SC38-0258-1
File No. S370-30

Advanced Communications Function for VTAM (ACF/VTAM)

Program Product

System Programmer's Guide

Program Number: 5735-RC2 (OS/VS)

IBM

Second Edition (May 1978)

This edition is a major revision of SC38-0258-0 and makes that publication and its associated Technical Newsletter SN31-0692 obsolete. The changes to this edition are described in the Summary of Amendments.

This edition applies 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. Information about the optional ACF/VTAM Encrypt/Decrypt Feature, available for OS/VS1 and OS/VS2 only, is also included.

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

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 (May 31, 1978) to SC38-0258-0 by Revision SC38-0258-1

New Documentation

This revision adds storage estimate worksheets for OS/VS2 SVS.

This revision adds Appendix D which contains information concerning TSO/VTAM.

This revision adds Appendix E which lists the ACF/VTAM storage pools and the ACF/VTAM control blocks that are allocated from them.

Changed Documentation

This revision updates storage estimate worksheets for OS/VS1 and OS/VS2 MVS.

This revision includes various technical and editorial changes.

Summary of Amendments (February 28, 1978) to SC38-0258-0 by TNL SN31-0692

ACF/VTAM Encrypt/Decrypt Feature

New Program Feature

Information about the ACF/VTAM Encrypt/Decrypt Feature is included (OS/VS1 and OS/VS2 MVS only).

ACF/VTAM

New Program Function

This TNL provides a description of a new method for updating SYS1.VTAMOBJ from changes in SYS1.VTAMLST.

New Documentation

This TNL adds a description of the backup and reconfiguration facilities available in a multiprocessor.

This TNL adds storage estimate worksheets and buffer pool information for OS/VS1 to pages C-9—C-33 in Appendix C.

Changed Documentation

This TNL includes various technical corrections and editorial changes.

Preface

This publication describes the ACF/VTAM definition statements used to install, tailor, tune, and maintain the Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM) in a data communication system. It is intended for system programmers, IBM programming service representatives, and others involved in installing or maintaining ACF/VTAM in conjunction with the OS/VS system control program. (In this book, the term 'OS/VS' refers collectively to the OS/VS1, OS/VS2 MVS, and OS/VS2 SVS systems.)

Before reading this book, the reader should be familiar with:

The overall concepts of the OS/VS system control program

The basic data communication concepts

The overall concepts of ACF/VTAM

The network configuration should have been planned, and the major ACF/VTAM options chosen, for the data communication system. (This information is provided in the prerequisite publication for this book, *Advanced Communications Function for VTAM (ACF/VTAM) Concepts and Planning,* GC38-0282.)

ACF/VTAM can operate with either 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 used unless ACF/NCP/VS is installed. *Advanced Communications Function for VTAM (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 Controller,* GA27-3051.

This book does not explain how to define specific IBM subsystems (such as the IBM 3600 Finance Communication System) or IBM data base/data communication program products. Separate manuals describe how to install, tailor, and maintain those subsystems and programs that operate with ACF/VTAM.

Organization

This book is organized by user tasks, rather than by macro instructions, to reduce the need for extensive cross-referencing. Although the information in one chapter is sometimes related to information in another, the chapters have been written as separate and complete units. Each chapter contains its own introductory information (immediately following the chapter title) and describes a specific aspect of installing tailoring, tuning, or maintaining ACF/VTAM. Special helpful information is provided in several appendixes at the back of this publication. The glossary contains ACF/VTAM terms and acronyms that are used in this publication.

Contents

This publication contains the following chapters and appendixes:

Chapter 1, "Introduction," provides a summary of ACF/VTAM and an overview about how this book is organized.

Chapter 2, "Defining the Network," describes how to define application program major nodes, local non-SNA major nodes, local SNA major nodes, switched SNA major nodes, NCP major nodes, and (with the Multisystem Networking Facility only) CDRM major nodes, CDRSC major nodes, and path tables. It also describes the conventions used in this book to explain the syntax of the statements, macro instructions, and commands and states the rules for coding them.

Chapter 3, "Starting and Controlling the Network," describes how to define the ACF/VTAM start options, how to create configuration and start option lists, and how to put these start option and configuration lists into the ACF/VTAM definition library.

Chapter 4, "Defining Connection and Disconnection Procedures," describes how to create logon mode tables and USS definition tables, and describes the IBM-supplied logon mode and USS definition tables. It also describes how to define automatic and OS/VS logon, how to define and install interpret tables, and how to use, modify, and install network solicitors.

Chapter 5, "ACF/VTAM Services," describes how to code and install authorization and accounting exit routines.

Chapter 6, "Reliability, Availability, and Serviceability Facilities," describes ACF/VTAM's configuration restart facility, and briefly discusses its error recording and recovery features, the ACF/VTAM trace facility, the NCP dump facility and the Teleprocessing Online Test Executive Program (TOLTEP).

Chapter 7, "Tuning ACF/VTAM," provides information that is useful when tuning ACF/VTAM.

Appendix A is a collection of device dependencies. This appendix should be referred to before or while coding the network control program macro instructions.

Appendix B explains the messages issued to a terminal user by ACF/VTAM. The appendix also provides background programming information that can help establish the reason for the message.

Appendix C provides work sheets for estimating buffer pool sizes and storage requirements.

Appendix D provides information concerning TSO/VTAM (OS/VS2 MVS only).

Appendix E provides information concerning ACF/VTAM buffer pool control blocks.

A glossary of ACF/VTAM terms and acronyms and an index for the entire book follow the appendixes.

Prerequisite Publication

This book requires an understanding of the information in *Advanced Communications Function for VTAM ACF/VTAM Concepts and Planning,* GC38-0282, which describes overall ACF/VTAM concepts and provides the programming considerations for planning an ACF/VTAM system.

Corequisite Publication

This book should be used in conjunction with *Advanced Communications Function for VTAM (ACF/VTAM) Installation Guide* for OS/VS, SC38-0269, which provides additional information on installing ACF/VTAM in an OS/VS system and examples of how to use the definition statements descıibed in this book.

Associated Publications

The publications listed below are frequently referred to throughout this book and should be used in conjunction with this publication. Many of these titles are referred to in an abbreviated form, in which the 'Advanced Communications Function for VTAM' part of the title has been omitted. For example, *Advanced Communications Function for VTAM (ACF/VTAM) Concepts and Planning* is referred to as *ACF/VTAM Concepts and Planning.* Note that for some publications, different order numbers are used for each of the OS/VS systems. When ordering these books, ensure that you use the order number that is appropriate for your system.

ACF/VTAM

Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Reference, SC38-0261

Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Guide, SC38-0256

Advanced Communications Function for VTAM (ACF/VTAM) Program Operator Guide, GC38-0257

Advanced Communications Function for VTAM (ACF/VTAM) TOLTEP, SC38-0283

Advanced Communications Function for VTAM (ACF/VTAM) Network Operating Procedures, SC38-0259

Advanced Communications Function for VTAM (ACF/VTAM) Messages and Codes, SC38-0271

Advanced Communications Function for VTAM (ACF/VTAM) Reference Summary, SX27-3021

Advanced Communications Function for VTAM (ACF/VTAM) Debugging Guide, SY27-8006

NCP

Throughout this book, the reader is directed to the *NCP Generation Manual* for information about the network control program. *NCP Generation Manual* 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

Operating System

OS/VS Access Method Services, GC26-3836

OS/VS Linkage Editor and Loader, GC26-3813

OS/VS Utilities, GC35-0005

For OS/VS1 users, these additional books:

OS/VS1 Planning and Use Guide, GC24-5090

OS/VS1 Service Aids, GC28-0665

OS/VS1 Storage Estimates, GC24-5094

For OS/VS2 users, these additional books:

OS/VS2 System Programming Library: Service Aids, GC28-0674

OS/VS2 System Programming Library: Storage Estimates, GC28-0604

OS/VS2 System Programming Library: Supervisor, GC28-0628

Operator's Library: OS/VS2 Reference (JES2), GC38-0210

OS/VS2 JCL, GC28-0692

OS/VS2 System Programming Library: Initialization and Tuning Guide, GC28-0681

OS/VS2 System Programming Library: OLTEP, GC28-0675.

OS/VS2 System Programming Library: Service Aids, GC28-0674.

OS/VS2 System Programming Library: System Generation Reference, GC26-3792

Installing the IBM 3790 Communications System for Use with OS/VS2, GC22-9022

TSO/VTAM Publications

For a list of TSO and ACF/VTAM related publications, see Appendix D.

Related Publications

In addition, the following manuals can be used with *ACF/VTAM Concepts and Planning* and Appendix A of this publication to determine device considerations:

Introduction to Programming the IBM 3270, GC27-6999

IBM 3600 Finance Communication System Programming Installation Guide, GC27-0009

IBM 3600 Finance Communication System Programmer's Guide and Component Description, GC27-0004

IBM 3650 Retail Store System Subsystem Definition and Programmer's Guide, GC30-3023

SPPS Programmer's Guide, GC30-3024

IBM 3660 Supermarket System Subsystem Definition and Programmer's Guide, GC30-3025

IBM 3770 Communication System Programmer's Guide, GC30-3028

IBM 3790 Communication System Host System Programmer's Guide, GC27-0026

IBM 3790 Programming Statements Guide, GC27-0015

IBM Host Services Guide, GC27-0017

Contents

Figures

Chapter 1. Introduction

The IBM Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM) is an option of OS/VS that provides basic modules for constructing a teleprocessing program. The program directs the transmission of data between application programs and local or remote terminals, and controls the terminals in a data communication network.

ACF/VTAM operates with IBM 3704 or 3705 Communications Controllers. Each communications controller contains a resident program, the network control program/virtual storage. ACF/VTAM can operate with either 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 used unless ACF/NCP/VS is installed. *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 Controller* GA27-3051.

The NCP can be generated to operate the communications controller:

Only in network control mode

Only in emulation mode

On a line basis, in either network control mode or emulation mode using partitioned emulation programming (PEP)

ACF/VTAM supports only the network control mode, either with the NCP alone or with PEP. Statements about the NCP in this publication refer only to the network control mode.

An NCP that was specified and generated to be part of a network controlled by VTAM Level 2 must be respecified and regenerated when the NCP is to operate in an ACF/VTAM-controlled network. For information on how an NCP generated for VTAM Level 2 must be changed if it is to operate with ACF/VTAM, refer to *ACF/VTAM Concepts and Planning.*

An ACF/VTAM system programmer is responsible for:

Ensuring that the operating system contains the facilities and resources needed by ACF/VTAM

Coding and generating one or more network control programs that meet the needs of the network

Coding and filing the statements and macro instructions that define the network configuration, program resources, and desired facilities

Tailoring the ACF/VTAM facilities to the requirements of a network

Including ACF/VTAM service aids in the defined system and understanding their capabilities

Updating the ACF/VTAM system definition to match changes in the configuration, physical devices, and application programs executed in the host computer

Components Of An ACF/VTAM Network

The elements of an ACF/VTAM network are organized into groups of major and minor nodes. Major and minor nodes are the controllable elements of the network. A major node is a set of controllable elements (minor nodes) in the network. Each major node can be controlled as a whole, or portions of it can be controlled through the minor nodes within each major node.

Figure 1-1 shows the types of major nodes that can form an ACF/VTAM network.

Application program major nodes consist of one or more user-written application programs that are executed in the host processor and use ACF/VTAM macro instructions to communicate with terminals. (In ACF/VTAM publications, logical units, application programs acting as logical units, and non-SNA terminals are called terminals unless a distinction is necessary.) The minor nodes of an application program major node are the individual application programs. The major node can consist of just one minor node, or it can consist of several.

Local non-SNA major nodes consist of one or more non-SNA terminals attached by channel to the host computer. The minor nodes are individual non-SNA terminals. For a list of the non-SNA terminals supported by ACF/VTAM, see *ACF/VTAM Concepts and Planning.*

Local SNA major nodes consist of one or more SNA physical units attached by channel to the host computer. The minor nodes of a local SNA major node are the physical units and their associated logical units.

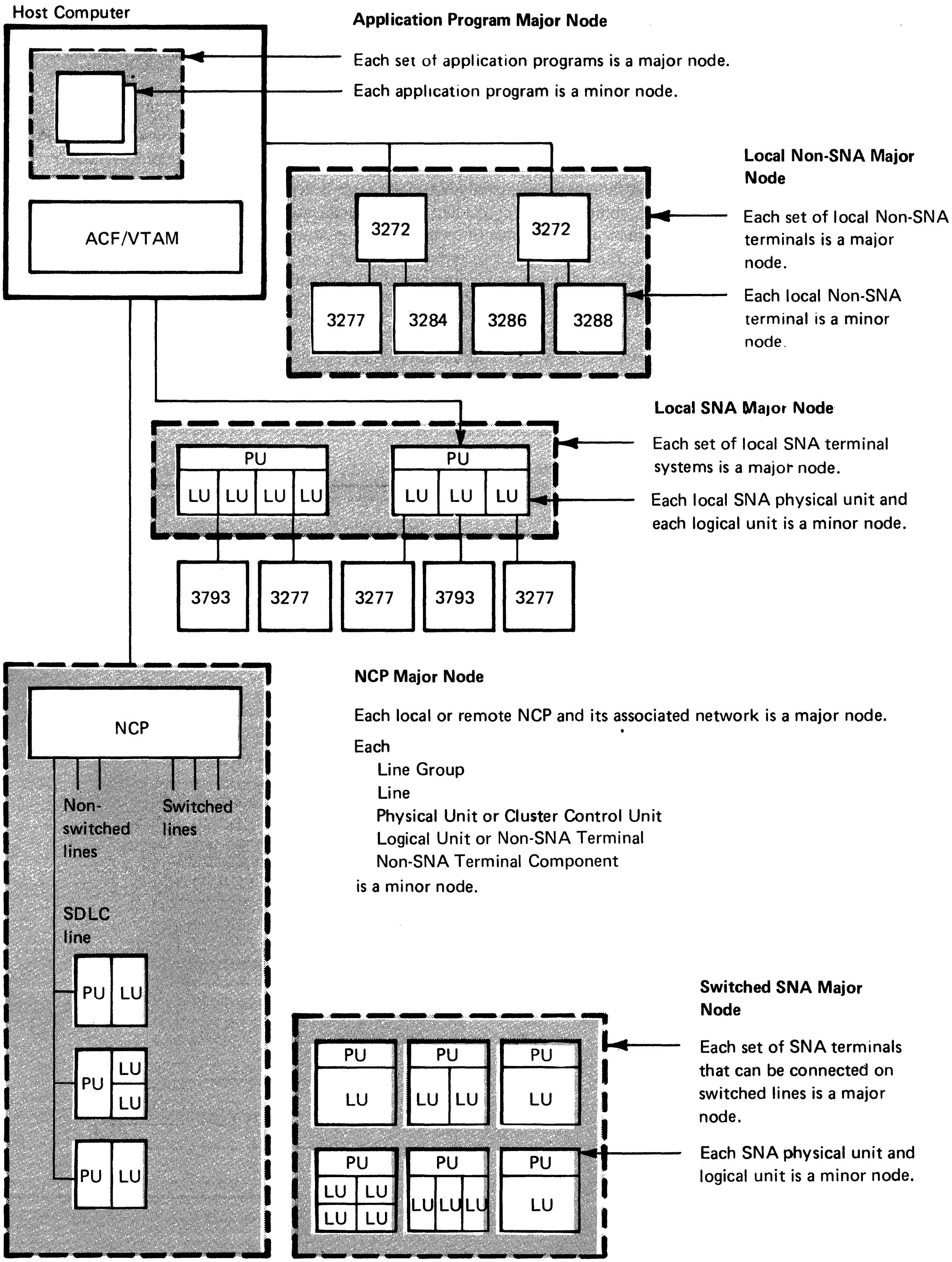
NCP major nodes consist of a communications controller, a network control program (NCP) being executed in that controller, and the physical configuration defined for that NCP during NCP generation. This physical configuration can include start-stop, BSC, and SDLC links (either switched or leased) and the terminals attached to them.

The communications controller can be either a local communications controller or a remote communications controller. A local communications controller is a communications controller that is attached locally (by a channel) to the host computer. A remote communications controller is a communications controller that is connected to a local communications controller by a synchronous data link control (SDLC) communication line. The communications controller acts as a remote concentrator. The remote terminals and the remote communications controller communicate with ACF/VTAM through the network control program that is executed in the local communications controller to which they are attached. Terminals connected to a remote communications controller communicate with ACF/VTAM through (1) the NCP that is executed in the remote communications controller and (2) the local NCP.

The minor nodes of an NCP are the terminals and lines that form the physical configuration.

Switched SNA major nodes consist of one or more of the SNA terminals supported by ACF/VTAM. For a list of the SNA terminals supported by ACF/VTAM, see *ACF/VTAM Concepts and Planning.*

Switched SNA major nodes do not include the switched SDLC links to which the terminals are attached (these lines are part of some NCP major node). For dial-out operations, however, the user can define alternate routes, or paths, over which contact with the terminal can be established by one or more NCPs. These switched path definitions, which associate the terminal with one or more switched lines, are part of the switched SNA major node. The minor nodes of a switched SNA major node are the SNA controllers and their associated logical units.



Application Program Major Node

Each set of application programs is a major node.

Each application program is a minor node.

Local Non-SNA Major Node

Each set of local Non-SNA terminals is a major node.

Each local Non-SNA terminal is a minor node.

Local SNA Major Node

Each set of local SNA terminal systems is a major node.

Each local SNA physical unit and each logical unit is a minor node.

NCP Major Node

Each local or remote NCP and its associated network is a major node.

Each
  Line Group
  Line
  Physical Unit or Cluster Control Unit
  Logical Unit or Non-SNA Terminal
  Non-SNA Terminal Component
is a minor node.

Switched SNA Major Node

Each set of SNA terminals that can be connected on switched lines is a major node.

Each SNA physical unit and logical unit is a minor node.

Figure 1-1. Major and Minor Nodes in a Single-Domain ACF/VTAM System

A specific network can differ from Figure 1-1 in the following significant ways:

More than one local communications controller, each with its own terminal configuration, might be attached to host processor channels.

More than one host processor might be attached to a local communications controller.

An NCP can be generated to include partitioned emulation programming (PEP). (ACF/VTAM supports PEP only for lines that are in network control mode.)

Although the switched SNA major node is depicted as being associated with the local communications controller, it can be associated with any number of switched SDLC links of any number of communications controllers. For example, a particular terminal can be associated with several of the local communications controller's lines and with several of the remote communications controller's lines.

The system need not have both local terminals and remote terminals.

Figure 1-2 shows two additional types of major and minor nodes in a multidomain system.

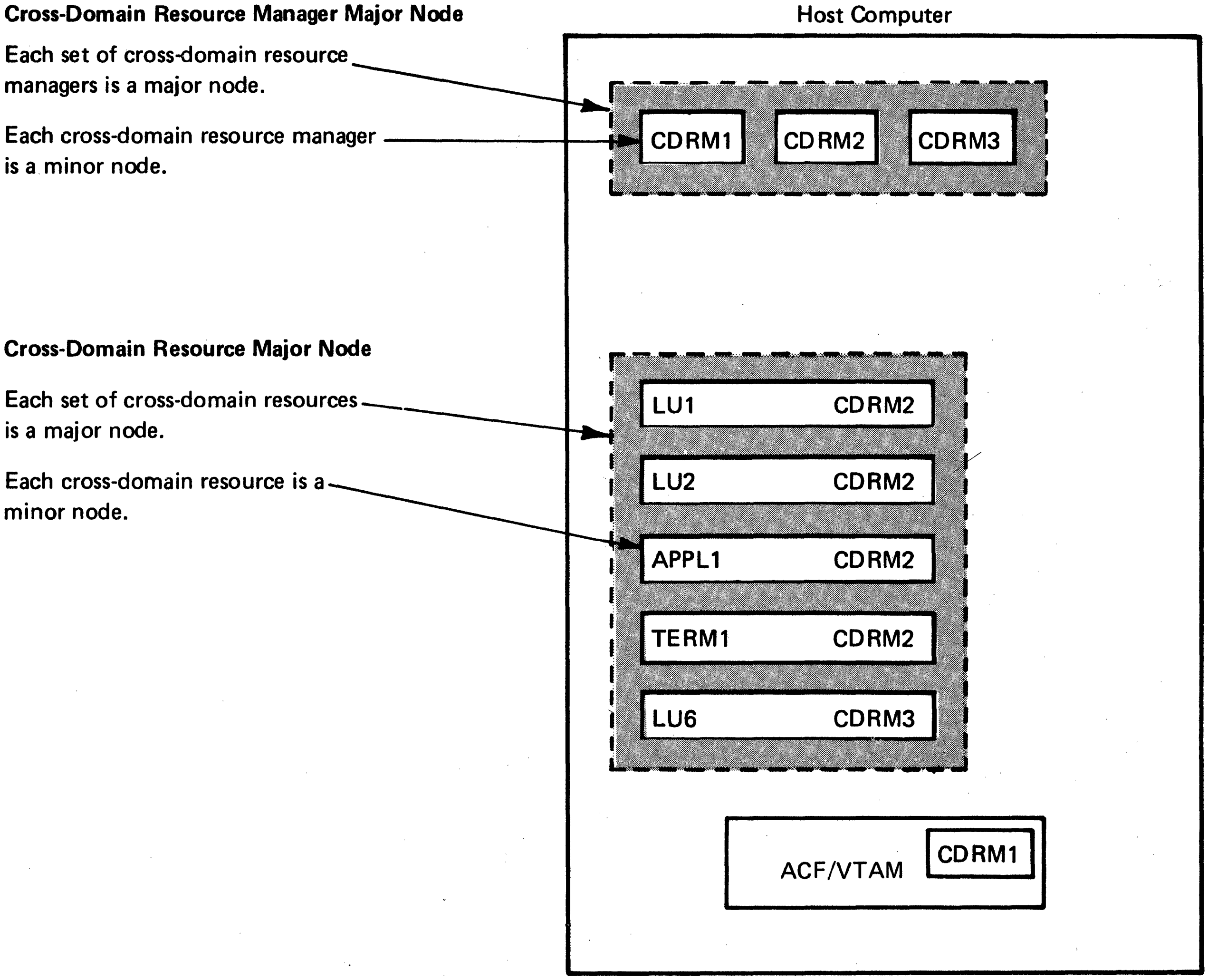

Cross-Domain Resource Manager Major Node
Each set of cross-domain resource managers is a major node.
Each cross-domain resource manager is a minor node.
Cross-Domain Resource Major Node
Each set of cross-domain resources is a major node.
Each cross-domain resource is a minor node.
Host Computer
CDRM1
CDRM2
CDRM3
LU1
CDRM2
LU2
CDRM2
APPL1
CDRM2
TERM1
CDRM2
LU6
CDRM3
ACF/VTAM
CDRM1

Figure 1-2. Additional Major and Minor Nodes in a Multiple-Domain ACF/VTAM System

In a multidomain network, information about other domains must be provided by listing the cross-domain resource managers in the network, listing the resources in the other domains that can be used, and associating them with the appropriate cross-domain resource managers.

One or more major nodes can be defined for cross-domain resource managers (CDRMs). A CDRM, which is part of the System Services Control Point in a host, is a program that controls the resources of a domain and that has the ability to communicate with other domains. ACF/VTAM and ACF/TCAM are two examples of access methods that contain CDRMs. A CDRM major node consists of one or more CDRMs. The minor nodes are the individual CDRM definitions that define the CDRM in this domain and one or more CDRMs in other domains.

One or more major nodes can be defined for cross-domain resources (CDRSCs). A CDRSC is any resource in another domain that can be used by your domain. Examples of possible CDRSCs are application programs, logical units, and BSC3270 terminals defined as logical units. A CDRSC major node consists of one or more CDRSCs. The minor nodes are the individual CDRSCs.

Differences Between Major And Minor Nodes

Major nodes are defined in a different manner than are minor nodes. Minor nodes are defined by ACF/VTAM definition statements (described later); the name of each minor node is the label coded on the corresponding statement. Major nodes are defined by combining the ACF/VTAM definition statements for the minor nodes into sets and by filing these sets as members in SYS1.VTAMLST. The name of each major node is the name assigned to the member when the statements are filed. The major node and minor node names are used to define the initial network configuration and to alter the configuration during program execution. The manner in which major nodes are activated in the initial configuration differs from the manner in which minor nodes are activated, as explained in the descriptions of the ISTATUS operand (Chapter 2) and the CONFIG start option (Chapter 3).

Physical Units And Logical Units

ACF/VTAM treats each SNA terminal system as a physical unit and one or more associated logical units. The physical unit corresponds to the controller or control unit that manages the logical unit within it and the devices attached to it.

The logical unit communicates with the host application program and with the devices attached to and managed by the physical unit. To the host application program using ACF/VTAM, the logical unit is the terminal to which it establishes connection and with which it communicates.

Subareas

Every host ACF/VTAM, and every NCP, local non-SNA, and local SNA major node must be assigned a unique identification number. The identification number is assigned by means of the SUBAREA operand that appears in the LBUILD and VBUILD statement for those major nodes and in each PCCU and HOST macro instruction, or for a host ACF/VTAM, by the HOSTSA start option. These identification numbers are referred to as subarea values throughout this publication and the *NCP Generation Manual.*

ACF/VTAM uses the subarea value to form part of the network address it assigns to each application program, line, physical unit, logical unit, and BSC or start-stop terminal. The subarea values of all active major nodes must be unique decimal numbers within the range established by the user. This range, or limit, is determined by the MAXSUBA start option, described in Chapter 3. For a more complete description of all the major installation procedures, see *ACF/VTAM Installation Guide.* Considerations in planning an ACF/VTAM network and the options and facilities of the network are discussed in *ACF/VTAM Concepts and Planning.*

Generating The Host Operating System

ACF/VTAM facilities are incorporated into a new system control program by performing an operating system generation. As a result:

ACF/VTAM support is included in the resident supervisor.

