNA) 191
 (start-stop) 191
3770 Data Communication System
 (BSC) 192
 (SNA) 191
3780 Data Communications Terminal 192
3790 Communication System
 (local) 191
 (remote) 191
5100 Portable Computer 191
5275 Direct Numerical Control Station 192
5937 Industrial Terminal
 (BSC) 192
 (local) 191

Order No. GC38-0282-1

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?

- | | |
|--|---|
| <input type="checkbox"/> As an introduction | <input type="checkbox"/> As a text (student) |
| <input type="checkbox"/> As a reference manual | <input type="checkbox"/> As a text (instructor) |
| <input type="checkbox"/> For another purpose (explain) _____ | |
-

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



BUSINESS REPLY MAIL
 FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

NO POSTAGE
 NECESSARY
 IF MAILED
 IN THE
 UNITED STATES



POSTAGE WILL BE PAID BY ADDRESSEE:

Postage will be paid by:

**International Business Machines Corporation
 Department 63T
 Neighborhood Road
 Kingston, New York 12401**

Fold

Fold



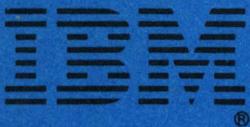
**International Business Machines Corporation
 Data Processing Division
 1133 Westchester Avenue, White Plains, N.Y. 10604**

**IBM World Trade Americas/Far East Corporation
 Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591**

**IBM World Trade Europe/Middle East/Africa Corporation
 360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601**

ACF/VTAM Concepts and Planning (File No. S370-30) Printed in U.S.A. GC38-0282-1

ACF/VTAM Concepts and Planning (File No. S370-30) Printed in U.S.A. GC38-0282-1



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601