ACF/VTAM libraries can be defined.

ACF/VTAM phases are included in system libraries. The IBM-supplied network solicitor and TOLTEP are also included.

For the ACF/VTAM network, the only device support that must be specified when generating the host system is for local devices. These local devices can be SNA and non-SNA systems and communications controllers. Remote devices are not specified during host system generation. These devices are specified for NCP generation and for network definition. The latter two processes are handled as batch jobs and do not disrupt other processing. Host system generation procedures related to ACF/VTAM are discussed in *ACF/VTAM Installation Guide.*

Data Sets Used By ACF/VTAM And The NCP

These are the data sets used only by ACF/VTAM and the NCP:

SYS1.VTAMLIB is the ACF/VTAM load module library.

SYS1.VTAMLST is the ACF/VTAM network definition library.

SYS1.VTAMOBJ holds a "fast start" copy of the resource definition table built by ACF/VTAM for each major node.

Configuration restart VSAM data sets are optionally associated with NCP, local non-SNA, local SNA, and switched SNA major nodes. A configuration restart data set can also be defined to maintain a list of active major nodes.

NCP load modules are placed in a data set defined when the NCP is generated.

The NCP, if dumped, is placed in a data set defined when the NCP is generated.

ACF/VTAM and the NCP share additional data sets with other programs of the operating system.

Installing The ACF/VTAM System

Several basic processes are required to create the control programs and the control data for an ACF/VTAM system that supports a particular hardware configuration and basic user-desired options. Additional optional facilities can be written by the system programmer and included in the system. The basic processes are:

Generating the operating system for the host computer or applying the appropriate selectable unit, where applicable (this process incorporates ACF/VTAM and its support code into the operating system)

Generating the network control program for each communications controller in the network

Defining the network to ACF/VTAM

For OS/VS1 only, creating a partition for ACF/VTAM

For a summary of all the major installation procedures, see *ACF/VTAM Installation Guide.* Considerations in planning an ACF/VTAM network and the options and facilities of the network are discussed in *ACF/VTAM Concepts and Planning.*

Generating The Host Operating System

ACF/VTAM facilities are incorporated into a new system control program (SCP) by performing an operating system generation. As a result:

ACF/VTAM support is included in the resident supervisor. The IBM-supplied network solicitor and TOLTEP are also included.

The ACF/VTAM load library is defined.

ACF/VTAM modules are included in system data sets.

For the ACF/VTAM network, the only device support that must be specified when generating the host system is for local devices. These local devices can be local SNA and non-SNA systems and communication controllers. Remote devices are not specified during host system generation. These devices are specified for NCP generation and by network definition. The latter two processes are handled as batch jobs and do not disrupt other processing. Host system generation procedures related to ACF/VTAM are discussed in *ACF/VTAM Installation Guide.*

Generating The Network Control Program (NCP)

A two-stage process, called NCP generation, must be performed to create each advanced communications function network control program (NCP) to be used in a communications controller. For this process, you must prepare a set of NCP generation macro instructions, which specify the operational characteristics of the NCP and the set of devices that it will control. NCP generation produces an advanced communications function network control program that can be loaded into a communications controller.

If there is little or no variation in the way you use the network, it might only be necessary to generate one NCP for each communications controller. However, if you have several application programs that require different remote configurations, you might generate more than one NCP for each communications controller.

The NCP generation macro instructions:

Describe the communications controller

Specify ACF/VTAM and NCP options

Describe elements, such as buffer sizes and block sizes, that will affect the interaction of the operating system with the NCP

Describe lines and line groups

Describe telecommunication device configuration, devices, and device components

Most of the operands specified in the NCP generation macro instructions are used only for generating the network control program. Some of the operands, however, are used (1) both in generating the NCP and later in defining an NCP network to ACF/VTAM, or (2) only in defining the NCP network to ACF/VTAM. The jointly used operands and the ACF/VTAM-only operands are made known to ACF/VTAM when the NCP generation deck is filed during network definition.

ACF/VTAM network definition, unlike NCP generation, involves no assembly or expansion of macro instructions. The NCP generation macro instructions, in their original card-image form, are simply filed in SYS1.VTAMLST in the same manner as are the ACF/VTAM definition statements for the other types of major nodes. Thus the NCP macro instructions are true macro instructions for the purpose of NCP generation, but are statements for the purposes of ACF/VTAM definition. To avoid confusion, this book calls any "macro" that forms part of the NCP major node a macro instruction, and statements that form part of any of the other major nodes (and never pass through NCP generation) definition statements.

The NCP-only operands and the jointly used operands are verified during NCP generation, but not later during network definition or ACF/VTAM initialization. Therefore, the NCP should be generated before the NCP deck is filed to define the network to ACF/VTAM.

The information in the ACF/VTAM-only macro instructions and operands is not verified until activation of the NCP, when the resource definition table (RDT) segments are built.

The NCP must be error-free. There must be no discrepancy between the operands coded in the NCP macro instructions (which are filed in the ACF/VTAM definition library) and the generated NCP. Some errors in coding NCP macro instructions are corrected by the assembler, as by supplying omitted values, and other errors are ignored by the assembler so that generation can continue. However, stage-1 diagnostic error messages must be reviewed, and the corrected macro instructions resubmitted for stage-1 generation.

The ACF/VTAM-only macro instructions included in the NCP generation deck are:

PCCU macro instruction, which identifies the communications controller in which the NCP is to reside. At least one PCCU macro instruction is required for each NCP generation deck.

VIDLIST macro instruction, which can be used with a switched network to specify BSC identification verification to be performed by ACF/VTAM. VIDLIST also can be used with a start-stop TWX terminal for identification verification.

VTERM macro instruction, which is used instead of the NCP TERMINAL macro instruction to identify to ACF/VTAM each type (terminal and line control) of dial-in non-SNA terminal on a multiple-terminal-access (MTA) line so that the terminal type can be independently connected.

The preceding three macro instructions are described in detail in Chapter 2.

By including the ACF/VTAM-only macro instructions and operands in the NCP generation deck, the same deck can be used both to generate the NCP and to file NCP major node definition statements in the ACF/VTAM definition library. See Chapter 2 for more information on defining an NCP major node.

Each of the two stages of NCP generation is briefly described next.

Stage one is a series of assembly jobs that uses an OS/VS assembler to prepare, from the NCP generation macro instructions, a sequential data set for input to stage two. The stage one output contains data constants, macro instructions that will cause stage two to generate control tables and conditionally assemble the required program modules, job control statements for stage two, and linkage editor control statements.

Stage two assembles the control tables and any program modules that require conditional assembly, then link-edits these and other modules already in object form into an NCP load module. This module can be loaded into the communications controller with the loader provided by ACF/VTAM or with an independent loader. This stage also produces a resource resolution table load module, and if any block handling routines were coded in the NCP, a block handler set resolution table. (Block handler sets can be assigned only to BSC and start-stop terminals.) These tables contain information required for ACF/VTAM initialization and are used only by ACF/VTAM.

The generation procedure puts all load modules (NCP, resource resolution table, and block handler set resolution table) on the library specified by the LOADLIB operand of the BUILD macro instruction. ACF/VTAM must obtain the load modules from that library. The data set name of the library must be included in a DD statement in the cataloged procedure used to start ACF/VTAM. The name of the data set can be assigned a first-level qualifier by use of the QUALIFY operand of the BUILD macro instruction. The name of the NCP load module is specified by the NEWNAME=membername operand of the BUILD macro instruction. The resource resolution table load module is named membernameR and the block handling set resolution table load module is named membernameB by the second stage of NCP generation.

ACF/VTAM-related details of NCP generation appear in Chapter 2. A complete description of NCP generation in a system with virtual storage appears in the *NCP Generation Manual.*

Initial Test Routine

The ACF/VTAM user has the option to specify that a diagnostic routine, called the initial test routine, be loaded into a local communications controller and successfully executed before the NCP is loaded. The initial test routine is distributed with the NCP generation tape. Prior to starting ACF/VTAM, the IEBCOPY utility program is used to copy the routine to a partitioned data set, usually SYS1.LINKLIB, as described in the documentation accompanying the NCP generation tape.

Defining The Network To ACF/VTAM

The process of defining the network to ACF/VTAM consists of filing in SYS1.VTAMLST, the ACF/VTAM definition library, a description of each major and minor node in the system. This information is read by ACF/VTAM routines when it is started or a VARY ACT command is issued. ACF/VTAM routines then check and interpret the operands in the ACF/VTAM definition library and build the resource definition table (RDT) segments for the configuration being activated.

Filing Network Definition Decks

Information about each type of major node is filed for use by ACF/VTAM in the ACF/VTAM definition library by using the IEBUPDTE utility program. Define the major nodes as follows:

For an NCP and its attached devices, file the source deck of NCP generation macro instructions in SYS1.VTAMLST.

For an application program major node, file a VBUILD statement in SYS1.VTAMLST (for the major node), along with an APPL statement for each application program.

For a local non-SNA major node, file an LBUILD statement in SYS1.VTAMLST (for the major node), along with a LOCAL statement for each local non-SNA terminal.

For a local SNA major node, file a VBUILD statement in SYS1.VTAMLST (for the major node), along with a PU statement or LU statement for each minor node.

For a switched SNA major node, file a VBUILD statement in SYS1.VTAMLST (for the major node), along with a PU statement or LU statement for each minor node, and with a PATH statement for each dial-out path to each physical unit.

In a multidomain network environment:

For a CDRM major node, file a VBUILD statement in SYS1.VTAMLST with a CDRM statement for each minor node.

For a CDRSC major node, file a VBUILD statement in SYS1.VTAMLST with a CDRSC statement for each minor node.

File a cross-domain PATH deck in SYS1.VTAMLST.

Multiple Major And Minor Nodes

More than one NCP can be generated for each communications controller and filed in SYS1.VTAMLST under member names that are unique within the domain. Similarly, multiple sets of minor node definition statements can be filed as members of SYS1.VTAMLST under major node names chosen by the user. This enables the user to selectively include various subsets of minor nodes in the system. Major nodes that are intended to be concurrently active must not contain the same minor node names. In addition, any minor nodes that will engage in cross-domain communication must be filed under member names that are unique in the entire network.

Starting And Controlling The Network

Most ACF/VTAM operations are controlled from the system operator's console. ACF/VTAM is loaded and the network is initialized with the ACF/VTAM start procedure; then the application programs are started. ACF/VTAM loads the NCPs into the communications controllers when they are activated.

Certain ACF/VTAM facilities (some of which are optional and some of which are required) can be invoked at the start of operations. These facilities are defined by start parameters and include:

Specifying a unique identifier for ACF/VTAM (this start parameter, SSCPID, is required)

Automatically activating major and minor nodes

Specifying the details of the buffer pools used by ACF/VTAM

Starting the ACF/VTAM network solicitor

Specifying that ACF/VTAM tuning statistics be collected.

Tracing I/O, buffer content, NCP line, and buffer pool activities

Suppressing the printing of certain classes of ACF/VTAM messages at the console

To enter start options, use any combination of these methods:

The network operator can enter them at the console.

The network operator can select a list of start options that the system programmer has predefined and stored.

ACF/VTAM can use a list of IBM-supplied start options.

ACF/VTAM can use a list of start options that the system programmer has predefined and stored.

The network operator can use four network operator commands while ACF/VTAM is running. The DISPLAY command is used to display the status of major or minor nodes, and of other ACF/VTAM resources. Two other commands, VARY and MODIFY, are used to change the configuration of the network and its operational characteristics. VARY commands are used to:

Activate or deactivate nodes (*ACF/VTAM Concepts and Planning* describes the requirements for activating and deactivating major nodes and minor nodes)

Request connection between an application program and a terminal, or between application programs

Acquire and release physical units in this domain for the purpose of switched network backup

When the Multisystem Networking Facility is installed, acquire and release resources in another domain

MODIFY commands are used to:

Start or stop the network solicitor

Start or stop the I/O trace, buffer content trace, and NCP line trace

Start or stop the ACF/VTAM internal trace

Start or stop the tuning statistics trace

Start or stop the buffer use trace (SMS trace)

Start or stop the Teleprocessing Online Test Executive Program (TOLTEP)

Request a dump of the contents of a communications controller's storage

The HALT command is used to close down ACF/VTAM and its domain.

You can authorize application programs to issue VARY, DISPLAY, MODIFY, and REPLY commands, as described in *ACF/VTAM Program Operator Guide.* The network operator commands are discussed in detail in *ACF/VTAM Network Operating Procedures.*

Optional ACF/VTAM Features

In addition to tailoring ACF/VTAM to a particular network configuration by means of the NCP and the major and minor node definition statements, you can further modify ACF/VTAM by selecting options and by replacing IBM-supplied facilities. The optional facilities allow the user to:

Select and file start options that specify ACF/VTAM starting conditions, including the initial network configuration

Select the means by which each terminal can be logged on

Tailor the network solicitor, or replace it with a user-written routine that monitors interactive and input-only non-SNA terminals and processes their requests to be logged on to application programs

Define interpret tables, USS definition tables, and logon mode tables

Write and install routines to collect accounting information and to control connections between application programs and terminals

File generalized trace facility options, including those required for the ACF/VTAM trace facility

Mark the NCP load program not reusable to reduce ACF/VTAM startup time when more than one NCP is to be loaded at the same time. (This is done at the expense of increased ACF/VTAM storage requirements.)

Explanations of related macro instructions, operands, and procedures appear later in this manual.

Exit Routines

The user can write exit routines to authorize or reject connection requests and to add accounting facilities to the system. When control is passed to them, ACF/VTAM makes available information that can be used by these routines. See Chapter 5 for a description of the accounting and authorization exit routines and the parameters passed to them.

Reliability, Availability, And Serviceability Facilities

The reliability, availability, and serviceability (RAS) facilities of ACF/VTAM are designed to be used as aids in maintaining the operation of the network. They include:

When the operating system is providing a dump of ACF/VTAM, key ACF/VTAM control blocks are formatted. The formatting routine is provided for dumps resulting from abnormal termination (ABEND) of ACF/VTAM itself, from execution of the ABEND or SNAP macro instructions in an application program, or from the OS/VS stand-alone dump.

The contents of a local or remote communications controller can be dumped by the network operator whenever desired or automatically in the event of an unrecoverable communications controller or NCP failure.

The contents of an acquired link-attached communications controller in another domain cannot be dumped.

Error recovery procedures and the recording of errors, both permanent and temporary, are initiated automatically.

Buffer and I/O traces, which record buffer content and I/O activity within ACF/VTAM

NCP line trace, which records operating parameters of a line operating in network control mode

Network control program dump

Configuration restart, which reestablishes major nodes in the network to either initial status or to preinactive status after deactivation; after loss of contact with a physical unit; or after a failure of the host operating system, host computer, communications controller, NCP, or ACF/VTAM

Teleprocessing online test executive program (TOLTEP), a program that uses online tests (OLTs) to invoke hardware tests on selected terminals (TOLTEP is discussed briefly in Chapter 7 and in detail in *ACF/VTAM TOLTEP*)

If the Multisystem Networking Facility is installed, then if ACF/VTAM or its host computer fails, its resources can be taken over by an adjacent domain.

Chapter 2. Defining The Network

Before ACF/VTAM can use a data communication network, you must define the application programs and the physical configuration of the network to ACF/VTAM. These definitions represent points or nodes in the network that can be addressed and used by application programs (using ACF/VTAM macro instructions) and by the network operator (using network operator commands). Defining the network to ACF/VTAM involves:

Defining application program major nodes

Defining local non-SNA major nodes

Defining local SNA major nodes

Defining switched SNA major nodes

Defining NCP major nodes

Setting up major nodes for nonswitched line backup

Defining the network in a multiple-domain ACF/VTAM system requires the steps listed above and the following:

Defining cross domain resource manager major nodes

Defining cross domain resource major nodes

Defining path tables

If the ACF/VTAM Encrypt/Decrypt Feature is installed, certain additional operands can be coded on some ACF/VTAM definition statements and macro instructions. These operands are described in this book. If you plan to install the ACF/VTAM Encrypt/Decrypt Feature, you should first read *IBM Cryptographic Subsystem Concepts and Facilities,* GC22-9063.

Code these definitions of major and minor nodes by using ACF/VTAM definition statements or NCP macro instructions, and put the definitions into SYS1.VTAMLST as separate members (major nodes).

Unique names must be used in all ACF/VTAM-related definitions that will be active at the same time. (For example, the name on a statement defining a minor node must not be the same as the member name.) In addition, the following names must not be used for major or minor node names:

Any name begining with the characters IST.

NETSOL (except as noted under description of NETSOL macro instruction)

PUNS

VTAM

VTAMSEG

VTAMTERM

In VTAM Level 2, if a major node definition in SYS1.VTAMLST is changed after the initial activation of that major node, it is necessary to delete the corresponding member of SYS1.VTAMOBJ before activating the changed major node. This same procedure is required for ACF/VTAM if SYS1.VTAMLST is concatenated. However, if SYS1.VTAMLST is *not* concatenated, the following method is used when activating a major node.

If the major node has a SYS1.VTAMOBJ member, ACF/VTAM checks to see if the corresponding SYS1.VTAMLST member has been changed since the SYS1.VTAMOBJ member was created. If the SYS1.VTAMLST member has been changed, the SYS1.VTAMOBJ member is replaced with one that matches the new SYS1.VTAMLST member. Otherwise, the SYS1.VTAMOBJ member is used unchanged.

To use the new method for activating major nodes that have been changed, observe these restrictions:

- Ensure that SYS1.VTAMLST is not concatenated.
- When modifying a member of SYS1.VTAMLST, ensure that the major node associated with that member is not activated while the modifications are in progress.
- Do not use the first eight bytes of the user-data area of the directory entries for SYS1.VTAMLST. ACF/VTAM uses these bytes to store information that allows ACF/VTAM to determine whether SYS1.VTAMOBJ must be updated to reflect changes to SYS1.VTAMLST.

Coding Conventions

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

Coding Rules

The rules summarized here are assembler language rules.

System definition macro instructions and statements have the following format:

Name	Operation	Operand
Symbolic name	Operation code of the macro instruction or statement	Required and optional operands

The Name field symbolically identifies the macro instruction, definition statement, or minor node. If a symbolic name is provided in the field, it can contain from one to eight alphanumeric characters, the first of which must be alphabetic or the national character @ or #. The name must begin in the first position of the macro instruction or statement and must be followed by one or more blanks.

The Operation field identifies the macro instruction or statement. It must be preceded and followed by one or more blanks.

The Operand field contains operands coded in any order and separated by commas. The operand field ends with one or more blanks placed after the last operand. When the syntax of a macro instruction or statement is shown in this book, any operands that are always required appear first, followed by the optional or conditional operands. In most macro instructions or statements, keyword operands are used in the operand field. Keyword operands are often followed by an equal sign (=) and the keyword value. The keyword value can be a single value or a list of values. If it is a list of values, the values must be separated by commas and the list must be enclosed in parentheses.

Comments can be written after the operand field, but they must be separated from the last operand of the operand field by one or more blanks. An entire card can be used for a comment by placing an asterisk in the first column of the card. A macro instruction that has no operands cannot have comments on the same card as the operation code.

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

Restrictions On Use Of Assembler Features

The NCP generation macro instructions and the definition statements are coded in standard operating system macro instruction format, as just described, with the following restrictions:

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

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

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

User assembler macro instructions that generate NCP macro instructions are not permitted.

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

Variable substitution at assembly time is not permitted.

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

Use of literals is not permitted.

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

Comments, statements, or remarks can be used in decks for all the types of major nodes.

Errors made in the major node definition decks filed in the ACF/VTAM definition library result in messages to the system operator's console during ACF/VTAM initialization or VARY ACT command processing, rather than to the SYSOUT device.

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

Describing Macro Instructions And Definition Statements

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

Capital letters represent values that are coded directly, without change. Brackets [ ], braces { }, "or" bar | , ellipses . . ., and subscripts are never coded.

Small letters represent operands for which a value or name must be supplied.

Brackets [ ] enclose operands or symbols that are either optional or conditional. Conversely, the lack of brackets indicates that an item or group of items must be coded.

An optional operand is one that can be coded or omitted independently of other operands that are coded or omitted. Depending on the operand, omitting it might cause the corresponding feature or function to be omitted or included, or omitting it can cause a specific value (the default value) to be assigned.

A conditional operand is one that might be coded or omitted, depending on how other operands are coded (or omitted) in the same macro instruction or statement, or in a different one. For each conditional operand, the conditions under which it should be coded or omitted are indicated.

Braces { } indicate that an operand must be given one of the values shown within the braces.

A vertical "or" bar (|) between operands indicates that one operand must be coded from among the values separated by the "or" bar.

An ellipsis (. . .) indicates that a series of values can be coded in the same format that precedes the ellipsis.

Parentheses and commas are coded as shown.

An underlined value represents the value that ACF/VTAM or the NCP uses if the operand is omitted.

Symbols coded in the name field of a macro instruction or definition statement must not begin with a $ character.

Single quotation marks or ampersands in character strings must be coded as two adjacent single quotation marks or two ampersands.

"Sift-Down" Effect In NCP Macro Instructions

The "sift-down" effect applies to many of the keyword operands in the NCP macro instructions. This means that if an operand is coded at a high level, there is no need to code that same operand for all lower levels for which the same value is desired. Note that for such operands, if a keyword operand is specified in a higher-level macro instruction, the IBM-supplied value is available in a lower-level macro instruction only by specifically coding that value. Also, if an operand is coded in a lower-level macro instruction, its value completely overrides (for that instruction) the value of the same operand specified in a higher-level macro instruction. If one of the suboperands is missing, the default value of the suboperand is used, not the sift-down value from a higher-level macro instruction.

Also note that a keyword does not necessarily apply to every macro instruction on which it can be coded. Some operands do not apply to or have any effect on higher-level macro instructions, but can be coded on them anyway for sifting purposes.

The time required to initialize ACF/VTAM can be reduced by taking advantage of the sift-down effect. For information about macro instruction sequencing and the sift-down level for each NCP operand, refer to *NCP Generation Manual.*

Defining Application Program Major Nodes

Each application program must be defined to ACF/VTAM as part of a logical set (group) of one or more application programs. This logical set or group is called an application program major node. The same application program can be included in more than one major node, but only one major node containing that name can be active at one time.

An application program major node is defined by filing one VBUILD statement for the major node and a separate APPL statement for each application program in the major node.

The VBUILD Statement

Name	Operation	Operand
[name]	VBUILD	TYPE=APPL

name
is any symbol valid in the assembler language and is optional. ACF/VTAM does not check this name for validity.

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

The APPL Definition Statement

Code one APPL definition statement (in 80-byte card-image format) for each application program that is in the ACF/VTAM network and file the APPL major node as a member of SYS1.VTAMLST.

For OS/VS2 MVS, if TSO/VTAM is being defined, refer to Appendix E for the TSO/ VTAM requirements when coding the APPL definition statement.

The format of the APPL definition statement is:

Name	Operation	Operands
name	APPL	[,ACBNAME=acbname] [,AUTH=([ACQ | NOACQ] [,BLOCK | NOBLOCK] [,PASS | NOPASS] [,PPO | SPO | NOPO] [,TSO | NOTSO] [1] [,VPACE |NVPACE])] [,AUTHEXIT=YES | NO] [2] [,BUFFACT=n | 1] [,DLOGMOD=default logmode entry name] [,EAS=n | 404] [,ENCR=SEL | REQD | OPT | NONE] [3] [,MODETAB=logon mode table name] [,PRTCT=password] [,VPACING=n | 0 | 1]

[1] For OS/VS2 MVS only
[2] For OS/VS1 and OS/VS2 SVS only
[3] For ACF/VTAM Encrypt/Decrypt Feature only

name

specifies a minor node name for this application program, and is required. No two active application programs in the network can have the same name.

The name must begin with an alphabetic character other than $ and if the ACBNAME operand is not coded it must be identical to the name specified in the APPLID operand of the ACB macro instruction for this application program. If no application program name is coded in the APPLID operand of the ACB macro instruction, the name in the ACBNAME operand must be the job-step name under which the application program is started. This can be done for only one application program that is active within a job-step.

If the Multisystem Networking Facility is installed, this network-unique name can be used for cross-domain terminal logons in cases where there are more than one copy of the same application program in the network. By specifying the network-unique name of the application program, a terminal user can gain access to a specific copy of the application program.

If there is only one copy of the application program in the network or if the application program is in the same domain as the logical unit that wants to use it, the name specified by the ACBNAME operand can be used instead.

The ACB macro instruction is described in *ACF/VTAM Macro Language Reference.*

ACBNAME=acbname

specifies the minor node name (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) assigned to this application program. This name must be unique within the domain in which the application program resides and can be used for terminal-logon only by a terminal, logical unit, or application program that is in the same domain as this application program. The name must begin with an alphabetic character other than $ and must also be identical to the name specified in the APPLID operand of the ACB macro instruction for this application program. If no application program name is coded in the APPLID operand of the ACB macro instruction, the name in the ACBNAME operand must be the job-step name under which the application program is started. This can be done for only one application program that is active within a job-step.

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

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

indicates whether this application program has the authority to use certain ACF/VTAM functions. The operands can be specified in any order and only one comma is needed to separate one operand from the next. The possible operands are listed below.

ACQ | NOACQ

indicates whether this application program can use either the SIMLOGON macro instruction or the OPNDST macro instruction with the ACQUIRE option. (These macro instructions enable the application program to acquire a connection with a particular terminal).

BLOCK | NOBLOCK

indicates whether this application program can request input data from start-stop or BSC terminals in blocks instead of in messages or transmissions. (This occurs for read specific or solicit operations.) This operand does not apply to devices that use the record mode of data transfer or to the terminals in the IBM 3270 Information Display System.

This operand corresponds to the BLOCK operand in the NIB (node initialization block) that represents the terminal (PROC=BLOCK in the NIB macro instruction).

When this application program issues an OPNDST macro instruction, ACF/VTAM compares this operand on the APPL definition statement with the operand in the NIB. If NOBLOCK is coded or assumed by default in the APPL definition statement, the application program cannot specify the BLOCK option in the OPNDST macro instruction.

PASS | NOPASS

indicates whether this application program can use the CLSDST macro instruction with the PASS option to pass connections to another application program. For more information on the CLSDST macro instruction, refer to *ACF/VTAM Macro Language Guide.*

If PASS is not specified on the APPL definition statement, the CLSDST macro instruction cannot be used to request ACF/VTAM to reconnect terminals to another application program (after disconnecting the terminals from this application program).

PPO | SPO | NOPO

specifies the status of the application program in regard to issuing network operator commands and receiving responses and unsolicited messages. This facility is described in *ACF/VTAM Program Operator Guide.*

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

If PPO is coded, the application program receives all unsolicited messages; that is, all messages such as informational and error messages that are not replies to operator commands. If SPO is coded, all unsolicited messages are directed to the PPO, if active, or to the system console. (Only if no PPO-designated application program is active when unsolicited messages occur will the unsolicited messages be directed to the system console.)

TSO | NOTSO (For OS/VS2 MVS only)

indicates whether or not the application program is a TSO/VTAM time-sharing program. If it is a TSO/VTAM time-sharing program, code TSO. If not, code NOTSO or allow that operand to take effect by default.

If AUTH=TSO is specified, all user exit routines receive control in supervisor state and in Key 6. If AUTH=TSO is specified, user exit routines must:

Return directly to ACF/VTAM in supervisor state

Exit with the same ESTAE environment as at entry. One ESTAE DELETE command must be issued for each ESTAE CREATE that was issued. ESTAE overlays must not be issued, except for a test application program exit routine that resulted from an ESTAE CREATE.

Not use SPIE

VPACE | NVPACE

specifies whether this application program is to be subject to the VPACING specifications of logical units with which the program will be in session. A specification of NVPACE is effectively the same as specifying VPACING=0 in the LU statements for all of the logical units with which the application program will be in session. A specification of NVPACE is ignored for sessions with logical units in a local major node.

VPACE is normally specified to prevent overloading buffers in the 3704 or 3705 with outbound messages from ACF/VTAM. However, NVPACE can be specified in two situations:

1. When the application program sends only single-element messages to any one logical unit and, after sending each message, the program waits for a response before sending the next message.
2. When the application program sends chains containing a limited number of elements and the program either (1) after sending one chain, waits for a response before it starts sending the next chain, or (2) sends the Change Direction Command indicator in the last element of each chain. The number of elements in each chain must be no larger than the results of this formula:

   number of elements = 2n – 1

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

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

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

(2 X 3) – 1 = 5

For information on the VPACING operand, see "LU (Local) Statement," "LU (Switched) Statement," and "Pacing" later in this chapter.

AUTHEXIT=YES | NO (For OS/VS1 and OS/VS2 SVS only)

specifies whether this application program is authorized to use expedited processing in its exit routines. If AUTHEXIT=YES is specified and if this application program's job-task name is included in APFLIB, this application program's exit routines will receive control by branch entry, rather than by a SYNCH.

BUFFACT=n | 1 (Basic Mode Only)

specifies a multiplier (decimal integer in the range 1 through 255) to be used in conjunction with the value specified in the BUFLIM operand of the LOCAL, TERMINAL, VTERM, and COMP statements. The product of the BUFFACT and BUFLIM values specifies the number of buffers that can be filled with data that has been obtained by ACF/VTAM but that has not yet been transferred to this application program's buffers.

For example, if BUFFACT=1 and BUFLIM=2 (the default values), then 2 becomes the maximum number of buffers that can be allocated to receive input from that terminal when it is connected to this application program.

After ACF/VTAM multiplies the BUFLIM and BUFFACT values together, the product becomes effective after this application program issues the OPNDST macro instruction.

If the amount of data read in exceeds the calculated maximum capacity of the elements for basic devices, ACF/VTAM issues a RESET Unconditional macro instruction. This macro instruction cancels the I/O operation, but the excess data remains available to be read.

To determine if the data was lost, the application program must check the return codes in the RPL fields for the I/O request (RTNCD and FDBK2). The RPLs to be checked are those for the DO, READ, or WRITE (OPTCD=CONV) macro instructions.

Consider the following information when determining the BUFFACT value:

Number of terminals expected to be in concurrent use

Characteristics of the terminals (such as the terminal's hardware features, speed, and use or type of transaction it is to perform)

Characteristics of this application program to be used with the expected terminals (such as updating or retrieving data)

The value coded for MAXDATA in the PCCU macro instruction

DLOGMOD=default logmode entry

specifies the name of the logmode entry to be used by default if one is not otherwise provided.

A logmode entry specifies which entry in the applicable logon mode to be used to provide a set of session parameters for the application program. The name specified for the DLOGMOD operand must be the name of an entry in a logon mode table. A logon mode table is not specified for the application program, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table.

If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logon mode table is used by default.

For more information, see the description of MODETAB later in this section, and also see Chapter 4.

EAS=n | <u>404</u>

specifies the approximate number of concurrent sessions this application program will have with its logical units (LU-LU sessions). ACF/VTAM uses this operand in a rapid-lookup scheme to find the representation of a session between the application program and a logical unit. Specify n as a value from 0 to 32767. If 0 is specified, ACF/VTAM uses EAS=404.

A value should always be specified for n if the application program is to be in session with any logical units. The value should be 10% to 20% greater than the estimated number of sessions the application program will have. If the actual number of concurrently active sessions with logical units is greater than the number specified in EAS, or if the actual number is greater than 8080, the ACF/VTAM mainline path-length is increased.

ENCR=REQD | SEL | OPT | <u>NONE</u>

can be specified only if the ACF/VTAM Encrypt/Decrypt Feature is installed. It specifies whether this application program has any special requirements for enciphering and deciphering messages.

REQD

specifies that ACF/VTAM must encipher all the messages that this application program sends and that ACF/VTAM must decipher all the messages sent to the application program. ENCR-REQD also specifies that this application program cannot start a session with a logical unit that is not able to encipher and decipher its messages. If ENCR=REQD is coded, this application program cannot be activated unless the host in which it resides is capable of handling cryptographic sessions.

SEL

specifies that this application program can choose which messages are to be enciphered by ACF/VTAM. ENCR=SEL also specifies that this application program cannot start a session with a logical unit that is not able to encipher and decipher its messages. If ENCR=SEL is coded, this application program cannot be activated unless the host in which it resides is capable of handling cryptographic sessions.

OPT or NONE

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

MODETAB=logon mode table name

specifies the logon mode table to be used to associate each logon mode name with a set of session parameters for the application program. The name specified for the MODETAB operand must be the name of a logon mode table created as described in Chapter 4. If a logon mode table is not specified for the application program, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table.

PRTCT=password

specifies a one to eight EBCDIC character password. ACF/VTAM compares this password to the one in the application program's ACB (access method control block) during Open processing and uses it to verify the authority of this application program to run (as the program being defined by this APPL definition statement). If this operand is not specified, no password checking is done.

VPACING=n | 0 | 1

specifies a decimal integer in the range 0 through 63 that indicates the maximum number of normal-flow requests that another logical unit can send to this application program in a session before waiting to receive a pacing response. This value, which controls the pacing of requests to the application program, is exchanged when a session is established. If the value is not in the range 0 through 63, the maximum of 63 is used. If VPACING=0 is specified, no pacing of requests to the application program is done.

If VPACING is not specified, the default value of 1 is used.

For more information on pacing, refer to the section "Pacing" later in this chapter.

Defining Local Non-SNA Major Nodes

ACF/VTAM definition statements define each local non-SNA terminal as part of a logical set (group) of local non-SNA terminals.

The LBUILD Definition Statement

One LBUILD definition statement (in 80-byte card-image format) must be specified for each logical group (major node) of local non-SNA terminals. Define each terminal (minor node) in the group by using a LOCAL definition statement.

The format of the LBUILD definition statement is:

Name	Operation	Operand
[name]	LBUILD	SUBAREA=n | _3_ [,CONFGDS=ddname] [,CONFGPW=password]

name

one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character. Specification of a name is optional.

SUBAREA=n

is a decimal number chosen by the user to identify the major node's subarea value.

Specify n as an integer in the range 3 through the value that is to be specified in the MAXSUBA start option. See Chapter 3, "Starting and Controlling the Network," for a description of the MAXSUBA start option and a description of the relationship between the SUBAREA and MAXSUBA values.

When a local non-SNA major node is activated, the SUBAREA value must be unique among the active major nodes. That is, the same SUBAREA value can be assigned to more than one major node, but only one of them can be active at a time.

CONFGDS=ddname

is a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure.

Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities", for a discussion of configuration restart.

CONFGPW=password

specifies the one to eight character password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified but is required by VSAM, VSAM prompts the network operator to provide the correct password when ACF/VTAM attempts to use the data set.

This operand can be specified only if the CONFGDS operand is specified.

The LOCAL Definition Statement

One or more LOCAL definition statements can be grouped with an LBUILD definition statement and put together in SYS1.VTAMLST.

Code one LOCAL definition statement (in 80-byte card-image format) for each local non-SNA terminal that is in the ACF/VTAM network.

The format of the LOCAL definition statement is:

Name	Operation	Operand
name	LOCAL	CUADDR=address ,TERM={3277 | 3284 | 3286} [,BUFLIM=n | 2] [,DLOGMOD=default logmode name] [,FEATUR2= ([MODEL1 | MODEL2] [,ANKEY | NOANKEY] [,DEKEY | NODEKEY] [,PFK | NOPFK] [,SELPEN | NOSELPEN])] [,ISTATUS=ACTIVE | INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name]

name

indicates the unique, one to eight alphanumeric character node name assigned to the device (whose address is specified in the CUADDR operand in this LOCAL definition statement). The name must begin with an alphabetic character other than a $ character.

CUADDR=address

indicates the hexadecimal channel unit address for this local terminal. Note that you must not enclose the address within quotation marks or apostrophes.

TERM=3277 | 3284 | 3286

indicates the specific, local non-SNA terminal (printer or display station component). Only 3277, 3284, or 3286 can be specified. For a list of devices that are supported as a 3277, 3284, or 3286, see *ACF/VTAM Concepts and Planning.*

BUFLIM=n | 2 (Basic Mode Only)

ACF/VTAM uses this multiplier (integer number in the range 1 through 255) to determine the maximum number of buffers that can be filled with data that has been read ahead for this terminal, but has not yet been transferred into the application program's buffers. The maximum number of buffers is the product of this BUFLIM value and the value coded in the BUFFACT operand (in the APPL definition statement that defines the application program being executed).

For example, if BUFFACT=1 (the default value) and BUFLIM=2 (the default value), 2 becomes the maximum number of buffers that can be allocated to receive input from the terminal.

After ACF/VTAM multiplies the BUFLIM and BUFFACT values together, the product becomes effective after the application program issues the OPNDST macro instruction.

If the amount of data read in exceeds the calculated maximum capacity of the buffers, ACF/VTAM issues a RESET macro instruction with the OPTCD=UNCON operand. This macro instruction cancels the I/O operation, and the excess data is lost.

To determine if data was lost, application programs must check the return codes in the RPL fields for the I/O request (RTNCD and FDBK2). The RPLs to be checked are those for the ACF/VTAM macro instructions DO, READ, WRITE, or WRITE (OPTCD=CONV).

Consider the following information when determining the BUFLIM value:

Number of terminals expected to be in concurrent use

Characteristics of the terminals (such as the terminal's hardware features, speed, and use or type of transaction it is to perform)

Characteristics of the application programs expected to be used with the terminals (such as updating or retrieving data)

DLOGMOD=default logmode entry name

specifies the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default. For more information on logmode entries, see Chapter 4.

FEATUR2=([operand] [,operand]. . .)

indicates the machine features for a specific, local non-SNA terminal. The operands are listed below.

MODEL1 | MODEL2

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

If TERM=3277 is specified on this LOCAL definition statement, the following operands can be coded:

ANKEY | NOANKEY

indicates whether this terminal has a standard alphanumeric keyboard.

DEKEY | NODEKEY

indicates whether this terminal has the data entry keyboard.

PFK | NOPFK

indicates whether this terminal has the program function keys.

SELPEN | NOSELPEN

indicates whether this terminal has the selector pen feature.

ISTATUS=ACTIVE | INACTIVE

indicates whether this terminal (minor node) is to be initially active when the local non-SNA major node to which it belongs is first activated. (Major nodes can be activated either when ACF/VTAM is started or, following the start of ACF/VTAM, by issuing the VARY command.)

When ISTATUS=ACTIVE is coded or assumed by default, automatic logon for this terminal to an application program occurs when the major node and application program are both active (if automatic logon has been specified for the terminal or logical unit).

When ISTATUS=INACTIVE is coded, the first activation of an inactive major node containing this terminal leaves the terminal inactive. Later activations of an already active major node activates all the terminals not previously active (including the ones specified INACTIVE).

LOGAPPL=application program name

indicates the name of an application program (one to eight alphanumeric characters) to which this terminal is to be automatically logged on when it is activated. The name must correspond to the network-unique name assigned to the application program by the name field of the APPL definition statement.

If this operand is not coded, either the application program or the network operator must initiate terminal logons.

If you use a network solicitor to monitor logons from terminals that use the basic mode of data transfer, code 'application program name' as the name assigned to that network solicitor. If the terminal is an output-only device, such as a printer, do not code NETSOL (the name of the IBM-supplied network solicitor) for this operand. The IBM-supplied network solicitor does not control output-only devices because they cannot be sources of logon requests.

LOGTAB=interpret table name

indicates the member name of an interpret table. For information on interpret tables, see Chapter 4.

LOGTAB permits a terminal user to initiate a logon from this terminal and associates the specified interpret table with this terminal.

If you do not use OS/VS logon, then LOGTAB must be coded in this LOCAL definition statement. For information on OS/VS logon, see Chapter 4.

If this terminal is to be controlled by a network solicitor, LOGTAB should be coded in this LOCAL definition statement. This allows the selected network solicitor to use the specified interpret table (to validate the logon message before routing it to the specified application program).

MODETAB=logon mode table name

specifies the logon mode table to be used to correlate each logon mode name with a set of session parameters. The name specified for the MODETAB operand must be the name of a logon mode table created as described in Chapter 4.

If this operand is not coded, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table.

Defining Local SNA Major Nodes

A local SNA major node is defined by filing a single VBUILD statement for the major node and a separate PU or LU statement for each minor node. One VBUILD statement must be included for each major node, placed before the first PU statement. The VBUILD statement assigns a subarea value to the major node for ACF/VTAM's use in assigning addresses to the minor nodes.

A PU statement is required for each physical unit (such as an SNA controller) in the major node. An LU statement for each logical unit is placed under the associated PU statement.

The PU and LU statements used to define a local SNA major node are very similar to the PU and LU statements used to define a switched SNA major node, and to the PU and LU macro instructions used to define an NCP major node. Where possible confusion might arise, the statements used for the local SNA major node are called PU (local) and LU (local) statements.

You can define multiple sets of local SNA cluster controllers. This allows the network operator to use the VARY command to activate a selected subset of all the local SNA controllers. However, all major and minor node names known to ACF/VTAM at any one time must be unique. Two major nodes that include the same physical unit or logical unit cannot be active at the same time. Refer to *ACF/VTAM Concepts and Planning* and *ACF/VTAM Network Operating Procedures* for further information.

The VBUILD Statement

Code a VBUILD statement in 80-byte card-image format for each set of locally attached SNA devices.

Write the VBUILD statement as follows:

Name	Operation	Operand
[name]	VBUILD	SUBAREA=n ,TYPE=LOCAL [,CONFGDS=ddname] [,CONFGPW=password]

name
one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. Specification of a name is optional.

SUBAREA=n
is a decimal number chosen by the user to identify the major node's subarea value.

Specify n as a decimal integer with a value of at least 3, but not greater than the value that is to be specified in the MAXSUBA start option. See Chapter 3, "Starting and Controlling the Network", for a description of the MAXSUBA start option.

When the local SNA major node is activated, the value must be unique among the active major nodes.

TYPE=LOCAL
specifies that the VBUILD statement defines a local configuration (as opposed to a switched configuration) to ACF/VTAM. This operand is required.

CONFGDS=ddname
ıs a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure. Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities" for a description of configuration restart.

CONFGPW=password
specifies the one to eight character alphanumeric password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified but is required by VSAM, the network operator is prompted by VSAM to provide the correct password when ACF/VTAM attempts to open the data set. This operand can be specified only if the CONFGDS operand is specified.

The PU (Local) Statement

Code a PU statement in 80-byte card-image format for each physical unit in the local SNA major node.

Write the PU statement as follows:

Name	Operation	Operand
name	PU	[CUADDR=address] [,DISCNT=([YES | NO] [,F | NF])] * [,DLOGMOD=default logon mode entry] [,ENCR=REQD | SEL | OPT | NONE] * [,ISTATUS=ACTIVE | INACTIVE] [,LOGAPPL=application program name] * [,LOGTAB=interpret table name] * [,MAXBFRU=number | 1] [,MODETAB=logon mode table name] * [,PUTYPE=2] [,SSCPFM=FSS | USSSCS] * [,USSTAB=USS definition table name] * [,VPACING=n | 0 | 1] *

*These operands can be specified in either the PU or the LU statement. The operands are meaningful for logical units, and the operands' explanations appear in the LU statement description. Coding one of these operands in a PU statement is equivalent to coding the operand in each of the associated LU statements. If an operand with a different value is then coded in an LU statement, the value in the LU statement overrides (for that logical unit) the value coded in the PU statement.

name

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. The symbolic name provides the minor node name of the physical unit and is required.

CUADDR=address

specifies the hexadecimal channel unit address to be used when activating the physical unit. If CUADDR is omitted, ISTATUS=INACTIVE must be specified and CUADDR must be included in the VARY command that activates the unit.

The value specified must match a channel unit address specified when the operating system is generated. The address must not be enclosed in quotation marks or apostrophes.

DISCNT=([YES | NO] [,F | NF])

YES | NO

specifies whether ACF/VTAM is to physically disconnect the physical unit when the last logical unit is disconnected by its application program (that is, when there are no more application program-logical unit communication sessions).

For a locally attached physical unit, disconnection means that the channel connection is broken (in effect, the device is set to nonoperational), and the SSCP-PU session is terminated. However, the device is still allocated to ACF/VTAM and is still active. (That is, sessions can be requested with its logical units; such a request causes a physical connection to be reestablished. The physical unit itself may also request a physical reconnection.)

YES
indicates that ACF/VTAM is to automatically disconnect the physical unit as soon as the last logical unit is disconnected by its application program. If any logical units request their own disconnection, ACF/VTAM ignores the part of their disconnection request that indicates whether the physical unit is to be disconnected (that is, the HOLD part of a character-coded logoff command or the LAST-NOTLAST part of a field-formatted Terminate Self command is ignored). ACF/VTAM also rejects any attempt made by the physical unit to request its own disconnection (using the Request Discontact command).

NO
indicates that ACF/VTAM is to disconnect the physical unit when one of the following conditions is met:

ACF/VTAM receives a Request Discontact Normal command from the physical unit and all logical units have been disconnected and there are no more application program to logical unit sessions. Existing sessions are allowed to end normally.

ACF/VTAM receives a Request Discontact Immediate command from the physical unit. ACF/VTAM immediately terminates existing application program to logical unit sessions. This command overrides any previous Request Discontact Normal command from the physical unit.

All logical units have been disconnected as a result of a character-coded logoff command for which HOLD=NO was specified (or inserted using a USS definition table), a Terminate Self command for which LAST was specified, or a VARY INACT command. If any logical unit was disconnected by any other means (for example, by an application program CLSDST), the physical unit is not disconnected.

F|NF
specifies whether ACF/VTAM is to indicate "final-use" status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES. This operand does not apply when DISCNT=NO is specified, nor does it have any effect on the VARY INACT command. If F is specified or assumed by default, "final-use" status is indicated. If NF is specified, "not-final-use" status is indicated. Each device has its own requirements regarding "final-use" status. To determine whether F or NF should be specified for a given device, consult the appropriate installation publication for the device.

ISTATUS=<u>ACTIVE</u>|INACTIVE
specifies whether the physical unit is to be activated when its major node is activated following the first start of ACF/VTAM or a cold restart of ACF/VTAM (that is, a restart to initial status).

MAXBFRU=n | 1
specifies the number of buffer units (elements of the IOBUF buffers) that will be used to receive data from the physical unit.

PUTYPE=2
specifies the physical unit type. Only PUTYPE=2 (the default) is supported. Physical unit types are described in *ACF/VTAM Concepts and Planning.*

The LU (Local) Statement

Code an LU statement in 80-byte card-image format for each logical unit associated with a physical unit within a local SNA major node. The LU statement must follow the PU statement that defines the physical unit with which the logical unit is associated. As for NCP generation, LU statements must be in ascending order according to the value specified for LOCADDR.

Write the LU statement as follows:

Name	Operation	Operand
name	LU	LOCADDR=n [,DLOGMOD=default logmode entry] [1] [,ENCR=REQD | SEL | OPT | NONE] [1] [2] [,ISTATUS=ACTIVE | INACTIVE] [,LOGAPPL=application program name] [1] [,LOGTAB=interpret table name] [1] [,MODETAB=logon mode table name] [1] [,SSCPFM=FSS | USSSCS] [1] [,USSTAB=USS definition table name] [1] [,VPACING=n | 0 | 1] [1]

[1] If any of these operands are specified in both the PU and LU statements, the values used are those in the LU statement.
[2] For the ACF/VTAM Encrypt/Decrypt Feature only.

name

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. The symbolic name provides the minor node name of the logical unit and is required.

LOCADDR=n

is a decimal value that specifies the logical unit's local address at the physical unit. The minimum value of n is 1 and the maximum value is 255.

An LU statement is not required for every possible local address, and LOCADDR values need not be consecutive. However, ACF/VTAM allocates 8 bytes of fixed storage for each skipped address. Unused local addresses smaller than the largest local address at the physical unit are assigned network resources. It is recommended, therefore, that local addresses not be skipped. For example, if only three logical units are contained in the physical unit, use values of 1, 2, and 3.

DLOGMOD=default logmode entry

specifies the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default.

ENCR=REQD | SEL | OPT | NONE

can be specified only if the ACF/VTAM Encrypt/Decrypt Feature is installed. It specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD

specifies that ACF/VTAM must encipher all messages to and from this logical unit. If ENCR=REQD is coded, no sessions can be established with this logical unit unless the host with which it is associated is able to handle cryptographic sessions.

SEL
has no meaning for logical units. If ENCR=SEL is entered, ENCR=OPT is used instead.

OPT
specifies that this logical unit is capable of engaging in cryptographic sessions, but allows the application program to determine whether or not to use cryptography.

NONE
specifies that this logical unit is incapable of engaging in cryptographic sessions.

ISTATUS=ACTIVE | INACTIVE
specifies whether the logical unit is to be activated automatically when the physical unit is activated following the first start of ACF/VTAM or a cold restart of ACF/VTAM (that is, a restart to initial status).

LOGAPPL=application program name
specifies the name of an application program to which the logical unit is to be automatically logged on when the logical unit is activated.

LOGTAB=interpret table name
specifies the name of an interpret table to be used by ACF/VTAM when processing logons originating from the logical unit. The interpret table specified by LOGTAB is used to interpret the APPLID portion of an Initiate Self or character-coded logon command, as described in Chapter 4.

MODETAB=logon mode table name
specifies the logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit. The name specified for the MODETAB operand must be the name of a logon mode table created as described in Chapter 4.

If a logon mode table is not specified for a logical unit by the MODETAB operand on either the PU or the LU statement, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table.

SSCPFM=FSS | USSSCS
specifies whether a logical unit can support character coded messages (SSCPFM=USSSCS) in its communication with the SSCP. If not, only formatted commands (SSCPFM=FSS) are used.

See the publications for each individual device to see if formatted or character coded messages are supported by that device.

USSTAB=USS definition table name
specifies the name of a USS definition table that is created as described in Chapter 4. If USSTAB is not specified, an IBM-supplied USS definition table is searched when character-coded input is received by ACF/VTAM from a logical unit.

VPACING=n | 0 | 1
specifies the way that ACF/VTAM is to pace the flow of data from ACF/VTAM to the logical unit. For information on pacing, see the "Pacing" section later in this chapter.

n
specifies the number of normal-flow messages that ACF/VTAM is to send for a given application program-logical unit session (LU-LU session) before waiting for a pacing response. No further normal-flow messages can be sent to the logical unit until the logical unit is ready to receive more messages.

Specify n as a decimal integer in the range 1 through 255.

0
specifies that no pacing is to be performed for sessions with the logical unit.

If the VPACING operand is omitted, VPACING=1 is assumed for a local SNA major node.

Defining Switched SNA Major Nodes

A switched SNA major node is defined by filing a single VBUILD statement for the major node and separate PU, PATH, and LU statements for each minor node. One VBUILD statement must be included in each member, placed before the first PU statement.

The PU and LU statements define physical units and logical units attached to communications controllers by switched SDLC lines. For dial-out operations, the PATH statement defines the paths to be used to establish connection between the communications controller and the physical unit.

The PU and LU statements used to define a switched SNA major node are very similar to the PU and LU statements used to define a local SNA major node, and to the PU and LU macro instructions used to define an NCP major node. Where possible confusion might arise, the statements used for the switched SNA major nodes are called PU (switched) and LU (switched) statements.

You can define multiple sets of switched SNA devices. This allows the network operator to use the VARY command to activate a selected subset of all the switched SNA devices. However, all major and minor node names known to ACF/VTAM at any one time must be unique. Two major nodes that include the same physical unit or logical unit cannot be active at the same time.

In addition, you can define a physical unit and its associated logical units in such a way that you can use a switched line to back up a nonswitched line. To do this, you must define 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 units and its logical units are active as part of an NCP major node and the nonswitched line fails, the network operator can release the physical unit and its logical units with the VARY REL command and then activate the backup switched line, the physical unit, and its logical units, as part of the SNA major node.

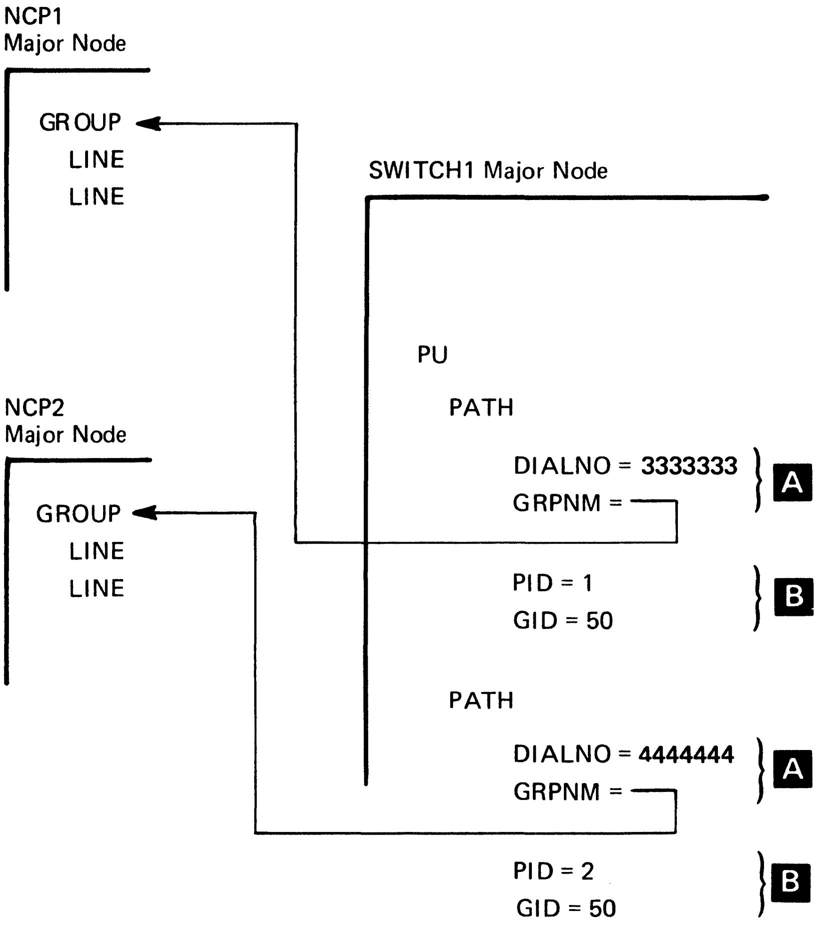
For further information, refer to *ACF/VTAM Network Operating Procedures.*

If contact is to be established with a physical unit by means of a dial-in or dial-out operation, the unit's station identification number is specified in the PU statement. As soon as a dial-in line has been activated and placed in answer mode (during ACF/VTAM startup or by the network operator), the physical unit can dial that line's number and establish contact with ACF/VTAM.

If ACF/VTAM is to establish contact with a physical unit by means of a dial-out operation, PATH statements must be placed after the PU statement to define the paths over which contact can be established. (Dial-out operations occur when an application program issues OPNDST OPTCD=ACQUIRE or OPNDST OPTCD=ACCEPT to establish connection to a logical unit whose switched physical unit is not already connected to ACF/VTAM.)

Each PATH statement tells ACF/VTAM the NCP line group and the dialing digits to be used, as shown in Figure 2-1, part A. If the line group contains more than one dial-out line, each line is tried in succession until contact is established. If contact cannot be established using that line group, ACF/VTAM tries again using the line group identified by the next PATH statement. If the line does not have auto-call capability, the network operator receives a message instructing the operator to manually dial the physical unit. The line groups identified in successive PATH statements need not be part of the same NCP.

Two additional operands of the PATH statement, PID and GID, are used to identify each path and to associate the paths into groups of paths (Figure 2-1, part B). Both operands are used by the network operator. The PID operand identifies the path with respect to its associated physical unit. (In effect, the PID is the equivalent of the PATH statement's label.) By citing the physical unit name and the PID, the network operator can render that specific path usable or not usable.



A These parameters supply the essential information needed to create a dial-out connection.

B These parameters identify the path (PID) and indicate the path group (GID) with which it is associated.

Figure 2-1. Major Elements of a PATH Statement

The GID operand is used to associate the path with any other path or group of paths within the same switched major node. By citing the GID, the network operator can render that path and all similarly-identified paths usable or not usable. The criteria used to assign paths to particular groups are entirely up to the user. One possibility is to group paths according to the type of call indicated by the dialing digits:

Internal (extension) calls

Local calls

Tie-line or WATS (Wide Area Telephone Service) calls

Long distance calls

The VBUILD Statement

Code a VBUILD statement in 80-byte card-image format for each switched SNA major node.

Write the VBUILD statement as follows:

Name	Operation	Operand
name	VBUILD	MAXGRP=n ,MAXNO=n ,TYPE=SWNET [,CONFGDS=ddname] [,CONFGPW=password]

name
one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. Specification of a name is optional.

MAXGRP=n
is the number of unique path groups (GROUP names) that are specified in the GRPNM operand of all PATH statements within the switched major node. The maximum value of n is 32767.

MAXNO=n
is the number of unique telephone numbers that are specified in the DIALNO operand of all PATH statements within the switched major node. The maximum value of n is 32767.

TYPE=SWNET
specifies that the VBUILD statement defines a switched major node to ACF/VTAM. All physical units defined in this major node can be connected only by means of a switched link. This operand is required.

CONFGDS=ddname
is a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure.

Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities" for a description of configuration restart.

CONFGPW=password

specifies the one to eight character password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified but is required by VSAM, VSAM prompts the network operator to provide the correct password when ACF/VTAM attempts to open the data set. This operand can be specified only if the CONFGDS operand is specified.

The PU (Switched) Statement

Code a PU statement in 80-byte card-image format for each physical unit in the switched major node.

Write the PU statement as follows:

Name	Operation	Operand
name	PU	ADDR=station address ,IDBLK=identification block ,IDNUM=identification number [,BATCH=YES | NO] * [,DISCNT=([YES | NO] [,F | NF])] [,DLOGMOD=default logmode entry name] * [,ENCR=REQD | SEL | OPT | NONE] * [,IRETRY=YES | NO] [,ISTATUS=ACTIVE | INACTIVE] [,LOGAPPL=application program name] * [,LOGTAB=interpret table name] * [,MAXDATA=size] [,MAXOUT=n | 1] [,MAXPATH=n | 0] [,MODETAB=logon mode table name] * [,PACING=n | 0 | 1] * [,PASSLIM=n | 1] [,PUTYPE=n | 2] [,SSCPFM=FSS | USSSCS] * [,USSTAB=USS definition table name] * [,VPACING=n | 0 | 2] *

*These operands can be specified in either the PU or the LU statement. The operands are meaningful for logical units, and the operand explanations appear only in the LU statement description. Coding one of these operands in a PU statement is equivalent to coding the operand in each of the associated LU statements. If an operand with a different value is then coded in an LU statement, the value in the LU statement overrides (for that logical unit) the value coded in the PU statement.

name

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. The symbolic name provides the minor node name of the physical unit and is required.

ADDR=station address

is the 8-bit SDLC station address for the physical unit and is required. Specify a hexadecimal address that is not enclosed in quotation marks or apostrophes. This address must be obtained from the person who planned the installation of the device.

IDBLK=identification block

is the 12-bit binary block number assigned by IBM to the specific device and is required. Specify a hexadecimal number that is not enclosed in quotation marks or apostrophes. This number must be obtained from the component description manual for the device or from the person who planned the installation of the device.

The block number occupies bits 16 through 27 of the 48-bit station ID constructed by ACF/VTAM for switched network operation. (See the IDNUM operand below for a description of the station ID.)

IDNUM=identification number

is the 20-bit binary identification number assigned to the station being defined and is required. The IDNUM value is either arbitrarily selected by the user or is obtained from the person who installed the device. Specification of identification numbers is different for various devices; review the appropriate device component description manual to determine the IDNUM for a given device.

IDBLK and IDNUM are used by ACF/VTAM to construct a 48-bit station ID that is used in XID exchange during the dial procedure. This station ID must be unique for each station within the network (not just within the major node). The station ID is structured as follows:

Bits 0-3:	Reserved
Bits 4-7:	PUTYPE
Bits 8-15:	X'00'
Bits 16-27:	IDBLK
Bits 28-47:	IDNUM

DISCNT=([YES | NO][,F | NF])

YES | NO

specifies whether ACF/VTAM is to physically disconnect the physical unit when the last logical unit is disconnected by its application program (that is, when there are no more application program-logical unit sessions).

For a physical unit on a switched link, disconnection means that the dial connection is broken (in effect, the telephone is hung up thus saving telephone charges), and the ACF/VTAM-physical unit session (in SNA terms, the SSCP-PU session) is terminated. Disconnection on a switched link, as contrasted with a leased link, does not involve the deactivation of the physical unit or its logical units. (That is, sessions can be requested with those logical units; such a request causes the physical connection to be reestablished.)

YES

indicates that ACF/VTAM is to automatically disconnect the physical unit as soon as the last logical unit is disconnected by its application program. If any logical units request their own disconnection, ACF/VTAM ignores the part of their disconnection request that indicates whether the physical unit is to be disconnected (that is, the HOLD part of a character-coded logoff command or the LAST-NOLAST part of a field-formatted Terminate Self command is ignored). ACF/VTAM also rejects any attempt made by the physical unit to request its own disconnection (using the Request Discontact command).

NO
indicates that ACF/VTAM is to disconnect the physical unit when one of the following conditions is met:

ACF/VTAM receives a Request Discontact Normal command from the physical unit and all logical units have been disconnected and there are no more application program to logical unit sessions. Existing sessions are allowed to end normally.

ACF/VTAM receives a Request Discontact Immediate command from the physical unit. ACF/VTAM immediately terminates existing application program to logical unit sessions. This command overrides any previous Request Discontact Normal command from the physical unit.

All logical units have been disconnected as a result of a character-coded logoff command for which HOLD=NO was specified (or inserted using a USS definition table), a Terminate Self command for which LAST was specified, or a VARY INACT command. If any logical unit is disconnected by any other means (such as by an application program CLSDST), the physical unit is not disconnected.

F | NF
specifies whether ACF/VTAM is to indicate "final-use" status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES. This operand does not apply when DISCNT=NO is specified, nor does it have any effect on the VARY INACT command. If F is specified or assumed by default, "final-use" status is indicated. If NF is specified, "not-final-use" status is indicated. Each device has its own requirements regarding "final-use" status. To determine whether F or NF should be specified for a given device, consult the appropriate installation publication for the device.

IRETRY=YES | NO
specifies whether the boundary NCP (the NCP to which the switched physical unit will become connected) is to retry a polling operation immediately for the device if an IDLE Detect Timeout follows a polling operation. For more information on this operand, see the *NCP Generation Manual.*

ISTATUS=ACTIVE | INACTIVE
specifies whether the physical unit is to be activated when the switched SNA major node is activated following the first start of ACF/VTAM or a cold restart of ACF/VTAM (that is, a restart to initial status).

MAXDATA=size
specifies the maximum amount of data in bytes, including the transmission header (TH) and request/response header (RH), that the physical unit can receive in one path information unit (PIU). The maximum value is 65535 bytes. For more information on this operand, see the *NCP Generation Manual.*

MAXOUT=n | 1
specifies the maximum number of path information units (PIUs) that the NCP will send to the physical unit before requesting a response from the physical unit. Specify n as a decimal integer from 1 through 7. For more information on this operand, see the *NCP Generation Manual.*

MAXPATH=n | 0
specifies the number of dial-out paths to the physical unit. Specify n as a decimal integer in the range 0 through 256. Zero indicates that only dial-in paths to the physical unit are available.

Refer to the PATH statement for a description of defining a dial-out path to a physical unit.

PASSLIM=n | 1

is the maximum number of contiguous path information units (PIUs) that the NCP will send to the physical unit at one time. Specify n as a decimal integer in the range 1 through the value specified for MAXOUT. For more information on this operand, see the *NCP Generation Manual.*

PUTYPE=n | 2

specifies the physical unit type of the physical unit. The physical unit type depends on the type of cluster controller (such as 3767 or 3791). To determine the physical unit type for a given device, see the component description manual for that device.

The value specified for PUTYPE occupies bits 4 through 7 of the 48-bit station ID constructed by ACF/VTAM for switched network operation. (See the IDNUM operand above for a description of the station ID.)

The PATH Statement

The PATH statement is used to define a dial-out path to a physical unit in a switched SNA major node. As many PATH statements as required, to a maximum of 256, can be specified for each physical unit. In the configuration deck, the PATH statement must immediately follow the PU statement which defines the associated physical unit. ACF/VTAM searches the PATH statements for an available path in the order specified in the configuration deck.

Code the statement in 80-byte card-image format. Write the PATH statement as follows:

Name	Operation	Operand

name	PATH	DIALNO=telephone number,
		GID=n,
		GRPNM=groupname,
		PID=n
		[,REDIAL=n | 3]
		[,USE=YES | NO]

name

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. Specification of a name is optional.

DIALNO=telephone number

specifies (in EBCDIC) the dial characters to be used in initiating a connection with a physical unit over a switched link. A vertical bar (|) can be inserted as many times as required to indicate a dialing pause. The communications controller sends to the modem only the four low-order bits of the digits specified in the sequence. The bit pattern for the end-of-number character is 1100 (hex C); the bit pattern for the separator character is 1101 (hex D). Any EBCDIC character whose four low-order bits equal one of these patterns can be used. For example, you could code DIALNO=(8'5799*), where ' (hex 7D) is the separator character and * (hex 5C) is the end-of-number character. The maximum length of the operand is 32 characters (including vertical bars, the separator character, and the end-of-number character). For more information on this operand, refer to the *NCP Generation Manual.*

GID=n

is an identifier for groupings of paths across all physical units in the switched SNA major node. Specify n as a decimal integer in the range 0 through 255.

Group identifiers can be assigned to let the network operator regulate the use of switched network services. For example, if GID=6 is assigned to all paths in a switched SNA major node that use direct distance dialing, the network operator can make all of the paths usable or not usable with a single command.

GRPNM=groupname

specifies the symbolic name of a GROUP macro instruction in an NCP definition deck that defines a group of SDLC switched links. The line group must have all the characteristics necessary to process the telephone number and must be compatible with the type of physical unit. For more information on this operand, refer to the *NCP Generation Manual.*

PID=n

is an identifier for the path being defined. This identifier is unique for a given physical unit. The operator uses this identifier to change the status of the path. Specify n as a decimal integer in the range 0 through 255.

REDIAL=n | 3

specifies the number of times dialing is to be retried at the NCP before returning a dialing error to ACF/VTAM. The minimum value for n is 0, which indicates that dialing is not to be retried. The maximum value for n is 254.

USE=YES | NO

specifies whether ACF/VTAM is to consider the path initially usable or not usable. This attribute of the path can be modified by the network operator. The effect of USE=YES and USE=NO for a path is similar to the effect of ISTATUS=ACTIVE and ISTATUS=INACTIVE for a minor node.

The LU (Switched) Statement

Code an LU statement in 80-byte card-image format for each logical unit associated with a physical unit within a switched SNA major node. The LU statement must follow the PU statement that defines the physical unit with which the logical unit is associated. As for NCP generation, LU statements must be in ascending order according to the value specified for LOCADDR.

Write the LU statement as follows:

Name	Operation	Operand
name	LU	LOCADDR=n [,BATCH=YES | NO] [1] [,DLOGMOD=default logmode entry] [1] [,ENCR=REQD | SEL | OPT | NONE] [1, 2] [,ISTATUS=ACTIVE | INACTIVE] [,LOGAPPL=application program name] [1] [,LOGTAB=interpret table name] [1] [,MODETAB=logon mode table name] [1] [,PACING=n | 0 | 1] [1] [,SSCPFM=FSS | USSSCS] [1] [,USSTAB=USS definition table name] [1] [,VPACING=n | 0 | 2] [1]

[1] If any of these operands are specified in both the PU and LU statements, the values used are those in the LU statement.

[2] For the ACF/VTAM Encrypt/Decrypt Feature only.

name

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. The symbolic name provides the name of the logical unit and is required.

LOCADDR=n

is a decimal value that specifies the logical unit's local address at the physical unit.

The range of valid local addresses depends on the PUTYPE specified for the physical unit with which the logical unit is associated. PUTYPE=1 allows local addresses from 0 to 63; PUTYPE=2 allows local addresses from 1 to 255. To determine the PUTYPE and the limits on local addresses for a particular physical unit, see the component description manual for the device.

An LU statement is not required for every possible local address, and LOCADDR values need not be consecutive. However, ACF/VTAM allocates 8 bytes of fixed storage for each skipped address. Unused local addresses smaller than the largest local address at a station are assigned network resources. It is recommended, therefore, that local addresses not be skipped. For example, if only three logical units are contained in the physical unit, use values of 1, 2, and 3.

BATCH=YES | NO

specifies the processing priority that the NCP is to use for the logical unit. BATCH=NO indicates a high priority (suitable for interactive applications); BATCH=YES indicates a low priority.

DLOGMOD=default logmode entry

specifies the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default.

ENCR=REQD | SEL | OPT | NONE

can be specified only if the ACF/VTAM Encrypt/Decrypt Feature is installed. It specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD

specifies that ACF/VTAM must encipher all messages to and from this logical unit. If ENCR=REQD is coded, no sessions can be established with this logical unit unless the host with which it is associated is able to handle cryptographic sessions.

SEL

has no meaning for logical units. If ENCR=SEL is entered, ENCR=OPT is used instead.

OPT

specifies that this logical unit is capable of engaging in cryptographic sessions, but allows the application program to determine whether or not to use cryptography.

NONE

specifies that this logical unit is incapable of engaging in cryptographic sessions.

ISTATUS=ACTIVE | INACTIVE

specifies whether the logical unit is to be activated automatically when the physical unit is activated.

LOGAPPL=application program name

specifies the name of an application program to which the logical unit is to be automatically logged on when the logical unit is activated. For logical units that are accessible only by dial-in paths, the application program receives control after the dial-in connection is made. For dialin/dialout or dialout paths, control is given immediately.

LOGTAB=interpret table name

specifies the name of an interpret table to be used by ACF/VTAM when processing logons originating from the logical unit. See Chapter 4 for a description of how an interpret table is defined.

MODETAB=logon mode table name

specifies the logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit. The name specified for the MODETAB operand must be the same name used for a logon mode table that is created as described in Chapter 4.

If a logon mode table is not specified for a logical unit by the MODETAB operand in either the PU or the LU statement, an IBM-supplied logon mode table is used. The user can modify or replace the IBM-supplied logon mode table.

PACING=n | 0 | 1

specifies the way that pacing is to be handled between the logical unit and the NCP to which the logical unit is connected. (In contrast, VPACING involves pacing between ACF/VTAM and the NCP.) For more information on pacing, see the "Pacing" section later in this chapter.

n

specifies the number of normal-flow messages that the NCP is to send to the logical unit before waiting for a pacing response.

Specify n as a decimal integer in the range 1 through 255, referring to the *NCP Generation Manual* for information on how to select the proper value.

0

specifies that no pacing is to be performed for sessions with the logical unit.

If the PACING operand is omitted, PACING=1 is assumed by default.

SSCPFM=FSS

Code SSCPFM=FSS or allow it to be assumed by default.

USSTAB=USS definition table name

specifies the name of a USS definition table that is defined as described in Chapter 4.

If USSTAB is not specified, the ACF/VTAM-supplied USS definition table is searched when input is received by ACF/VTAM from a logical unit.

VPACING=n | 0 | 2

specifies the way that ACF/VTAM is to pace the flow of data from ACF/VTAM to the NCP to which the logical unit is connected. For more information on pacing, see the "Pacing" section later in this chapter.

n
specifies the number of normal-flow messages that ACF/VTAM is to send for a given application program-logical unit session (LU-LU session) before waiting for a pacing response. No further requests can be sent to the NCP until it replies with a pacing response to ACF/VTAM to indicate that it is ready to receive more messages.

Specify n as a decimal integer in the range 1 through 255, and greater than the value of n used with the corresponding PACING operand.

0
specifies that no pacing is to be performed for sessions with the logical unit.

If the VPACING operand is omitted, VPACING=2 is assumed in a switched SNA major node.

Defining Network Control Program (NCP) Major Nodes

Each NCP for a communications controller and its attached terminals must be defined to ACF/VTAM. Also, either (1) the same NCP can be used for another communications controller (although not concurrently) if the communications controller and its attached terminal configuration are identical, or (2) one communications controller can have more than one NCP to define different terminal configurations for the communications controller. To generate an NCP for use in an ACF/VTAM network and to define that NCP to ACF/VTAM involves following the detailed instructions given in the *NCP Generation Manual* and using them in conjunction with these considerations and requirements:

NCP line control

Initial test routine for local communications controller

Identification verification for BSC and TWX (teletypewriter exchange service) devices

Pacing

Device dependencies of terminals attached to communications controllers

NCP generation procedure

The NCP Line Control

ACF/VTAM supports NCPs in network control mode and NCPs that use the partitioned emulation programming (PEP) extension. ACF/VTAM does not support emulation mode.

To allow the NCP to operate communications lines in network control mode, the appropriate TYPGEN operand (TYPGEN=NCP, TYPGEN=NCP-LR, TYPGEN=NCP-R, TYPGEN=PEP, or TYPGEN=PEP-LR) must be specified in the BUILD macro instruction.

If the type of NCP generation is for only network control mode (TYPGEN=NCP, TYPGEN=NCP-LR, or TYPGEN=NCP-R), then any TYPE or USE operands that are coded on GROUP or LINE macro instructions for this NCP must specify network control mode (USE=NCP and TYPE=NCP). If you omit these two operands, the correct values are assumed by default. No NCP generation macro instructions or operands can be coded that apply to emulation mode.

If the type of NCP generation is for use with PEP (TYPGEN=PEP or TYPGEN=PEP-LR), ACF/VTAM can only support the generated NCP for the lines in network control mode, although NCP operands and generation macro instructions can be coded that apply to emulation mode. This occurs if the TYPE=NCP operand is specified in the GROUP and LINE macro instructions or if TYPE=PEP is specified and the line is in network control mode. A PEP line can be placed in network control mode with a network operator command or when the NCP is activated and the USE=NCP operand has been specified in that GROUP or LINE macro instruction.

ACF/VTAM manages the assignment of PEP lines by changing line assignments 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.)

Caution should be used when activating lines because a request to ACF/VTAM to activate a line proceeds even if that line is currently being used by another teleprocessing access method through emulation mode.

Initial Test Routine

The NCP initial test routine is a diagnostic routine that is executed in the communications controller before the NCP is loaded. This routine tests the communications controller for certain hardware malfunctions. If it does not detect any malfunctions, then ACF/VTAM loads the NCP. If it does detect a malfunction, the routine stops, the HARD STOP light on the communications controller is turned on, and the NCP is not loaded (an IBM customer engineer should be called).

Remote communications controllers perform initial testing automatically when they are started (power is turned on). However, initial testing is optional for local communications controllers. For ACF/VTAM to initiate initial testing in a local communications controller, the INITEST=YES operand must be coded in the PCCU macro instruction for the local NCP. Also a DD statement with the ddname INITEST must be added to the cataloged procedure for starting ACF/VTAM.

ID Verification For BSC And TWX Terminals

To help provide telecommunication security in the network, the user provides identification (ID) verification for BSC and TWX terminals on switched lines. This type of ID verification allows the user to identify specific terminals and application programs and to control their access to privileged or sensitive data and resources.

To provide this verification authority, the terminals and their associated ID sequences are specified in the VIDLIST or IDLIST macro instructions. Then, ID sequences for dial-up BSC and TWX terminals in the network can be verified by:

ACF/VTAM, with the VIDLIST macro instruction

The NCP, with the IDLIST macro instruction

The application program, using the UTERM operand in the TERMINAL macro instruction

ACF/VTAM and the NCP (if ID verification authority is to be distributed)

ID verification cannot be performed either by ACF/VTAM or an NCP for a TWX terminal on a line supported by the multi-terminal access (MTA) facility. For more information on this topic, see *NCP Generation Manual.*

ACF/VTAM Verification

By coding the following NCP macro instructions and operands, for example, you can have ACF/VTAM perform all ID verification in the host computer.

```
.
.
.
symbol VIDLIST VIDSEQ= . . .
.
.
.
LINE VIDSEQ=symbol [1]
TERMINAL CTERM=YES,IDSEQ=PASS [2]
```

NCP Verification

You can also code the following NCP macro instructions and operands, for example, to have an NCP perform all ID verification in a communications controller.

```
.
.
.
symbol IDLIST IDSEQ= . . . ,NOMATCH=STOP . . . [3]
.
.
.
LINE
TERMINAL CTERM=YES,IDSEQ=symbol
```

Application Program Verification

The UTERM operand (in the TERMINAL macro instruction) allows ACF/VTAM to pass unidentified terminals (those with ID sequences that ACF/VTAM or an NCP could not verify or those without ID sequences) to an application program. Any further ID verification must be done by the application program's logon-interpret routine.

ACF/VTAM And NCP Verification

You can distribute verification authority between the ACF/VTAM in the host computer and an NCP in a communications controller. The following NCP macro instructions and operands can be coded to combine ACF/VTAM and NCP ID verification.

[1] symbol must correspond to symbol for the VIDLIST macro instruction specifying the ID sequence for terminals on this line.

[2] PASS informs the NCP that it is to pass all ID sequences it receives to the host computer.

[3] STOP specifies that the NCP is to break a line connection if it does not recognize an ID sequence. The NCP checks the ID sequence only if the terminal calls the communications controller. When the NCP calls the terminal, it does not check ID sequences.

```
    .
    .
    .
symbol1 IDLIST[1] IDSEQ=. . .,NOMATCH=PASS. . . [2]
symbol2 VIDLIST[1] VIDSEQ=. . . [3]
    .
    .
GROUP
LINE VIDSEQ=symbol2
TERMINAL CTERM=YES,IDSEQ=symbol1
```

See the topic "Coding ACF/VTAM-Only Definition Statements" for the coding requirements for the VIDLIST macro instruction and for the VIDSEQ and UTERM operands. See the *NCP Generation Manual* for the coding requirements for the other NCP macro instructions and operands mentioned in the preceding examples in this topic and for more information about ID verification and ID sequences for BSC and TWX terminals on switched lines.

Device Considerations

Many NCP macro instructions (especially GROUP, LINE, CLUSTER, LU, PU, TERMINAL, VTERM, and COMP) contain device-dependent operands that affect both NCP generation and the ACF/VTAM application programs. Therefore, before the coding NCP generation macro instructions that are to support an ACF/VTAM network:

Review ACF/VTAM and NCP device dependencies in the appropriate IBM publications for each device being defined for the network (the *NCP Generation Manual, ACF/VTAM Concepts and Planning, ACF/VTAM Macro Language Reference,* and the appropriate component publications).

Have a copy of the NCP generation source code available for reference by the persons writing application programs.

Review the information in Appendix A in this publication for devices that use the basic mode of data transfer and for MTA line considerations

Review the device dependencies for each device that uses the record mode of data transfer, in the installatiun publication for each IBM data communication subsystem

Pacing

Pacing is the SNA facility used to control the number of normal-flow requests allowed to flow at one time between the sender and the receiver in a session. The receiver can control the rate at which the sender sends requests, and thus can better control the use of its own resources. For example, pacing helps the receiver control the allocation of its buffer resources. Pacing applies independently to requests flowing from the primary to the secondary and from the secondary to the primary end of a session.

[1] These macro instructions must be placed in the NCP generation procedure anywhere following the SYSCNTRL macro instruction, but preceding the first GROUP macro instruction.

[2] Specifies that the NCP is to pass to the host computer an ID sequence does not recognize as valid in the IDLIST macro instruction. This in turn causes ACF/VTAM to check the ID sequence against the list specified in the VIDLST macro instruction to associate the ID sequence with a terminal name and, possibly, a waiting application program.

[3] This operand must contain (at a minimum) all the terminals (ID sequence and associated terminal name of each terminal) that are identified in the IDSEQ operand of the IDLIST macro instruction.

In one stage pacing, the pacing occurs directly between the two logical units involved in the session. For example, one stage pacing is used when two ACF/VTAM application programs are in session. In two stage pacing, the pacing occurs between one of the logical units and a boundary function and then between the boundary function and the other logical unit in the session (see Figure 2-2). (A boundary function is part of a host computer or 3704 or 3705 communications controller that supports a cluster controller attached to the host computer or communications controller.) An example of two stage pacing is when an application program is in session with a logical unit associated with a cluster controller attached to a 3705. In this case, the NCP in the 3705 provides the boundary function for the logical unit.

The number of pacing stages can be the same for both directions (primary to secondary and secondary to primary) in a session or it can be different. In the previously mentioned example of two application programs being in session, one stage pacing is used for both directions. For the session between the application program and a logical unit associated with a cluster controller attached to a 3705, two stage pacing is used for the primary to secondary direction and one stage pacing is used for the secondary to primary direction.

When pacing is in effect, the sender for a given pacing stage in a session is allowed to send a limited number of normal-flow requests (this number is specified by the PACING or VPACING operand, or by a logon mode table, as described later in this section). The sender is not allowed to send any more such requests until the receiver indicates that it is ready to receive more by returning a pacing response to the sender. The sender can then send more requests, up to the previously mentioned limit. If the receiver sends the pacing response early (that is, before having received the maximum number of requests), the sender completes sending the first group of requests and can then send another group of requests, as before. The pacing of requests flowing toward a given logical unit (including an application program) is called inbound pacing for that logical unit. Similarly, the pacing of requests flowing away from a logical unit is called outbound pacing for that logical unit.

A separate pacing value is specified for each of the potential pacing stages in a session. These four numbers are part of the session parameters defined in *ACF/VTAM Macro Language Reference.* The four numbers are carried in the SNA Bind command from the primary logical unit to the boundary function (if one exists for this session) and then to the secondary logical unit. In this way, all the participants in the session are informed of the pacing values to be used. The four pacing numbers are the Primary Send (PS) count, the Secondary Receive (SR) count, the Secondary Send (SS) count, and the Primary Receive (PR) count. (Refer to Figure 2-2.)

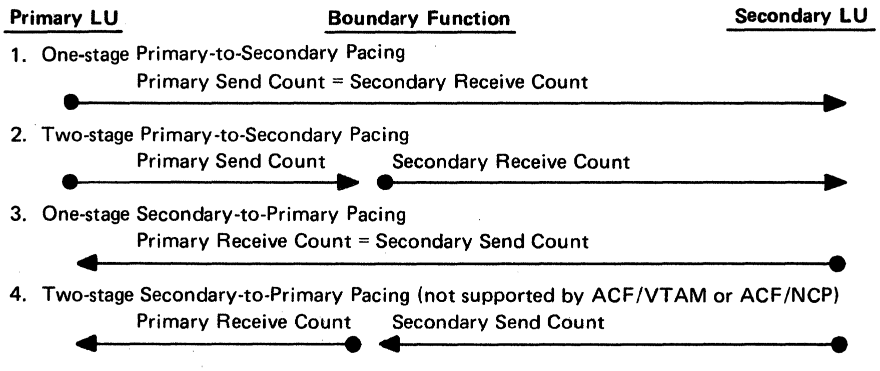

Primary LU
Boundary Function
Secondary LU
1. One-stage Primary-to-Secondary Pacing
Primary Send Count = Secondary Receive Count
2. Two-stage Primary-to-Secondary Pacing
Primary Send Count
Secondary Receive Count
3. One-stage Secondary-to-Primary Pacing
Primary Receive Count = Secondary Send Count
4. Two-stage Secondary-to-Primary Pacing (not supported by ACF/VTAM or ACF/NCP)
Primary Receive Count
Secondary Send Count

Figure 2-2. Pacing Counts Used to Control One and Two Stage Pacing

The PS and SR counts control pacing from the primary to the secondary end of the session, with PS being the count (the limiting number mentioned previously) used to control the flow of requests from the primary to the boundary function and SR being the count controlling the flow from the boundary function to the secondary. For two stage pacing, the PS and SR counts can be different; for one stage pacing they are the same. The SS and PR counts are used in a similar manner; they control the flow from the secondary to the primary end of the session. A pacing count of zero means that that particular pacing stage is unpaced; an unlimited number of requests can be sent. A pacing count of zero, although causing less total network traffic to flow because no pacing responses are sent, should still be used with caution because the receiver might not be able to control the rate of input to it.

Definition Of Pacing Values

Generally, the various pacing values can be defined in either of two ways. They can be specified on the definition statement that defines a logical unit to ACF/VTAM (for example, the PACING and VPACING operands on a LU statement in a switched SNA major node) or they can be specified in a logon mode table entry that is derived from the logon that requests the session.

Figure 2-3 summarizes how the PACING and VPACING operands relate to the various pacing stages for current ACF/VTAM and ACF/NCP operations.

ACF/VTAM Definition Statement[1]	Meaning of VPACING	Meaning of PACING	What the Operands Control
APPL	PR and SS	Cannot be specified	One stage, secondary to primary, if the application program is the primary.
APPL	PS and SR	Cannot be specified	One stage, primary to secondary, if the application program is the secondary.
LU in local SNA major node	PS and SR	Cannot be specified	One stage, primary to secondary.
LU in switched SNA major node	PS	SR	Two stage, primary to secondary.
LU in NCP major node	PS	SR	Two stage, primary to secondary.
LOCAL in local non-SNA major node	Cannot be specified	Cannot be specified	One stage, primary to secondary. (Can be specified by logon mode table[2].)
TERMINAL in NCP major node	Cannot be specified	Cannot be specified	One stage, primary to secondary. (Can be specified by logon mode table[2].)

Notes:

1. The resource specified can act as one end of a session with an ACF/VTAM application program which uses record mode.
2. Three operands on the MODEENT macro instruction, which defines a logon mode table entry, can specify pacing counts: PSNDPAC specifies the PS count, SRCVPAC specifies the SR count, and SSNDPAC specifies the SS count.

Figure 2-3. Specification of Pacing Counts During Network Definition

If an application program acting as a primary does an INQUIRE for the session parameters associated with a pending logon, the following information about pacing counts is made available:

The Primary Receive count equals the VPACING value on the APPL definition statement, unless the logon mode table entry SSNDPAC operand is set to zero. If SSNDPAC is set to zero, the Primary Receive count is also set to zero. (This can be used to suppress secondary to primary pacing.)

The Secondary Send count equals the Primary Receive count because only one stage primary to secondary pacing is supported.

The Secondary Receive count equals the SRCVPAC value from the logon mode table entry, unless that value is zero. If SRCVPAC is zero, the appropriate PACING or VPACING value of the secondary application program or logical unit is supplied. See Figure 2-3 to determine how the SR count is specified for a specific secondary.

The Primary Send count equals the PSNDPAC value from the logon mode table entry, unless that value is zero. If PSNDPAC is zero, the VPACING value associated with the secondary application program or logical unit is supplied. See Figure 2-3 to determine how the PS count is specified for a specific secondary.

The primary application program can override the primary receive value for a specific session. It does this by specifying a nonzero value in the primary receive field of the session parameters it specifies at OPNDST. If a zero value is specified, the primary receive value present in the session parameters in the pending logon are used. Only values less than or equal to the value in the pending logon should be specified by the application program.

In summary, the pacing counts specified by the logical unit definition statements apply to all sessions with a logical unit. These counts can be overridden for a specific session by the counts in the logon mode table entry used for that session (if the counts in the table are not set to zero). Finally, before actually establishing the session, an application program acting as a primary can respecify the primary receive count before it is sent in the Bind command.

NCP Generation Procedure For ACF/VTAM

To define an NCP and the terminals it is to control in an ACF/VTAM network, the NCP generation procedure must specify the capabilities of the NCP, the interface between the NCP and ACF/VTAM, and the network configuration. You supply this required information to ACF/VTAM and the NCP by using NCP generation macro instructions.

Additional information is provided to ACF/VTAM by coding ACF/VTAM-only definition statements and ACF/VTAM-only operands in NCP macro instructions. The ACF/VTAM-only definition statements and operands must appear in the NCP generation procedure that defines this NCP to ACF/VTAM, even though they provide no information to the NCP. (ACF/VTAM uses them as input to its initialization process.)

Listed below are all the NCP macro instructions in a correct coding sequence for NCP generation, along with an indication of whether they apply to ACF/VTAM, NCP, or both:

PCCU	(ACF/VTAM-only)
BUILD	(Both ACF/VTAM and NCP)
SYSCNTRL	(Both ACF/VTAM and NCP)
HOST	(Both ACF/VTAM and NCP)
CSB	(NCP-only)
IDLIST	(NCP-only)

LUPOOL	(Both ACF/VTAM and NCP)
MTALCST	(Both, no ACF/VTAM Requirements)
MTALIST	(Both, no ACF/VTAM Requirements)
MTAPOLL	(NCP-only)
MTATABL	(Both, no ACF/VTAM Requirements)
DIALSET	(Both, no ACF/VTAM Requirements)
PATH	(NCP-only)
SDLCST	(NCP-only)
VIDLIST	(ACF/VTAM-only)

GROUP	(Both ACF/VTAM and NCP)
LINE	(Both ACF/VTAM and NCP)
SERVICE	(Both, no ACF/VTAM Requirements)
CLUSTER	(Both ACF/VTAM and NCP)
COMP	(Both ACF/VTAM and NCP)
TERMINAL	(Both ACF/VTAM and NCP)
VTERM	(ACF/VTAM-only)

GROUP	(Both ACF/VTAM and NCP)
LINE	(Both ACF/VTAM and NCP)

SERVICE	(Both, no ACF/VTAM Requirements)
PU	(Both ACF/VTAM and NCP)
LU	(Both ACF/VTAM and NCP)

STARTBH	(Both, no ACF/VTAM Requirements)
ENDBH	(NCP-only)
DATETIME	(NCP-only)
EDIT	(NCP-only)
UBHR	(NCP-only)
BHSET	(NCP-only)
SPAFPT3	(NCP-only)
GENEND	(Both, no ACF/VTAM Requirements)

If a macro instruction is marked 'ACF/VTAM-only', you will find a complete description of the macro instruction in this book.

If a macro instruction is marked 'Both ACF/VTAM and NCP', it indicates that this macro instruction has ACF/VTAM-only operands or restrictions. You will find a description of the ACF/VTAM operands and restrictions in this book, while the description of the macro instruction itself will be found in the *NCP Generation Manual.*

If a macro instruction is marked 'Both, No ACF/VTAM Requirements', it indicates that this macro instruction is used by both ACF/VTAM and the NCP, but it has no special ACF/VTAM requirements and should be coded as described in *NCP Generation Manual.*

If a macro instruction is marked 'NCP-only', you will find a complete description of the macro instruction in the *NCP Generation Manual.*

Coding ACF/VTAM-Only Definition Statements

ACF/VTAM requires information that is contained in the NCP generation macro instructions. Most of the information required by ACF/VTAM is also required by the NCP, although some additional information is required only by ACF/VTAM. The

ACF/VTAM-only definition statements convey no information to the network control program generation assembly process. If applicable, however, they must appear in the NCP generation procedure that defines this NCP to ACF/VTAM. (ACF/VTAM uses them as input to its initialization process.)

The macro instruction assembly step of the NCP generation process permits each ACF/VTAM-only definition statement to appear in its proper sequence. However, the assembly process does not check these macro instructions for proper sequence, syntax, or verify that any related operands are present or absent. (ACF/VTAM does this during its initialization or activation processing.)

The PCCU Macro Instruction

The PCCU (programmed communications control unit) macro instruction, a required ACF/VTAM-only definition statement, identifies the communications controller into which a specific NCP is loaded. This macro instruction defines the ACF/VTAM functions that are to be provided for this specific NCP. Code at least one PCCU macro instruction for each NCP to be defined to ACF/VTAM.

All PCCU macro instructions must occur at the very beginning of the NCP generation deck, preceding all other macro instructions.

The format of the PCCU macro instruction is:

```
Name       Operation     Operand

symbol     PCCU          {CUADDR=local address | RNAME=rname}
                         [,AUTODMP=YES | NO]
                         [,AUTOIPL=YES | NO]
                         [,AUTOSYN=YES | NO]
                         [,BACKUP=YES | NO]
                         [,CONFGDS=ddname]
                         [,CONFGPW=password]
                         [,DUMPDS=dumpname]
                         [,INITEST=YES | NO]
                         [,MAXDATA=size]
                         [,OWNER=ownername]
                         [,SUBAREA=n]
```

symbol
is one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character. Specification of symbol is optional.

CUADDR=local address or RNAME=remote name

CUADDR=local address
local address is three hexadecimal digits that identify the channel unit address of the local communications controller in which this NCP is resident.

CUADDR indicates that this PCCU macro instruction is for an NCP that is to control either a local communications controller or a local communications controller to which a remote communications controller is attached (TYPGEN=NCP, TYPGEN=NCP-LR, TYPGEN=PEP, or TYPGEN=PEP-LR operand is specified in the BUILD macro instruction).

If CUADDR is not specified, the network operator must provide it as an operand of the VARY command when activating a local communications controller.

The VARY command also allows the network operator to override the address specified in the CUADDR operand in the PCCU macro instruction and to load the NCP into another communications controller. However, the two communications controllers and the terminal networks attached to them must be identical.

Do not specify CUADDR when the PCCU macro instruction is for an NCP that is to be loaded into a remote communications controller (TYPGEN=NCP-R operand is specified in the BUILD macro instruction).

If the channel adapter on the communications controller has a manual two-channel switch feature (#8002), local address is the primary address for the communications controller specified in the ADDRESS operand of the IODEVICE system generation macro instruction. (See the "Configuration Restart" section of Chapter 6 for an explanation of switching to a backup computer.)

RNAME=rname

must be identical to the symbol specified on the type 4 PU macro instruction for this NCP in the local communications controller. (The PU macro instruction specifies to the local NCP the SDLC link on which this remote communications controller is attached.)

RNAME applies only to remote NCPs. ACF/VTAM ignores RNAME if it is coded for a PCCU instruction that is for a local NCP or for a local NCP to which a remote NCP is attached.

If RNAME is not specified, it must be provided by the network operator as an operand in the VARY command when activating a remote communications controller.

AUTODMP=YES | NO

indicates whether, after a failure either in the NCP or the communications controller, a dump of storage in the communications controller is to be taken without network operator intervention. (This occurs before ACF/VTAM reloads another copy of the same NCP into the communications controller and restarts it.) When AUTODMP=NO, the network operator is prompted to specify whether a dump is to be taken.

This operand is valid only when the DUMPDS operand is also coded in this PCCU macro instruction. Otherwise no NCP dump can be taken.

In an NCP for a multiple-channel-attached communications controller, only one PCCU macro instruction can have AUTODMP=YES specified on it.

AUTOIPL=YES | NO

indicates whether, after an unrecoverable failure either in the NCP or the communications controller (or after a dump is taken), ACF/VTAM is to load another copy of the NCP into the communications controller and restart it.

After successfully reloading the communications controller, ACF/VTAM initiates its configuration restart facility. Configuration restart attempts to reinstate the status of the NCP's network as it existed at the time of failure. This includes reactivating the communications controller itself, reactivating links that were active at the time of failure, and restarting any remote communications controllers or SDLC cluster controllers (with their associated logical units) attached to the reloaded communications controller.

In an NCP for a multiple-channel-attached communications controller, only one PCCU macro instruction can have AUTOIPL=YES specified on it.

AUTOSYN=YES | NO

When activated, the communications controller advises ACF/VTAM of the name of the NCP loaded in the controller.

If AUTOSYN=YES is specified and the name returned by the controller matches that which ACF/VTAM expects, ACF/VTAM automatically synchronizes itself with the NCP without operator intervention or reloading the controller.

If AUTOSYN=NO is specified and the name returned by the controller matches that which ACF/VTAM expects, the operator is queried to determine whether the NCP already loaded is to be used or the NCP is to be refreshed with a new copy.

After properly reacting to the operator's direction, ACF/VTAM synchronization processing ensues.

If the name returned does not match, the controller is reloaded automatically, without operator intervention.

BACKUP=YES | NO

The BACKUP operand is used for NCPs in multiple-channel-attached communications controllers. This operand applies only to local communications controller, and is ignored when specified for a remote communications controller. If the OWNER operand is not also specified, the BACKUP operand has no meaning and is ignored.

If BACKUP=YES is specified, all resources with an OWNER name different from the one specified on this macro instruction are held in reserve. The host ACF/VTAM cannot use these resources until a VARY ACQ command is issued for this NCP.

The BACKUP operand has no meaning if the OWNER operand is not specified.

CONFGDS=ddname

is a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure.

Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities," for a description of configuration restart.

CONFGPW=password

specifies the one to eight character password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified, but is required by VSAM, VSAM prompts the network operator to provide the correct password when ACF/VTAM attempts to open the data set.

DUMPDS=dumpname

indicates the data definition name of a DD statement in the cataloged procedure for starting ACF/VTAM. This DD statement defines the data set that is to contain the data from a storage dump of a communications controller.

This operand must be coded:

If AUTODMP=YES is also specified in this PCCU macro instruction.

If the network operator is to be given the ability to dump an NCP (when prompted by ACF/VTAM) after a failure in either the communications controller or NCP.

(This occurs prior to reinitializing and restarting the failing communications controller.)

If the network operator is to be given the ability to request a dump for the communications controller (using the MODIFY command).

To format and print the data in the NCP dump data set, use the NCP independent dump utility program, as described in *NCP Generation Manual.*

Before being formatted and printed, the data in an existing NCP dump data set can be overwritten by the next requested NCP dump. To avoid this, consider either having a separate NCP dump data set for each communications controller in the ACF/VTAM system, or at least allocating one NCP dump data set for dumping local communications controllers and another one for dumping remote communications controllers.

INITEST=YES | NO

indicates whether ACF/VTAM is to load a diagnostic routine for a local communications controller (the initial test routine). This routine tests the communications controller for any machine malfunctions (before ACF/VTAM loads the NCP into the communications controller). If INITEST=YES is coded, a DD statement with the data definition name INITEST must have been placed in the cataloged procedure for starting ACF/VTAM. This DD statement enables ACF/VTAM to locate the initial test routine.

If INITEST=YES is coded, ACF/VTAM loads the NCP every time the communications controller is activated.

INITEST applies only to local communications controllers. If INITEST=YES is coded for a NCP that is to control a remote communications controller, ACF/VTAM ignores the operand.

MAXDATA=size

specifies the maximum amount of data in bytes including the transmission header (TH) and the request/response header (RH) that the NCP can receive in one segment or path information unit (PIU). The MAXDATA size should, if possible, be equal to the size of the largest PIU the network will handle, but should not exceed the size in bytes of the IOBUF buffer pool, nor should it exceed the product of the MAXBFRU and UNITSZ values for the NCP. The maximum size is 65535 bytes.

If this operand is omitted, ACF/VTAM sends in one PIU as much data (up to 65535 bytes) as has been passed to it by the application program; if the NCP has insufficient buffers to handle the data, the NCP enters slowdown mode.

This operand applies only to the PCCU macro instruction of a local NCP. However, the MAXDATA value should also not exceed the capacities of any remote NCPs or of any cross-domain NCPs. An SNA path error can result if the MAXDATA value exceeds the capacities of the remote NCPs.

OWNER=ownername

specifies a name (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) that is used to associate a host ACF/VTAM with the resources that it controls. This operand is required for NCPs in multiple-channel-attached communications controller. This operand applies only to local communications controllers and is ignored when specified for a remote communications controller.

SUBAREA=n

applies only to NCPs. It specifies the subarea number of the host computer in which the host ACF/VTAM named by the OWNER operand resides. This number should be the number assigned to the HOSTSA operand when ACF/VTAM was started. ACF/VTAM uses its own subarea number and examines each PCCU macro instruction until it finds a matching number. It then uses the associated OWNER name to locate the resources (defined later in the deck) that it controls. Each SUBAREA value for subareas to be used in cross-domain communications must be unique throughout the whole network.

VIDLIST Macro Instruction (For Basic Mode)

VIDLIST (ACF/VTAM identification list) is an optional, ACF/VTAM-only definition statement that defines a list of identification (ID) sequences. ACF/VTAM compares these ID sequences with ID sequences that are transmitted from BSC terminals or teletypewriter exchange service (TWX) terminals when calling a communications controller over a switched line operated in network control mode. A separate VIDLIST macro instruction is coded for each group of lines with similar physical characteristics or for each line with BSC or TWX terminals.

It is recommended that resources in different domains not be included under the same VIDLIST macro instruction.

The VIDLIST macro instruction can be used with the IDLIST macro instruction to distribute ID verification between ACF/VTAM and an NCP. For a coding example of this method, see the topic in this chapter "ID Verification for BSC and TWX Terminals."

Any ID sequence list defined in the VIDLIST macro instruction can be named in the VIDSEQ operand in the GROUP or LINE macro instructions.

ID verification cannot be performed by either ACF/VTAM or an NCP for a TWX terminal on a line supported by the multiple terminal access (MTA) facility. For information about the MTA facility, see *NCP Generation Manual.*

VIDLIST can be placed anywhere following the SYSCNTRL macro instruction, but it must precede the first GROUP macro instruction.

The format of the VIDLIST macro instruction is:

Name	Operation	Operand
symbol	VIDLIST	VIDSEQ=((chars,termname)[,(chars,termname) . . .])

symbol

is one to eight alphanumeric characters beginning with an alphabetic character other than a $ character.

This symbol is required unless the VIDSEQ operand is being continued from an immediately preceding VIDLIST macro instruction.

VIDSEQ=((chars,termname)[,(chars,termname) . . .])

indicates the ID sequence and associated terminal name of each terminal that is permitted to call into ACF/VTAM for ID verification.

A maximum of 255 characters can be coded in the VIDSEQ operand (including the beginning and ending parentheses and all commas). This limit applies regardless of the number of entries (ID sequence and associated terminal name are one entry) coded within the operand. To specify more than 255 characters, code one or more additional VIDLIST macro instructions (omitting the symbol field in each) and place them directly following the first VIDLIST macro instruction, and code the remaining characters in the VIDSEQ operands of the additional VIDLIST macro instructions.

chars

indicates one ID sequence for ACF/VTAM to recognize as valid. Anywhere from 1 to 40 hexadecimal digits (equivalent to 20 EBCDIC characters) can be specified for each ID sequence.

All EBCDIC characters except the data link control characters (such as the hexadecimal equivalents for EOT, ETB, ETX, and STX) are valid.

Do not specify any EOT (end of transmission), ENQ (enquiry), or ACK (acknowledge) characters in the ID sequence for a TWX terminal. This is because the NCP recognizes these characters as control characters and deletes them as it receives the message sequence into one of its buffers.

termname

indicates the symbolic name (one to eight alphanumeric characters, which must not begin with a $ character) that identifies the terminal with which the ID sequence (coded as chars) is to be associated. The termname must be identical to the symbol specified in the TERMINAL macro instruction that represents this terminal.

Do not specify termname as either the name of a TERMINAL macro instruction in which the CTERM=YES operand is specified, or as the name specified in the UTERM operand in a TERMINAL macro instruction.

If the name (symbol) of this VIDLIST macro instruction is to be specified in the VIDSEQ operand in a GROUP or LINE macro instruction, all the terminals represented by termname must be using the same line or group of lines.

VTERM Macro Instruction (For Basic Mode)

VTERM is an optional, ACF/VTAM-only definition statement that provides automatic logon capability for specific types of dial-up, start-stop terminals that use the MTA facility. The multiple-terminal-access (MTA) facility is an NCP option that allows the NCP to communicate in network control mode with a variety of different start-stop terminals over the same network of switched lines. For a complete description of the MTA facility and the terminals it supports, see the *NCP Generation Manual.*

The UTERM operand must be coded if a VTERM macro instruction is not coded for each different type of dial-up terminal that can be connected to a specific line supported by MTA. The UTERM operand of the TERMINAL macro instruction defines those remaining types of dial-up terminals. (The UTERM operand manages all line control types defined in the MTALIST macro instruction, but not defined in the VTERM macro instruction.)

Place this macro instruction directly following the TERMINAL macro instruction (with the CTERM=YES operand specified) that represents the MTA terminal. Also, this TERMINAL macro instruction must follow a LINE macro instruction in which either the CALL=IN or CALL=OUT operand has been specified.

The format of the VTERM macro instruction is:

Name	Operation	Operand
symbol	VTERM	LCST=mtalcst name [,BUFLIM=n | 2] [,LOGAPPL=application program name] [,LOGTAB=interpret table name]

symbol

is one to eight alphanumeric characters and must not begin with a $ character. This symbolic name identifies this as the node name assigned to the terminal (identified by the LCST operand in this VTERM macro instruction).

LCST=mtalcst name

indicates the name of the MTALCST macro instruction associated with this VTERM macro instruction. The name must be identical to the symbol specified in the MTALCST macro instruction. (The MTALCST macro instruction defines the specific operating parameters for the type of terminal and line control that are represented by this VTERM macro instruction.)

BUFLIM=n | 2

indicates a multiplier (integer number in the range 1 through 255) that ACF/VTAM uses to determine the maximum number of buffers that can be filled with data that has been read ahead from this terminal, but has not yet been transferred into the application program's buffers. The maximum number of buffers is the product of this BUFLIM value and the value coded in the BUFFACT operand (in the APPL definition statement that defines the application program being executed).

For example, if BUFFACT=1 (the default value) and BUFLIM=2 (the default value), 2 becomes the maximum number of buffers that can be allocated to receive input from the terminal.

After ACF/VTAM multiplies the BUFLIM and BUFFACT values together, the product becomes effective after the application program issues the OPNDST macro instruction.

If the amount of data read in exceeds the calculated maximum capacity of the buffers, then ACF/VTAM issues a RESET macro instruction with the OPTCD=UNCON operand. This macro instruction cancels the I/O operation, and the excess data is lost.

To determine if data was lost, application programs must check the return codes in the RPL fields for the I/O request (RTNCD and FDBK2). The RPLs to be checked are those for the READ, WRITE (OPTCD=CONV), or DO macro instructions.

The following information should be considered when determining the BUFLIM value:

Number of terminals expected to be in concurrent use

Characteristics of the terminals (such as the terminal's hardware features, speed, and use or type of transaction it is to perform)

Characteristics of the application programs expected to be used with the terminals (such as updating or retrieving data)

LOGAPPL=application program name

indicates the one to eight character name of an application program (defined by an APPL statement) to which this terminal is to be automatically logged on when it is active and dialed up. The name must correspond to the applname assigned to the application program by an APPL definition statement. If this operand is not coded, either the application program or the network operator must initiate logon to the terminal type.

LOGTAB=interpret table name

indicates the name of an interpret table. LOGTAB permits a terminal user to initiate a logon from this terminal, and associates the specified interpret table with this terminal. LOGTAB must be coded in this VTERM macro instruction if you plan to use terminal-initiated logon, unless you plan to exclusively use OS/VS logon.

ACF/VTAM-Only Operands And Coding Restrictions

ACF/VTAM requires information contained in the operands in the NCP generation macro instructions. Most of the information required by ACF/VTAM is also required by the NCP, although some additional information is required only by ACF/VTAM. The ACF/VTAM-only operands convey no information to the network control program generation assembly process. However, they must appear in the NCP generation deck that defines this NCP to ACF/VTAM. (ACF/VTAM uses them as input to its initialization or activation processing.)

The macro instruction assembly step of the NCP generation process permits each ACF/VTAM-only operand to appear in the appropriate macro instruction. However, the assembly process does not check these operands for proper syntax or verify that any related operands are present or absent. ACF/VTAM does this when the NCP is first activated.

The BUILD Macro Instruction

ACF/VTAM places restrictions on these BUILD macro instruction operands:

LOADLIB=dsname

specifies the data set in which the NCP resides. Specify this data set name on a DD card in the ACF/VTAM start procedure.

MAXSUBA=n

Specify the MAXSUBA value as 3 or greater. See description of MAXSUBA start option in Chapter 3 for more information.

NEWNAME=symbol

Specifies the name of the generated NCP load module. Specify this name in the ACF/VTAM start procedure.

OLT=NO | YES

If TOLTEP is to be used for terminals connected to this communications controller are to use TOLTEP, OLT=YES must be coded or assumed by default.

SUBAREA=n

Specify the SUBAREA value as 2 or greater. This value must be unique in the network when the major node is active. This is the SUBAREA address of the NCP being generated.

TYPGEN=value

must be NCP-R if RNAME is specified in the PCCU definition statement. Must not be NCP-R if CUADDR is specified in the PCCU definition statement.

The DIALSET Macro Instruction

It is recommended that dial sets do not span across domains. That is, all the resources specified in one DIALSET macro instruction should be in the same domain.

The SYSCNTRL Macro Instruction

ACF/VTAM always requires these OPTION parameters:

BHSASSC	Modify block handler set association*
ENDCALL	Physical disconnection for dial-up terminals
MODE	Set destination mode
RCNTRL	Request control mode reset
RCOND	Reset conditional
RECMD	Reset at end of command
RIMM	Reset immediate

*If an MNOTE (severity code) 4 occurs from NCP generation because no block handlers were specified for the NCP, ignore the MNOTE.

ACF/VTAM requires these OPTION parameters for certain operator control functions:

NAKLIM	Change line negative polling response limit
SESSION	Change session limit
SSPAUSE	Change service-seeking pause
XMTLMT	Change device transmission limit

The HOST Macro Instruction

ACF/VTAM places restrictions on these operands:

BFRPAD=0
must be coded exactly as shown.

STATMOD=YES
should be coded exactly as shown.

UNITSZ=value
The UNITSZ value must be the same as the bufsize value in the IOBUF buffer pool start option. If more than one NCP is to be active concurrently, the UNITSZ value must be the same for each NCP. The value should be a number of bytes that is evenly divisible by 4.

The MAXBFRU, SUBAREA, and UNITSZ values must be the same for ACF/VTAM as they are for the NCP.

The NCP Network Configuration Macro Instructions

The NCP network configuration macro instructions are the LU, PU, GROUP, LINE, CLUSTER, TERMINAL, COMP, and VTERM macro instructions. Figure 2-4 summarizes the ACF/VTAM-only and ACF/VTAM-restricted operands for these macro instructions. It does not indicate the conditions (for example, type of line control or type of terminal) under which an operand can be used. For this and other information, consult the individual description of each ACF/VTAM-only operand that follows. The information in this book about the ACF/VTAM-only operands and restrictions should be used in conjunction with the NCP coding requirements described in the *NCP Generation Manual.*

Figure 2-4 can also be used to determine what procedure is required to change operands and macro instructions for an existing NCP. If ACF/VTAM-only operands are to be changed or replaced, no NCP generation is required. If any NCP macro instruction or operand is changed or replaced, a partial or complete NCP generation is required.

	PU	LU	GROUP	LINE	CLUSTER	TERMINAL	COMP	LUPOOL
ADDRESS						*		
ANSWER			V	V				
ATTN			*	*		*	*	
AUTO				*				
BATCH	*	*	*	*				
BHSET			*	*	*	R	*	
BNNSUP	*		*	*				
BUFLIM			V	V		V	V	
CALL			*	*				
CDATA			*	*	*	*	*	
CONV			*	*		*		
CUTYPE			*	*	*			
DEVICE						V	V	
DIAL			*					
DIALNO							*	
DIALSET			*	*			*	
DIRECTN			*	*			*	
DISCNT	V		V	V	V			
DLOGMOD	V	V	V	V	V	V		
ENCR	V	V	V	V				
ENDTRNS			*	*		*	*	
EXEC			*	*	*	*	*	
FEATURE			*	*	*	*		
FEATUR2			V	V	V	V		
GPOLL					R			
IDSEQ			*	*		*		
IRETRY	*		*	*				
ISTATUS	V	V	V	V	V	V	V	
ITBMODE			*	*	*	*	*	
LCST								
LNCTL			*					
LOCADDR		*						
LOGAPPL	V	V	V	V	V	V	V	
LOGTAB	V	V	V	V	V	V	V	
MAXDATA	*		*	*				
MAXLU	*		*	*				
MAXOUT	*			*				
MODETAB	V	V	V	V	V	V		
MTALIST			*	*				
OWNER			V	V				V
PACING	*	*	*	*				
PASSLIM	*		*	*				
PAUSE			*	*				
POLIMIT			R	R				
POLL						*		
POLLED			*	*				
PT3EXEC			*	*	*	*	*	
PU			V	V				
PUTYPE	*		*	*				
SESSION			R	R				
SPEED			*	*				
SSCPFM	V	V	V	V				
TERM			R	R	R	R		
TYPE			*	*				
USE			R	R				
USSTAB	V	V	V	V	V	V		
UTERM					V			
VIDSEQ			V	V				
VPACING	V	V	V	V				
XMITLIM			R	R	R	R	R	

V = ACF/VTAM only; description is in this book.

R = ACF/VTAM restriction described in this book; the operand is described in *NCP Generation Manual.*

* = NCP only; described in *NCP Generation Manual.*

Figure 2-4. NCP Generation Operands Used by ACF/VTAM

After changing an ACF/VTAM-only or ACF/VTAM-restricted operand in an NCP macro instruction, a copy of the updated NCP generation deck must also be put in SYS1.VTAMLST.

When a member in SYS1.VTAMLST is updated, the copy of the corresponding resource definition table (RDT) on SYS1.VTAMOBJ should be deleted (using an operating system utility program that can delete a member of a partitioned data set). If the copy is deleted, ACF/VTAM builds (the next time the major node is activated) a new RDT (based on the modified definition) and stores a new copy on SYS1.VTAMOBJ. If the copy is not deleted, ACF/VTAM uses the old RDT and the changes are not reflected in ACF/VTAM's operations.

Although not all the information coded in the NCP macro instructions is used by both ACF/VTAM and the NCP, all the NCP macro instructions should be coded with the possible needs of both ACF/VTAM and the NCP in mind.

The COMP Macro Instruction

The COMP macro instruction represents start-stop or BSC components in the network. ACF/VTAM does not allow the COMP macro instruction on a dial line.

Description of ACF/VTAM-Only Operands And Restrictions

ANSWER=ON | OFF

is valid only if the SDLC line has dial-in capability. If ANSWER=ON is specified, physical units can dial in to the NCP when the line is active. If ANSWER=OFF is specified, the physical unit cannot dial in to the NCP, regardless of the active-inactive status of the line.

BHSET=setname | NONE | DYNAMIC

If the TERMINAL macro instruction is used to define a port for dial-in terminals (CTERM=YES specified), a block handler set can be assigned to the dial-in terminal only if BHSET=DYNAMIC is specified. If BHSET=DYNAMIC is specified, ACF/VTAM assigns a block handler in accordance with the BHSET=setname specified on the TERMINAL macro instruction for the dial-in terminal.

BUFLIM=n | <u>2</u>

indicates a multiplier (integer number in the range 1 through 255) that ACF/VTAM uses to determine the maximum number of buffers that can be filled with data that has been read ahead from this terminal, but has not yet been transferred into the application program's buffers. The maximum number of buffers is the product of this BUFLIM value and the value coded in the BUFFACT operand (in the APPL definition statement that defines the application program being executed).

For example, if BUFFACT=1 (the default value) and BUFLIM=2 (the default value), then 2 becomes the maximum number of buffers that can be allocated to receive input from the terminal.

After the BUFLIM and BUFFACT values have been multiplied together, the product becomes effective when the application program issues the OPNDST macro instruction.

If the amount of data read in exceeds the calculated maximum capacity of the buffers, then ACF/VTAM issues a RESET macro instruction with the OPTCD=UNCON operand. This macro instruction cancels the I/O operation, and the excess data is lost.

To determine if data was lost, application programs must check the return codes in the RPL fields for the I/O request (RTNCD and FDBK2). The RPLs to be checked are those for the READ, WRITE (OPTCD=CONV), or DO macro instructions.

The following information should be considered when determining the BUFLIM value:

Number of terminals expected to be in concurrent use

Characteristics of the terminals (such as the terminal's hardware features, speed, and use or type of transaction it is to perform)

Characteristics of the application programs expected to be used with the terminals (such as updating or retrieving data)

CALL=IN | OUT | INOUT (LINE macro instruction, for SDLC)
The CALL operand indicates whether terminals, the host computer, or both are able to cause switched connections to be set up over the line to which this operand applies.

If the line is to be used only for terminal initiated connections, CALL=IN must be coded on the LINE macro instruction for the line.

If the line is to be used only for connections initiated by ACF/VTAM, the network operator, or application programs, CALL=OUT must be coded on the LINE macro instruction for the line.

If the line is to be used for connections initiated by terminals, in addition to connections initiated by ACF/VTAM, the network operator, or application programs, CALL=INOUT must be coded on the LINE macro instruction for the line.

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

DEVICE=devicetype
indicates the device type for a terminal component of either the IBM 1050 or 2770 Data Communication Systems. The valid devicetypes are:

1050 System Components	2770 System Components
1052	50
1053	545
1054	1017
1055	1018
1056	1053
1057	1255
1058	2203
1092	2213
1093	2265
	2502
	5496

DEVICE should be coded on the COMP macro instruction if ACF/VTAM is to be capable of establishing individual sessions with each component, or if an application program is to be capable of determining device type by using the INQUIRE macro instruction (OPTCD=DEVCHAR).

The DEVICE operand in the COMP macro instruction can also be used in conjunction with the TERM and DEVICE operands in the TERMINAL macro instruction. For example:

```
symbol1 TERMINAL TERM=1050,DEVICE=1052,...
symbol2 COMP DEVICE=1054,...
symbol3 COMP DEVICE=1056,...
```

In the preceding example, if DEVICE is not coded in the COMP macro instruction labeled symbol2, ACF/VTAM uses the DEVICE value in the TERMINAL macro instruction associated with this COMP macro instruction. However, if DEVICE is not coded in the TERMINAL macro instruction, ACF/VTAM uses the TERM values in the TERMINAL macro instruction for this COMP macro instruction.

DISCNT=([YES | NO] [,F | NF])

YES | NO

specifies whether ACF/VTAM is to physically disconnect the physical unit when the last logical unit is disconnected by its application program (that is, when there are no more sessions between an application program and a logical unit).

When DISCNT=YES is specified, the physical unit is automatically disconnected as soon as the last logical unit is disconnected by its application program. If any logical units request their own disconnection, ACF/VTAM ignores the part of their disconnection request that indicates whether the physical unit is to be disconnected (that is, the HOLD part of the character-coded logoff command, or the LAST-NOT LAST part of a Terminate Self command is ignored). ACF/VTAM also rejects any attempt by the physical unit to request its own disconnection (using the Request Discontact command).

When DISCNT=NO is specified, ACF/VTAM disconnects the physical unit only when one of the following conditions is met:

ACF/VTAM receives a Request Discontact Immediate command from the physical unit.

ACF/VTAM receives a Request Discontact Normal command from the physical unit, and all logical units have been disconnected.

All logical units have been disconnected as a result of a character-coded logoff command for which HOLD=NO was specified (or inserted using a USS definition table) or a Terminate Self command for which LAST was specified.

When DISCNT=YES is specified for a physical unit on a non-switched line, the physical unit is deactivated and before it can be used again, it must be reactivated by the network operator.

F | NF

specifies whether ACF/VTAM is to indicate "final-use" status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES. This operand does not apply when DISCNT=NO is specified, nor does it have any effect on the VARY INACT command. If F is specified or assumed by default, "final-use" status is indicated. If NF is specified, "not-final-use" status is indicated. Each device has its own requirements regarding "final-use" status. To determine whether F or NF should be specified for a given device, consult the appropriate installation publication for the device.

DLOGMOD=default logmode entry

specifies the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default.

ENCR=REQD | SEL | OPT | NONE

can be specified only if the ACF/VTAM Encrypt/Decrypt Feature is installed. It specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD

specifies that ACF/VTAM must encipher all messages to and from this logical unit. If ENCR=REQD is coded, no sessions can be established with this logical unit unless the host with which it is associated is able to handle cryptographic sessions.

SEL

has no meaning for logical units. If ENCR=SEL is entered, ENCR=OPT is used instead.

OPT

specifies that this logical unit is capable of engaging in cryptographic sessions, but allows the application program to determine whether or not to use cryptography.

NONE

specifies that this logical unit is incapable of engaging in cryptographic sessions.

FEATUR2=(operand,operand . . .)

indicates the machine features for a specific remote non-SNA terminal. The operands are:

MODEL1 | MODEL2

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

PRINTR | NOPRINTR

indicates whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only if the TERM=3275 operand is also coded or assumed by NCP macro instruction sequencing for this macro instruction.

The following operands can be coded only if the TERM=3275 or TERM=3277 operand has been coded or assumed by the NCP macro instruction sequence:

ANKEY | NOANKEY

indicates whether this terminal has a standard alphanumeric keyboard.

DEKEY | NODEKEY

indicates whether this terminal has the data entry keyboard.

PFK | NOPFK

indicates whether this terminal has the program function keys.

GPOLL=chars

GPOLL (general polling) must be specified for 2980 and 3270 non-SNA clusters; ACF/VTAM does not support specific polling.

ISTATUS=ACTIVE | INACTIVE

indicates whether this minor node is to be initially active when the NCP major node to which it belongs is first activated, either due to a cold start of ACF/VTAM, or, after ACF/VTAM is started, by issuing the VARY ACT,COLD command for the NCP major node. For logical units, ISTATUS is also effective whenever the associated physical unit is activated.

This operand "sifts" to subordinate nodes as described in " 'Sift-down Effect' in NCP Macro Instructions" earlier in this chapter. For a TERMINAL macro instruction defining a port or logical connection (CTERM=YES), ISTATUS applies to both the port and the dial-in terminal defined by the UTERM operand.

When coding the ISTATUS operand, consider these special cases:

For a line with TYPE=PEP specified, ISTATUS does not apply and is ignored if specified. For a line with USE=NCP specified, the line will be initially active. For a line with USE=EP specified, the line will be initially inactive.

For a remote NCP, ISTATUS does not apply to any line connected back to the local NCP and is ignored if specified.

If a line is to be used only as a backup, it is recommended that ISTATUS=INACTIVE be specified.

When coding a PU macro instruction for a type 4 physical unit, ISTATUS=ACTIVE should be specified only if TYPE=(4,LOCAL) and the SUBAREA operand are specified. If ISTATUS=ACTIVE is specified for any other kind of type 4 physical unit, an error message is issued and ISTATUS=INACTIVE is used instead.

Note that in ACF/VTAM the ISTATUS specification on a LINE macro instruction applies both to the line itself and to the nodes that are subordinate to the line. In VTAM Level 2, the ISTATUS specification on a LINE macro instruction applied only to the nodes subordinate to the line.

LOGAPPL=application program name

indicates the name of an application program (one to eight alphanumeric characters) to which this terminal is to be automatically logged on when it is activated. The name must correspond to the name assigned to the application program by an APPL definition statement.

If you want to use a network solicitor to monitor logons from terminals that use basic mode, 'application program name' must be coded as the name assigned to that network solicitor.

If the terminal is a printer, do not code name as the name assigned to a network solicitor. The IBM network solicitor cannot control output-only devices.

For non-SNA terminals, if this operand is not coded, either the application program or the network operator must initiate terminal logons. For logical units, USS facilities can be used for terminal initiated logon.

For more information on automatic logon, see Chapter 4.

LOGTAB=interpret table name

indicates the name of an interpret table (one to eight alphanumeric characters).

LOGTAB permits a terminal user to initiate a logon from this terminal and associates the specified interpret table with this terminal. If you do not use OS/VS logon, LOGTAB must be coded in the NCP macro instructions that define this terminal.

If this terminal is to be controlled by a network solicitor, LOGTAB must be coded in the NCP macro instructions that define this terminal. This allows the selected network solicitor to use the specified interpret table to validate the logon message before routing it to the specified application program.

MODETAB=logon mode table name

specifies the name of a logon mode table to be used for the logical unit. Logon mode tables are described in Chapter 4. If this operand is omitted, the IBM-supplied logon mode table is used for the logical unit.

OWNER=ownername

specifies a name (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) that is used to associate a host ACF/VTAM with the resources that it controls. This operand is intended for the division of resources in a multiple-channel-attached communications controller. This operand indicates that this resource is a subsidiary resource associated with the PCCU macro instruction having the same OWNER value. This operand applies only to a local communications controller and is ignored when specified for a remote communications controller.

PACING=n | 0 | 1

specifies the way that pacing is to be handled between the logical unit and the NCP to which the logical unit is connected. (In contrast, VPACING involves pacing between ACF/VTAM and the NCP.) For more information on pacing, see the "Pacing" section in this chapter.

n

specifies the number of normal-flow messages that the NCP is to send to the logical unit before waiting for a pacing response.

Specify n as a decimal integer in the range 1 through 255, referring to the *NCP Generation Manual* for information on how to select the proper value.

0

specifies that no pacing is to be performed for sessions with the logical unit.

If the PACING operand is omitted, PACING=1 is assumed by default.

POLIMIT=([n][,WAIT | NOWAIT | QUEUE])

This operand must be coded for a polled nonswitched line. Specify QUEUE as the action to be taken by the NCP if the specified number (n) of negative responses is exceeded when polling any terminal except the IBM 3735. For the 3735, specify WAIT.

The NCP default value of (1,NOWAIT) can only be used with BSC 3270s.

PU=YES | NO

specifies whether a BSC 3270 is to be treated as a physical unit and all terminals beneath the cluster are to be treated as logical units. When PU=YES is specified, the terminals can communicate only with application programs that operate in record mode. This operand cannot be specified on the CLUSTER macro instruction. It must be coded on the GROUP or LINE macro instruction, to be sifted down to the cluster level.

SESSION=count | 1

This operand establishes the NCP line scheduling session limit for the line. It should be specified equal to the number of terminals on the line so that it is possible for the NCP to have concurrent sessions with all terminals on the line. This operand refers to NCP polling operations, not SNA sessions.

SSCPFM=FSS | USSSCS | USS3270

specifies whether or not a logical unit or terminal can support character coded messages (SSCPFM=USSSCS) in its communication with the SSCP. The default for this operand is SSCPFM=FSS, except when coded on a CLUSTER or TERMINAL macro instruction (see below).

Code SSCPFM=USS3270 for terminals attached to a control unit defined as a BSC 3271, BSC 3275, SDLC 3271 (PUTYPE=1), or SDLC 3275 (PUTYPE=1). For a list of these control units, see *ACF/VTAM Concepts and Planning.*

Code SSCPFM=FSS or SSCPFM=USSSCS for all other terminals for which the SSCPFM operand is valid. Consult the individual terminal component description manual to determine whether character coded messages (SSCPFM=USSSCS) or formatted messages (SSCPFM=FSS) are supported for SSCP communication.

If SSCPFM is to be coded on a CLUSTER or TERMINAL macro instruction, it must be specified as SSCPFM=USS3270 and the operand PU=YES must be specified on the GROUP or LINE macro instruction. For CLUSTER macro instructions with PU=YES in effect, the default for SSCPFM is SSCPFM=USS3270.

TERM=type

Device types 2020 and 2025 are invalid.

USE=NCP | EP

If USE=EP is specified, the line is not activated unless a VARY ACT command is issued for the line. Activating such a line with a VARY command gives the line to ACF/VTAM and NCP whether or not it is in use by EP.

USSTAB=USS definition table name

specifies the name of a USS definition table to be used for the logical unit. USS definition tables are described in Chapter 4. If this operand is omitted, the IBM-supplied USS definition table is used for the logical unit when character-coded requests are received.

If USSTAB is to be coded on a CLUSTER or TERMINAL macro instruction, the operand PU=YES must be specified on the GROUP or LINE macro instruction.

UTERM=name (TERMINAL macro instruction)

indicates the name (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) of a dial-up terminal with which an application program can establish connection. UTERM is coded only if the CTERM=YES operand is also coded in the same TERMINAL macro instruction. This terminal cannot also be specified in an VIDLIST or VTERM macro instruction.

UTERM is required for these dial-up terminals:

A BSC dial-up terminal for which no ID verification is to be performed or for which no ID verification is to be performed in the application program

A start-stop dial-up TWX terminal for which ID verification is to be performed in the application program

A start-stop dial-up terminal that is not on a line supported by MTA

A start-stop dial-up terminal that is on a line supported by MTA, but which has a terminal type that ACF/VTAM is not checking

If a dial-up terminal calls in to ACF/VTAM and ACF/VTAM cannot identify it (by using the UTERM operand specified in the related TERMINAL macro instruction, the VIDLIST macro instruction, or the VTERM macro instruction), ACF/VTAM does not allow the terminal to establish a connection.

VIDSEQ=symbol

indicates the VIDLIST macro instruction that defines the ID sequence and associated terminal (BSC or TWX on a switched line) that is permitted to call into ACF/VTAM for ID verification. The symbol in the VIDSEQ operand must be identical to the symbol assigned to that VIDLIST macro instruction.

This operand is valid only for BSC or TWX terminals on a switched line.

If VIDSEQ is coded in a GROUP macro instruction, the DIAL=YES operand must also be specified in that macro instruction.

If VIDSEQ is coded in a LINE macro instruction, the POLLED=NO operand must also be coded or assumed by macro instruction sequencing in that macro instruction.

VPACING=n | 0 | 2

indicates whether pacing is to be performed (during a session) with the logical units represented by the appropriate NCP macro instructions. For more information on pacing, refer to the "Pacing" section this chapter.

n

indicates the number of data requests that ACF/VTAM is to send to an NCP (for an associated session), and then wait for a pacing response. No further requests can be sent to the NCP until the NCP replies with a pacing response to ACF/VTAM.

0

indicates that no pacing is to be performed.

If the VPACING operand is omitted, VPACING=2 is assumed.

XMITLIM=1

must be coded for all terminals on a multipoint line.

Defining A Multidomain ACF/VTAM Network

If the Multisystem Networking Facility is installed in your system, you must define some additional network elements before you can use this facility. The additional elements to be defined are cross-domain resource managers, cross-domain resources, and path tables. A cross-domain resource manager is the portion of the system services control point (SSCP) that controls cross-domain connections. A cross-domain resource is an application program or logical unit in another domain that is available for use by resources in your domain (for example, sessions can be established between resources in your domain and cross-domain resources). A path table is used to define the paths between various domains.

Defining Cross-Domain Resource Managers

One or more major nodes can be defined for cross-domain resource managers (CDRMs). Each CDRM major node is defined with a VBUILD definition statement and each minor node is defined with a CDRM definition statement. Within a domain there must be exactly one cross-domain resource manager defined for every other domain with which this domain will communicate. Each domain must also have within itself a definition of its own cross-domain resource manager.

The VBUILD Statement

Code a VBUILD statement in 80-byte card-image format for each set of CDRM definition statements.

Write the VBUILD statement as follows:

Name	Operation	Operand
[name]	VBUILD	TYPE=CDRM [,CONFGDS=ddname] [,CONFGPW=password]

name
one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. Specification of a name is optional.

When this major node is activated, the value must be unique among the active major nodes.

TYPE=CDRM
specifies that the VBUILD statement defines a CDRM major node to ACF/VTAM. This operand is required.

CONFGDS=ddname
is a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure. Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities" for a description of configuration restart.

CONFGPW=password
specifies the one to eight character alphanumeric password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified but is required by VSAM, the network operator is prompted by VSAM to provide the correct password when ACF/VTAM attempts to open the data set. This operand can be specified only if the CONFGDS operand is specified.

The CDRM Definition Statement

Name	Operation	Operands
cdrmname	CDRM	SUBAREA=n [,ELEMENT=n | 1] [,ISTATUS=ACTIVE | INACTIVE] [,VPACING=n | 0 | 63]

CDRM definition statements define cross-domain resource manager (CDRM) minor nodes. One or more of these statements, when preceded by a VBUILD statement, define a CDRM major note, which is filed as a member in SYS1.VTAMLST.

cdrmname

is the name (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) of the CDRM represented by this statement. This name is required.

SUBAREA=n

this is the subarea number (a decimal integer in the range 1 through 255) that represents the subarea of the host computer in which the CDRM represented by this statement resides. The SUBAREA value for each subarea to be used in cross-domain communications must be unique throughout the whole network. This operand is required.

ELEMENT=n | 1

is the element part of the network address of the CDRM. This operand, in conjunction with the previous SUBAREA value, defines the network address of the CDRM. For CDRMs that represent a host ACF/VTAM, the ELEMENT value must be 1. For other types of CDRMs this number can be a decimal integer from 0 up to the maximum number of elements in the subarea. If ELEMENT is not specified, ELEMENT=1 is assumed by default.

ISTATUS=ACTIVE | INACTIVE

specifies the initial status of the CDRM.

For the host CDRM, ACTIVE specifies that it is capable of going into session with an external CDRM (a CDRM in another domain).

For the host CDRM, INACTIVE specifies that it is *not* capable of going into session with an external CDRM.

For an external CDRM, ACTIVE specifies that a CDRM to CDRM session should be established (from the CDRM in the host in which this set is being defined to the CDRM named by this statement). Note that the attempt to establish this session fails if the host CDRM is not active, so ISTATUS=ACTIVE should be specified only if this CDRM is in the same major node as the host CDRM, or if this CDRM is to be activated after the host CDRM is active.

For an external CDRM, INACTIVE specifies that a CDRM-CDRM session is *not* to be established.

After its major node has been activated, a CDRM defined as INACTIVE can be activated by using the VARY command, or in the case of a CDRM in another domain, by receiving a session activation request from that domain.

VPACING=n | 0 | 63

must be a decimal number in the range 0 through 63 that indicates the maximum number of requests that other CDRMs should send to this CDRM before waiting to receive a pacing response. This value is exchanged during CDRM-CDRM session establishment. If the value is not in the range 0 through 63, the maximum of 63 is used. If VPACING=0 is specified, no pacing is done.

Defining Cross-Domain Resources

One or more major nodes can be defined for cross-domain resources (CDRSCs). Each CDRSC major node is defined with a VBUILD definition statement and each minor node is defined with a CDRSC definition statement.

The VBUILD Statement

Code a VBUILD statement in 80-byte card-image format for each set of CDRSC definition statements.

Write the VBUILD statement as follows:

Name	Operation	Operand
[name]	VBUILD	TYPE=CDRSC [,CONFGDS=ddname] [,CONFGPW=password]

name
one to eight alphanumeric characters beginning with an alphabetic character other than a $ character. Specification of a name is optional.

TYPE=CDRSC
specifies that this VBUILD statement defines a CDRSC major node to ACF/VTAM. This operand is required.

CONFGDS=ddname
is a one to eight character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the ACF/VTAM start procedure. Refer to Chapter 6, "Reliability, Availability, and Serviceability Facilities" for a description of configuration restart.

CONFGPW=password
specifies the one to eight character alphanumeric password, if required, for ACF/VTAM to gain access to the configuration restart data set. If CONFGPW is not specified but is required by VSAM, the network operator is prompted by VSAM to provide the correct password when ACF/VTAM attempts to open the data set. This operand can be specified only if the CONFGDS operand is specified.

The CDRSC Definition Statement

Name	Operation	Operands
cdrscname	CDRSC	CDRM=cdrmname [,ISTATUS=ACTIVE | INACTIVE]

CDRSC definition statements define CDRSC (cross-domain resource) minor nodes for those cross-domain resources with which an application program or logical unit within this domain can have a session. One or more of these statements, when preceded by a VBUILD statement, defines a CDRSC major node and is filed as a member in SYS1.VTAMLST.

cdrscname

This required symbol (one to eight alphanumeric characters, beginning with an alphabetic character other than a $ character) is the name of a resource in another domain (either a logical unit or an application program) represented by this statement. If the CDRSC is an ACF/VTAM application program, the name must be the same as the name specified by the 'name' field (not the ACBNAME operand) of the application program's APPL statement in the ACF/VTAM definition of the other domain.

CDRM=cdrmname

specifies the name of the CDRM that is in the same domain as the cross-domain resource, and which controls the resource. The CDRM named here must not be this host CDRM.

ISTATUS=ACTIVE | INACTIVE

specifies the initial status of this CDRSC.

ISTATUS=ACTIVE indicates that this resource is logically active to this domain. The resource is not necessarily active in its own domain.

ISTATUS=INACTIVE indicates that this resource is not logically active to this domain. The resource is not necessarily inactive in its own domain.

Defining Path Tables

Path tables are representations of the routes ACF/VTAM takes to communicate with subareas not in its own domain. One or more PATH statements (VBUILD is not used) defines a path table and together they are filed as a member in SYS1.VTAMLST. The member name is used to activate a path table.

Cross-domain communication is not possible unless all the required path tables are active. It is recommended, therefore, that any required path tables be activated before any of the CDRM and CDRSC major nodes are activated.

Note that there is another PATH statement, described elsewhere in this chapter, that is used when defining a switched SNA major node. Be careful not to confuse that PATH statement with the one used to define cross-domain paths.

The Cross-Domain Path Statement

Name	Operation	Operands
symbol	PATH	ADJSUB=n ,DESTSUB=(n[,n] [,n] [. . .])

symbol

any symbol valid in the assembler language. This provides a name for the PATH statement and is not checked by ACF/VTAM for validity.

ADJSUB=n

specify n as the subarea number of the adjacent subarea, lying within the domain, through which records intended for the DESTSUB subareas are to be routed.

DESTSUB=(n[,n][,n][...])

specify n as the subarea number of the destination subarea, lying outside of the domain, to which records are to be sent. More than one destination subarea can be specified for each adjacent subarea.

Filing Network Definition Statements

To define major nodes and file the definitions as members in SYS1.VTAMLST, follow these steps. (These steps can also be followed when making changes or additions to the definitions after they have been put into SYS1.VTAMLST.)

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

   Follow the coding requirements for the NCP-only definition statements and operands and use the NCP generation instructions in the *NCP Generation Manual.*

   To improve the time needed to concurrently load multiple NCPs into their communications controllers, the NCP load program attributes can be changed.

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

   If changing or adding NCP macro instructions or operands, change the NCP source deck and do a partial or complete NCP generation (as described in the *NCP Generation Manual.*

   When defining any major node, use *ACF/VTAM Installation Guide* to also consider the coding dependencies with any related ACF/VTAM definition statements, NCP macro instructions, and ACF/VTAM console (network operator) commands.

2. Before starting ACF/VTAM, an operating system utility program (such as the IEBUPDTE utility program) must be used to put the definitions in SYS1.VTAMLST as members.

3. When updating a member in SYS1.VTAMLST, the copy of the corresponding resource definition table (RDT) on SYS1.VTAMOBJ must be deleted (using an operating systems utility program that can delete a member of a partitioned data set). If the copy is deleted, ACF/VTAM builds (the next time that major node is activated) a new RDT (based on the modified definition) and stores a new copy on SYS1.VTAMOBJ.

   If the MAXSUBA value for the host ACF/VTAM is changed, SYS1.VTAMOBJ must be deleted and all the associated NCPs must be regenerated.

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

Defining Host Backup And Resource Takeover Facilities

ACF/VTAM and ACF/NCP/VS allow you to define the system in such a way that different hosts can share the same set of NCP resources. In such a system, if one host fails another host can assume control of those NCP resources that belonged to the failing host. This is done with the VARY ACT command or, if the Multisystem Networking Facility is installed, with the VARY ACQ command. If the Multisystem Networking Facility is installed, the host in one domain can acquire the resources of a channel-attached 3705 communications controller in an adjacent domain. Also, if the Multisystem Networking Facility is installed, a local NCP acquired over a cross-domain link can be redefined as a remote NCP.

Before you code the BACKUP, OWNER, and SUBAREA operands of the various ACF/VTAM definition statements you must decide if and how you will use these backup and resource sharing facilities. For information on these topics, see *ACF/VTAM Installation Guide.*

Defining Uninstalled Devices Or Programs

You can define groups, lines, clusters, logical units, terminals, or application programs that are not available when starting ACF/VTAM. Then, after the devices or programs are installed, the ACF/VTAM definition statements or NCP macro instructions do not have to be changed. To do this, code the appropriate definition statements or macro instructions, and specify the ISTATUS=INACTIVE operand for each uninstalled device.

When defining a large future configuration, remember that unused RDT entries increase virtual storage space requirements, affect performance, and increase initialization time.

Changing NCP Load Program Attributes

To reduce the time needed to concurrently load multiple NCPs into communications controllers (especially into remote communications controllers), you can change the attributes for an NCP load program. Two modules, the NCP load program (IFLOADRN) and the ACF/VTAM load program interface (ISTINC05), load NCPs sequentially into communications controllers. By using the linkage editor, you can change the attribute of these modules from "serially reusable" to "not reusable". This attribute causes separate copies of these NCP load programs to be brought into virtual storage for each NCP that is to be loaded, reducing loading time by providing concurrent NCP loading.

Chapter 3. Starting And Controlling The Network

When starting ACF/VTAM, the network operator can specify start options to define the initial network configuration and to select optional ACF/VTAM facilities. These start options can be specified in one or a combination of the following ways:

The network operator can specify start options in the START command.

The network operator can respond to prompting messages during ACF/VTAM initialization.

The network operator can specify lists of start options that were previously defined by the system programmer.

The IBM-supplied start options can be assumed by default (unless they were overridden in one of the preceding ways).

Once specified, the start options remain in effect until the network operator modifies them or until ACF/VTAM is terminated.

Start Options

By using ACF/VTAM start options you can define how ACF/VTAM starts and controls the network. By creating a configuration list, you can specify the major nodes that are to be in the network's initial configuration. By selecting optional facilities, you can start or stop ACF/VTAM traces, activate or deactivate a network solicitor, allow the network operator to be prompted, and establish the size, number, slowdown point, storage location (fixed or pageable), expansion size, and expansion point for each ACF/VTAM buffer pool. After selecting the start options for the network, you can include them in start option lists, provide them to the network operator, or use them in a combination of both ways.

When determining what start options to use, ACF/VTAM gives different priorities to different types of start options. Start options with a high priority override those with a lower priority. Here is a list of the different sources of start options, with the highest priority sources listed first and the lowest priority sources listed last:

Start options (other than LIST) entered by the network operator

Predefined start option list in ATCSTRyy (if LIST=yy was specified)

Predefined start option list in ATCSTR00 (user-supplied default values)

IBM-supplied values

If start options conflict, the highest priority start option that was entered overrides the lower priority ones. For example, if NETSOL=YES is specified in ATCSTRyy, but NETSOL=NO is entered as a start option, ACF/VTAM uses NETSOL=NO.

Specifying Start Options

The network operator can enter start options (unless otherwise indicated in the following list of start options) as parameters on the START command or the options can be specified in a start option list. If the network operator enters the start options as parameters on the START command, the length of the entire number of options must not exceed the console's line length.

For OS/VS1: The format of the START command is:

```
procname.Pnn,,,(option,option,. . .,option)
```

procname.Pnn
is the name of the ACF/VTAM start procedure followed by the number of the partition in which ACF/VTAM is to run.

For OS/VS2: The format of the START command is:

```
procname,,,(option,option, . . . ,option)
```

procname
is the name of the ACF/VTAM start procedure.

option
is one of the ACF/VTAM start options. The total number entered is limited by the line length of the console.

The formats of the start options are:

```
  SSCPID=n
  pool name=(baseno,bufsize,slowpt,F,xpanno,xpanpt) (see note 1)
  COLD | WARM
  CONFIG=xx | 00 | name
  DLRTCB=n
  HOSTSA=n | 1
  LIST=yy | 00 (see note 2)
  MAXAPPL=n | 10
  MAXSUBA=n | 15
  NETSOL=YES | NO
  NODELST=name
  PROMPT | NOPROMPT (see note 3)
  SUPP=NOSUP | INFO | WARN | NORM | SER
{ TNSTAT[,CNSL | NOCNSL] [,TIME=n | 60] }
{ NOTNSTAT                              }
  {TRACE | NOTRACE}, ID= { nodename,TYPE= BUF | IO | LINE } (see note 4)
                         { VTAMBUF,TYPE=SMS               }
  VTAMEAS=n | 10
```

If more than one option is specified, separate them with commas (see Figure 3-1).

NOTES
1. More than one buffer pool option can be specified for ACF/VTAM initialization.
2. LIST=00 can be included only in ATCSTR00 or ATCSTRyy. LIST=yy can be entered only by the network operator.
3. The network operator should not enter the PROMPT | NOPROMPT start option. It can only be specified in a predefined list.
4. Do not specify NOTRACE when starting ACF/VTAM, except to override a TRACE start option specified in a predefined list. Also, ACF/VTAM accepts more than one TRACE start option during initialization.

The SSCPID Start Parameter

SSCPID=n
is a decimal number between 0 and 65535. and unique in the network. It is part of an SSCP identifier, unique in the network, used when a physical unit or external CDRM establishes contact with ACF/VTAM. This parameter is required.

ACF/VTAM uses the SSCPID value to construct a 48-bit identification sequence that is sent to a physical unit when a session is established using the SNA ACTPU command. When the Multisystem Networking Facility is installed, it is also sent to another CDRM when a session is established by the SNA ACTCDRM command. The physical unit or cross-domain resource manager can thereby identify the SSCP with which it is in session. The identifier has the following form:

Bits 0-7:	X'05'
Bits 8-31:	X'000000'
Bits 32-47:	SSCPID in binary

The Buffer Pool Start Options

pool name=(baseno,bufsize,slowpt,F,xpanno,xpanpt)
describes a buffer pool used by ACF/VTAM for holding data or building control blocks.

pool name
is the name of the buffer pool to which these options apply. Below is a list of the buffer pool names and, for each buffer pool, an indication of the buffer pool's function and of whether it is normally located in fixed or pageable storage.

Pool Name	
APBUF	Active and inactive connections buffer pool in pageable storage
CRPLBUF	RPL-copy pool in pageable storage
IOBUF	Fixed storage message pool
LFBUF	Large fixed-storage buffer pool
LPBUF	Large pageable-storage buffer pool
NPBUF	Device connection buffer pool in pageable storage
PPBUF	Pageable data buffer pool
SFBUF	Small fixed-storage buffer pool
SPBUF	Small pageable-storage buffer pool
UECBUF	Use exit control block buffer pool in pageable storage
WPBUF	Message-control buffer pool in pageable storage

baseno
specifies the base number of buffers in the pool. Information to calculate baseno appears in Appendix C. The minimum number is 1; the maximum number is 32767.

Do not code a comma within the number. If baseno is specified as 0 or more than 32767, ACF/VTAM issues an error message and prompts the operator to reenter the specifications for the pool.

bufsize
specifies the size in bytes of each buffer in the pool. For buffer pools other than IOBUF and PPBUF, the *bufsize* value should not be changed from the IBM-supplied defaults (shown in Chapter 7) unless absolutely necessary.

For IOBUF and PPBUF, the value of *bufsize* must be the value used for the UNITSZ operand in the HOST macro instruction. If the values do not correspond, the NCP cannot be activated.

A user who includes local SNA major nodes in the network configuration must ensure that the bufsize value for the IOBUF buffer pool is large enough to include the largest possible RU segment that a physical unit can send.

The minimum and maximum values for bufsize are different for different buffer pools. If the value specified for *bufsize* is less than the minimum, ACF/VTAM increases the value to the minimum value and issues a message to the network operator. If *bufsize* is specified as 0 or more than the maximum, ACF/VTAM issues an error message and prompts the network operator to reenter the specification for the buffer pool. If *bufsize* for PPBUF is not at least as large as *bufsize* for IOBUF, ACF/VTAM increases the PPBUF *bufsize* value to the value of the IOBUF *bufsize* during ACF/VTAM initialization and issues a message notifying the network operator of this action.

slowpt[1]
is a decimal integer that specifies the slowdown point. When the number of buffers not in use in this buffer pool is equal to or less than the number specified, the pool enters slowdown mode as described in Appendix C. The value specified for *slowpt* must be less than *baseno.*

F

specifies that a buffer pool is to be fixed in storage. The F attribute has meaning only for buffer pools that are normally in pageable storage, but it can be coded for buffer pools that are located in fixed storage by default. If you intend to use the *xpanno* and *xpanpt* suboptions, but not the F attribute, then you must code a double comma between the *slowpt* suboption and the *xpanno* suboption. For example: LPBUF= (baseno, bufsize, slowpt,,xpanno,xpanpt). This applies whether the buffer pool's default attribute is fixed or pageable.

xpanno
is a decimal integer that specifies the number of buffers that ACF/VTAM is to acquire when expanding the buffer pool. This value is rounded upward to the number of buffers that will fill the nearest whole page of storage. The value specified must be in the range 0 through 32767. If *xpanno* is not specified, no buffer pool expansion occurs.

xpanpt
is a decimal integer that specifies the expansion point for this buffer pool. When the number of buffers not in use in the buffer pool falls to a value that is equal to or less than *xpanpt,* ACF/VTAM schedules an asynchronous routine to expand the buffer pool by the number of buffers specified by *xpanno.* The value of *xpanpt* must be greater than the value of *slowpt,* but less than the value of *baseno* minus *adjval,* where *adjval* is an adjustment value for this buffer pool. (See Figure 7-8 for the values of the adjustment value for each pool.) If you specify an *xpanpt* value, but omit the *slowpt* value, make sure that the *xpanpt* value is greater than the default *slowpt* value for the pool. If *xpanpt* is not specified, no buffer pool expansion occurs.

For example, suppose you want to specify that the user-exit control block buffer pool is to be in fixed storage with a base allocation of 10 buffers, have the IBM-supplied value for the buffer pool size, a slowdown point at 2 buffers, an expansion size of 5 buffers, and an expansion point at 3 buffers. To specify these conditions, code:

```
UECBUF=(10,,2,F,5,3).
```

[1] Do not confuse *slowpt* with the VTAM Level 2 subparameter *bth.* For VTAM Level 2, *bth* is specified in terms of the number of buffers that are in use, whereas for ACF/VTAM, *slowpt* refers to the number of buffers *not* in use.

Overriding Buffer Pool Values

All specified buffer pool values override the previously entered values, while unspecified values default to the IBM-supplied values. For example, if LPBUF=(10,92,9) is specified in ATCSTROO (LIST=00), but LPBUF=(15,92,13,F,15,11) is specified in ATCSTRST (LIST=ST), the latter specification for LPBUF is the effective start option after ATCSTRST is processed at startup. If, however, one or more suboptions are omitted from a buffer pool specification, an IBM-supplied value is used. In the preceding example, if LPBUF=(15,,13) is specified in ATCSTRST, the effective start option is LPBUF=(15,84,13,,0,0). The default value of 84 has been substituted for the bufsize suboption that was omitted. The double comma after the 13 indicates that the default attribute of pageable storage has not been overridden and the 0,0 specification for the expansion options indicates that this buffer pool is not to be expanded.

Continuing with the same example, if LPBUF is not specified in ATCSTROO or in ATCSTRST, ACF/VTAM supplies default values for the suboptions. (See Chapter 7 for a list of the IBM-supplied default values.) If LPBUF is specified in ATCSTROO, but omitted from ATCSTRST, the specification in ATCSTROO is in effect.

The COLD/WARM Start Option

COLD | WARM

specifies the status to which the configuration restart facility of ACF/VTAM is to restore each major node in the predefined configuration list referred to by the CONFIG start option. See Chapter 6 for further information about configuration restart.

COLD

instructs ACF/VTAM to restore each major node to its initial status as defined by the user. ACF/VTAM issues the equivalent of VARY NET,ACT,ID=major node name, COLD for each major node identified by the CONFIG start option.

WARM

instructs ACF/VTAM to issue the equivalent of VARY NET,ACT,ID=major node name,WARM for each major node identified by the CONFIG start option.

When WARM is specified during ACF/VTAM startup, ACF/VTAM uses the contents of VSAM configuration restart data sets to restore each major node to its status at the time of failure or deactivation. The network operator can therefore restart a subset of the configuration with one command instead of issuing an individual VARY command for each major node. You define VSAM configuration restart data sets and associate them with major nodes as described in "Delayed Configuration Restart" in Chapter 6.

If a major node identified by the CONFIG start option does not have an associated configuration restart data set or has an associated configuration restart data set that has not been used previously by ACF/VTAM, the major node is activated to its initial status. ACF/VTAM also issues an informational message. An operator-issued VARY command fails if the WARM option is specified for a major node that does not have an associated configuration restart data set.

ACF/VTAM ignores the COLD and WARM start options for application program major nodes and for major nodes that are not associated with configuration restart data sets.

Refer to the descriptions of the activation of specific major nodes *ACF/VTAM Network Operating Procedures* for information on the effect of the COLD and WARM options.

The CONFIG Start Option

CONFIG=xx|00|name

specifies a list of major nodes to be activated when ACF/VTAM is started. This option can be specified within a predefined list of start options named by the LIST start option or by the network operator when starting ACF/VTAM.

xx

is any 1 or 2 alphanumeric characters that identify the member of SYS1.VTAMLST that contains a list of major nodes to be activated when ACF/VTAM is started. You can file the list of major nodes under the name ATCCONxx as described in this chapter under "Filing Start Options." More than one list of major nodes can be filed to give the user a choice of configurations and to avoid having to issue a separate VARY command to activate each major node.

00

is the default value for the CONFIG start option. A user-defined configuration list of major nodes can be filed under the name ATCCON00. ACF/VTAM always uses this predefined list unless the network operator uses the CONFIG option to specify another list or the CONFIG option is included in the set of LIST start options. If a default configuration list does not exist when ACF/VTAM is started, an error message is sent to the network operator. ACF/VTAM initialization continues without the configuration list.

name

specifies the 3 to 8 character data definition name of a configuration restart VSAM data set containing a list of major nodes that were active at the time of failure or deactivation. ACF/VTAM must have previously used the data set to record a list of active major nodes; that is, the name of the data set must have been specified in the NODELST option during some previous execution of ACF/VTAM.

Include a DD statement using this data definition name in the ACF/VTAM start procedure.

If the data set is empty, ACF/VTAM does not activate any major nodes during startup. The network operator is prompted for a password if one is required to gain access to the data set.

The DLRTCB Start Option

DLRTCB=n

specifies the greatest number of task control blocks (TCBs) to be used by dump-load-restart subtasks. Specify n as a decimal number in the range 1 through 64. If 0 is specified, ACF/VTAM uses the default value. If a value greater than 64 is specified, ACF/VTAM issues an error message and prompts the network operator to re-enter the DLRTCB value.

DEFAULT: For OS/VS1 and OS/VS2 SVS, the default value is DLRTCB=8. For OS/VS2 MVS, the default value is DLRTCB=32.

The default value is large enough for most configurations and should not be exceeded unless there are very few tasks and subtasks concurrently in the job stream and the tuning statistics trace (TNSTAT) indicates a great demand for dump-load-restart TCBs. See Chapter 7 for more information on the tuning statistics trace.

The HOSTSA Start Option

HOSTSA=n | 1

is a decimal number chosen by the user to identify this host ACF/VTAM's subarea. This start option must be coded for any ACF/VTAM that engages in cross-domain communication. Specify n as an integer in the range 1 through the MAXSUBA value. When this host ACF/VTAM is active, no other host ACF/VTAM, NCP major node, or local SNA major node that has the same subarea number can also be active.

The LIST Start Option

LIST=yy | 00

indicates which list of predefined start options (identified by the last two alphanumeric digits in ATCSTRyy) is used to initialize a specific network. Each ATCSTRyy list is a member in SYS1.VTAMLST, and the user can specify any two alphanumeric digits (for yy) to identify user-defined lists of start options.

LIST=yy can be entered only by the network operator when starting ACF/VTAM. LIST=00, identifying the default start list ATCSTR00, cannot be entered by the network operator. LIST=00 can be included only in ATCSTR00 and ATCSTRyy. If the network operator enters more than one LIST start option, ACF/VTAM processes only the last LIST start option that was entered.

The MAXAPPL Start Option

MAXAPPL=n | 10

specifies the greatest number of access method control blocks that are to be open concurrently. ACF/VTAM uses this value to allocate fixed storage for control of application programs. Specify n as a decimal number in the range of 1 up to the maximum MAXAPPL value. The maximum MAXAPPL value is related to the MAXSUBA value, as shown by the following table.

MAXSUBA Value	Maximum MAXAPPL Value
3	16378
4 through 7	8186
8 through 15	4090
16 through 31	2042
32 through 63	1018
64 through 127	506
128 through 255	250

If the number specified by MAXAPPL exceeds the maximum value, ACF/VTAM issues an error message and prompts the network operator to correct the value. If the number specified by MAXAPPL is too small in proportion to the number of ACBs to be opened, the opening of some ACBs is unsuccessful because there is not enough storage available. If the number specified by MAXAPPL is too large in proportion to the number of ACBs to be opened, an excessive amount of fixed storage is reserved for controlling application programs.

If the MAXAPPL operand is not coded, MAXAPPL=10 is assumed by default.

The MAXSUBA Start Option

MAXSUBA=n | 15

indicates a decimal integer in the range 3 through 255. The number specifies, for the network configuration being activated, the highest SUBAREA value that can be assigned to a major node ID, and the maximum number subareas that can be active at one time.

The following example illustrates how to select a MAXSUBA value. Assume that these major nodes are being installed:

A host ACF/VTAM

HOSTSA=1

Two local non-SNA major nodes

LBUILD SUBAREA=2

LBUILD SUBAREA=3

Two switched SNA major nodes (which do not take a SUBAREA value)

A local NCP

BUILD SUBAREA=4

Two remote NCPs

BUILD SUBAREA=5

BUILD SUBAREA=9

In this configuration, six major nodes have SUBAREA values. A MAXSUBA value of at least 9 is required, however, because the highest SUBAREA value is 9.

If the SUBAREA value of one of the major nodes is arbitrarily set to 50, a minimum MAXSUBA value of 50 is required. Selecting a MAXSUBA value greatly in excess of the required minimum severely limits the number of minor nodes that can be contained in each major node. The number of minor nodes permitted in each major node is governed by the MAXSUBA value, as shown in Figure 3-1.

A minor node (in Figure 3-1) attached to an NCP is defined by each COMP, LINE, TERMINAL, LU, PU, and CLUSTER (but not GROUP) macro instruction; a minor node (terminal) for a local terminal configuration is defined by each LOCAL definition statement. The third column of Figure 3-1 shows that as the value of MAXSUBA increases, the number of minor nodes in each major node decreases.

Range of MAXSUBA Values	Range of SUBAREA Values	Maximum Number of Minor Nodes for Each Major Node	Maximum Number of Possible Major Nodes
3	2-3	16,381	2
4-7	2-7	8,089	6
8-15	2-15	4,093	14
16-31	2-31	2,045	30
32-63	2-63	1,021	62
64-127	2-127	509	126
128-255	2-255	253	254

Figure 3-1. MAXSUBA Values

On the other hand, if the value of MAXSUBA is too low (such as the absolute minimum of 9 in this example), the MAXSUBA value must be increased if the user adds more major nodes. The MAXSUBA start option must be changed, all the NCPs containing the MAXSUBA value have to be changed and regenerated.

The MAXSUBA value can be selected as follows:

1. Determine the minimum required MAXSUBA value (in this example, 9).
2. Locate the row in Figure 3-1 that contains this MAXSUBA value (in this example, the third row contains the MAXSUBA range of 8-15, which encompasses the MAXSUBA value of 9).
3. Use the highest MAXSUBA value of the next row (in this example, 31).

The value selected for MAXSUBA is also specified for the MAXSUBA start option of each NCP's BUILD macro instruction. These MAXSUBA values need not necessarily be identical, but they must all be within the same MAXSUBA range (as shown in the first column of Figure 3-1) and they must all be at least equal to the highest SUBAREA value in the network.

The MAXSUBA relationship is due to ACF/VTAM's internal addressing structure. ACF/VTAM uses a 16-bit field to form the address of each minor node. The field has two parts. The first part identifies the major node and is formed directly from the binary SUBAREA value selected for that major node; the second part identifies the minor node and is a binary value derived from the relative position of the minor node's definition statement. The sizes of the two parts are not fixed (although the total is always 16 bits).

If MAXSUBA=3, meaning that only 2 bits are required for the major node part, the remaining 14 bits are available for the minor node part; over 16,000 minor nodes can be represented for each major node. Because a higher MAXSUBA value requires more bits, fewer bits are available to represent minor nodes. Note that the ranges shown in the first column of Figure 3-1 correspond to the values represented by a given number of bits.

Figure 3-1 shows (1) the valid values that can be assigned to the MAXSUBA start option, (2) the range of valid SUBAREA values and (3) the number of minor nodes that can be defined to each major node.

These limits apply if the network contains both SDLC terminals and BSC or start-stop terminals. If the network contains only SDLC or only BSC and start-stop terminals, add 1 to each limit.

The value specified in the MAXSUBA operand (in the BUILD macro instruction) and the SUBAREA values (specified in the SUBAREA operands in the BUILD macro instructions and the LBUILD and VBUILD definition statements) must correspond to the value in the MAXSUBA start option to the extent indicated in Figure 3-1.

If the MAXSUBA value is changed and a previously defined major node (created when the old value was in effect) is used with the new MAXSUBA value, its RDT must be deleted from SYS1.VTAMOBJ before the node is activated so that a new RDT can be created with the correct values in it.

For each NCP that is active at the same time, the value specified in the MAXSUBA operand (in the BUILD macro instruction) must be the same and must correspond to the value in the MAXSUBA start option.

The value in the MAXSUBA start option must be 3, 7, 15, 31, 63, 127, or 255. If the value assigned to MAXSUBA is not one of these values, ACF/VTAM rounds the specified value to the next higher such value. (For example, any value specified from 16 to 30 is rounded to 31.)

In a multidomain network, MAXSUBA must be in the same range for each domain that is to engage in cross-domain communication.

The NETSOL Start Option

NETSOL=YES | NO

indicates whether the ACF/VTAM logon monitor facility (referred to as the network solicitor), for devices that use basic mode, is to be attached as a subtask when ACF/VTAM is started.

YES

indicates that ACF/VTAM is to attach either the IBM-supplied network solicitor or a modified network solicitor during ACF/VTAM initialization.

For this start option to apply to a modified network solicitor, the modified one must be named NETSOL and link-edited as load module ISTNSC00 in SYS1.VTAMLIB (replacing the IBM-supplied network solicitor).

The command MODIFY NETSOL=YES or NETSOL=NO can also be used to start and stop either the IBM-supplied network solicitor or a modified network solicitor.

NO

indicates that ACF/VTAM is not to attach a network solicitor during ACF/VTAM initialization.

The NODELST Start Option

NODELST=name

specifies the data definition name of a VSAM data set in which ACF/VTAM is to maintain a list of all currently active major nodes.

After the halting of the host ACF/VTAM or the failure of the host operating system, the host computer, or the host ACF/VTAM, the network operator can reactivate the major nodes that were active at the time of deactivation or failure by specifying CONFIG=name in the ACF/VTAM start procedure, where 'name' is the NODLEST name that was in effect when ACF/VTAM was last running.

The network operator is prompted for a password if one is required.

If NODELST and CONFIG specify the same name when ACF/VTAM is started, ACF/VTAM attempts to activate the major nodes listed in the configuration restart data set. This same data set is then used as the NODELST data set, and is updated when major nodes are activated or deactivated.

If the name specified in NODELST is different than the name specified in CONFIG when ACF/VTAM is restarted, the data set identified by NODELST is erased before any major nodes are activated.

The PROMPT Start Option

PROMPT | NOPROMPT

PROMPT and NOPROMPT can be coded only in the ATCSTR00 predefined list of start options. PROMPT specifies that ACF/VTAM is to prompt the network operator to enter ACF/VTAM start options if the network operator does not enter them on the START command. If NOPROMPT is specified, ACF/VTAM does not prompt the network operator. A specification of NOPROMPT in ATCSTR00 cannot be overridden by specifying PROMPT on the START command or by including it in a predefined list. Do not specify NOPROMPT unless you are sure that the network operator will never want to be prompted by ACF/VTAM.

The SUPP Start Option

SUPP=<u>NOSUP</u> | INFO | WARN | NORM | SER

specifies the highest class of ACF/VTAM error messages for which ACF/VTAM suppresses printing at the network operator console and suppresses transmission to a program operator, if one exists. After ACF/VTAM has been started, the network operator can use the SUPP operand of the MODIFY command to suppress messages. If multiple console support (MCS) has been included in the system, all suppressed messages are sent to the hard copy log.

The SUPP start option and the SUPP operand of the MODIFY command are identical in form and effect.

NOSUP

specifies that all ACF/VTAM messages are to be printed at the console (NOSUP indicates "no suppression").

INFO

specifies that informational messages are to be suppressed. Informational messages are those that tell the operator that commands or procedures have been accepted for processing.

WARN

specifies that warning messages (as well as informational messages) are to be suppressed. Warning messages indicate error conditions that do not cause commands to fail or be rejected. These messages tell the operator that there is a problem, such as an invalid operand or a minor node that cannot be activated, but that ACF/VTAM can continue to process other parts of the command or procedure.

NORM

specifies that normal completion messages (as well as informational and warning messages) are to be suppressed. Normal completion messages tell the operator that commands have completed processing successfully, that a configuration has been activated successfully, that a procedure has been terminated, and so on.

SER

specifies that serious error messages (as well as informational, warning, and normal completion messages) are to be suppressed. Serious error messages indicate error conditions that cause commands or procedures to fail. These messages tell the operator that a command must be reentered or a procedure reinitiated.

Error messages that indicate an even more serious situation, such as the abnormal termination of a user task or of ACF/VTAM itself, cannot be suppressed. Messages that are generated in response to an operator request (such as the DISPLAY command) and messages that require a response (prompting messages) also cannot be suppressed.

The TNSTAT Start Option

```
{ TNSTAT [,CNSL |NOCNSL] [,TIME=n |60] }
{ NOTNSTAT                             }
```

TNSTAT
specifies that tuning statistics should be kept.

NOTNSTAT
specifies that tuning statistics should not be kept.

If neither TNSTAT nor NOTNSTAT are specified, NOTNSTAT is assumed. If NOTNSTAT is specified or assumed by default, the MODIFY TNSTAT command will not be accepted by ACF/VTAM.

The following operands are valid only if TNSTAT is specified.

CNSL
specifies that the tuning statistics records are to be written to the console as well as to the System Management Facilities file.

NOCNSL
specifies that the tuning statistics records are to be written only to the System Management Facilities file.

If neither CNSL nor NOCNSL are specified, NOCNSL is assumed.

TIME=n
specifies the number of minutes that should elapse between records. One record is written every n minutes for each local SNA controller in the network. A record is also written when a controller is deactivated or when a MODIFY TNSTAT or MODIFY NOTNSTAT command is issued.

The value for TIME must be in the range 1 to 1440. If TIME is not specified, TIME=60 is assumed.

The TRACE Start Option

```
                       { nodename,TYPE=BUF |IO| LINE }
TRACE | NOTRACE,ID=    { VTAMBUF,TYPE=SMS            }
```

TRACE
indicates that ACF/VTAM is to start a specific type of trace for a specific node or for monitoring the usage of all the ACF/VTAM buffer pools. Once started, the trace remains in effect until it is stopped either by stopping ACF/VTAM or by the network operator entering the MODIFY command with the NOTRACE operand. More than one trace can be in effect at the same time; however, a separate TRACE start option must be specified to start each trace.

NOTRACE
indicates that ACF/VTAM is to cancel the trace specified in the ID and TYPE suboptions. This start option should only be specified when it is necessary to override a predefined TRACE start option. A separate TRACE start option must be specified to stop each trace.

ID=nodename

indicates the specific node for which an ACF/VTAM trace is to be started or stopped. If it is a buffer or I/O trace (TYPE=BUF or TYPE=IO respectively), then nodename corresponds to the name assigned to a terminal, logical unit, or application program (when the major and minor nodes were put in SYS1.VTAMLST by the IEBUPDTE utility program). ID=VTAM can be specified to trace all SSCP activity and, if the Multisystem Networking Facility is installed, all cross-domain resource manager activity.

If it is a line trace (TYPE=LINE), then nodename corresponds to the symbol on the LINE macro instruction that represents the same line.

Each terminal or logical unit to be traced must be explicitly specified in a TRACE start option.

TYPE=BUF | IO | LINE

indicates the specific type of ACF/VTAM trace that is to be started or stopped for a node. Only one type of ACF/VTAM trace can be specified in each TRACE start option (whether specified by the network operator or predefined in a list of start options). In the TYPE suboption, BUF is specified for an ACF/VTAM buffer trace, IO for an ACF/VTAM I/O trace, or LINE for an NCP line trace. There can be up to 8 concurrent line traces for each NCP, depending on how each NCP was generated.

ID=VTAMBUF,TYPE=SMS

indicates that ACF/VTAM is to monitor the number of ACF/VTAM requests to obtain buffers in the ACF/VTAM buffer pools. (After a specified number of requests occur, the trace creates a record to show how the pools were being used at that time.) When specifying TYPE=SMS, you must also specify ID=VTAMBUF. For more information, refer to *ACF/VTAM Debugging Guide.*

The VTAMEAS Start Option

VTAMEAS=n | 404

specifies the maximum number of concurrently active SNA network addressable units. ACF/VTAM uses this operand in a rapid-lookup scheme to find a representation of a session between the host ACF/VTAM and a SNA network addressable unit.

Specify n as a decimal number in the range 0 through 32767.

Code this operand if the ACF/VTAM system is to have one or more active SNA network addressable units. Specify a number that is 10% to 20% greater than the total number of SNA network addressable units that you expect to be active at one time. One way to estimate the maximum number of concurrently active SNA network addressable units is to take the sum of the number of PCCU macro instructions and the number of LOCAL, LU, and PU statements used to define the network to ACF/VTAM. If the actual number of active network addressable units in the ACF/VTAM system is greater than the number specified in VTAMEAS, or if the actual number is greater than 8080, the ACF/VTAM path length is increased.

If the VTAMEAS operand is not coded, VTAMEAS=404 is assumed.

Filing Start Options

Filing start options reduces the amount of data that has to be typed in by the network operator when starting ACF/VTAM. This reduces the chance of keying errors and permits quick establishment of reconfiguration of the network. Refer to *ACF/VTAM Concepts and Planning* for considerations in setting up starting procedures.

Two types of start option sets can be filed, sets of major nodes referred to by the CONFIG start option and sets of start options, including the CONFIG start option, referred to by the LIST start option.

Creating Start Option And Configuration Lists

If a start option list is to be used, the list must have been created and put in SYS1.VTAMLST under the member name ATCSTRyy. A member named ATCSTR00 must be filed. If no start options are to be filed in ATCSTR00, file a record of blanks. Whether or not LIST=yy is entered as a start option by the network operator, ACF/VTAM attempts to locate ATCSTR00. If it does not exist, ACF/VTAM sends an error message to the network operator, followed by a message prompting the operator for start options.

If a configuration list is specified either in a start option list or by the network operator, this optional list of major node names must have been created and put in the source statement library under the book name B.ATCCONxx. SYS1.VTAMLST under the member name ATCCONxx. Also, the same major node name can be specified in more than one configuration list. For example, a group of local terminals named LOCCON01 can be in ATCCON00 and ACCON06. However, LOCCON01 can only be activated once during ACF/VTAM initialization.

The major node names in the configuration list correspond to the names assigned to the major nodes when they are defined and put into SYS1.VTAMLST using the IEBUPDTE utility program. If no configuration lists exist, the network operator must use the VARY command to activate each major node.

ACF/VTAM definition statements can be filed during the same execution of IEBUPDTE used for filing start options. For information on IEBUPDTE, refer to *OS/VS Utilities.*

Format Of Sets Of Start Options

Code a set of start options as one or more 80-byte records in this format:

```
Card                      Card
Column                    Column
1                         71
[b̸ . . .] item[,item] . . .
```

[b̸ . . .] signifies that one or more blanks may precede the first item.

item represents (1) a major node name if a configuration list is being coded or (2) a start option if a list of start options is being coded.

It is good practice to list APPL major node names first. This prevents the messages that are issued to the operator if LOGAPPL operands are coded in LOCAL, PU, or LU statements or NCP macro instructions and the specified application program has not yet been activated.

If any item is the name of a remote NCP, that major node name must follow the local NCP major node name to which the remote communications controller is attached. This is because major nodes are activated during ACF/VTAM initialization in the order in which their names appear in the list.

In a multidomain network, list the NCP major nodes before the CDRM major nodes so that the paths over which the CDRM sessions must flow will be active before any attempt is made to start a session.

Separate items with commas. There must be no intervening blanks.

Column 72 is reserved for the continuation indicator. Columns 73 through 80 should not be used at all.

To continue onto another record, place a comma after the last item in the current record, enter a nonblank character in column 72 of the current record, and start the next item in any column of the new record. If desired, the current record can be coded through column 71, a nonblank character entered in column 72, and the item continued in any column of the new record.

To create a comment card, enter an asterisk (*) in column 1 of any card other than a continuation card.

Examples Of Using Start Option And Configuration Lists

This section shows examples of major nodes to be filed for access by specifying CONFIG=00 and CONFIG=06, respectively. It also shows examples of start options to be filed for access by the automatically processed start option, LIST=00, and also by LIST=ST.

Example 1

Card image to be filed as ATCCON00

```
A1,NCPL1,L3270A,SW1
```

This is the default configuration. If the CONFIG start option is not specified when ACF/VTAM is started, these major nodes are activated: a set of application programs named A1, a local NCP named NCPL1, a set of local non-SNA terminals named 3270A, and a set of switched SNA terminals named SW1.

Example 2

Card image to be filed as ATCCON06

```
A2,NCPL1,NCPR1,L3790A,L3790B,SW1,SW2
```

When CONFIG=6 is specified by the network operator, or is processed in a predefined set of start options, these major nodes are activated: a set of application programs named A2, a local NCP named NCPL1, a remote NCP named NCPR1, two sets of local 3790 terminals named L3790A and 3790B, and two sets of switched SNA terminals named SW1 and SW2. Note that the application program major node name was coded first and that the local NCP major node name was coded before the remote NCP major node name.

Example 3

Card image to be filed as ATCSTR00

```
MAXSUBA=5,NETSOL=YES,. . .
```

This is the automatically-initiated default start-option list. For start options not included here or entered by the network operator, default values are assumed by ACF/VTAM, including CONFIG=00 (see Example 1).

Example 4

Card image to be filed as ATCSTRST

```
CONFIG=06,MAXSUBA=7,NETSOL=YES,...
```

This set of start options is used when the network operator enters LIST=ST. Additional start options, or overriding start options, can also be entered by the network operator. The configuration defined in Example 2 is activated when ACF/VTAM is started.

Overriding Start Options

When ACF/VTAM is started, start options can be provided with any combination of the following methods:

IBM-supplied values are used if they are not provided in one of the ways listed here.

No start options need be explicitly entered. Start options are secured automatically from ATCSTR00, which is always processed. ATCSTR00 can be used to specify a predefined list of start options.

Start options (other than PROMPT or NOPROMPT) can be entered by the network operator in the start procedure. The LIST start option can be entered only at the console. It can be entered along with an overriding operand for any operand in the list, and additional operands not specified in the list, including operands to override the operands in the predefined list.

The prompting facility can be used only in ATCSTR00 to request start options from the network operator.

The network operator is notified if there was an error in a previously entered start option. ACF/VTAM prompts the network operator to reenter the start option. If ACF/VTAM detects an error in ATCSTRyy, the network operator is notified and the operator must reenter all start options that were entered as part of the start procedure.

The overriding priorities of the various ways of entering start options, arranged from the highest priority to lowest priority, are listed below:

Options reentered by the network operator if ACF/VTAM finds an error in previously entered start options

Options entered by the network operator (except for LIST) as part of the start procedure

Options listed in ATCSTRyy, used when the network operator enters the LIST start option

Options filed in ATCSTR00

IBM-supplied values

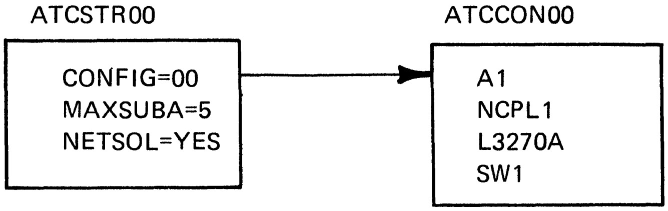
If options conflict, the highest priority option overrides the lower priority option. For example, if ATCSTR00 specifies CONFIG=01, but the operator enters LIST=ST and ATCSTRST contains the option CONFIG=02, the configuration defined by ATCCON02 is activated.

Any or all of the start options can be filed together in a single member, in which case the LIST start option refers to a file of start options. As noted above, the member referred to by LIST=00 (ATCSTR00) is automatically processed when ACF/VTAM is started and must exist prior to starting ACF/VTAM.

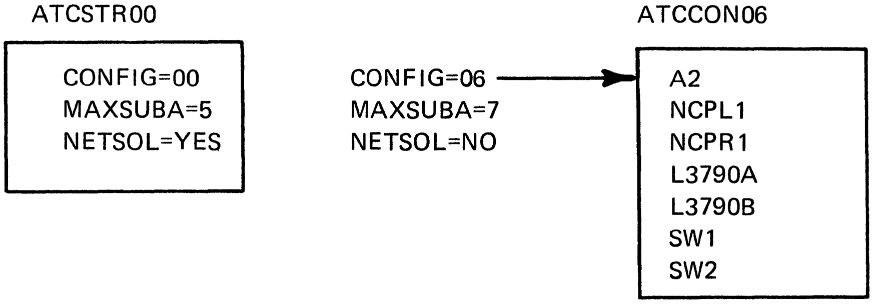
Figure 3-2 contains examples of the start procedure that show how CONFIG and LIST can be used.

Refer to the examples in "Examples of Using Start Option and Configuration Lists" in this chapter.

Example A. The network operator enters the START command with no start options. The automatically processed member ATCSTR00 contains a set of start options including CONFIG=00. The result is:



Example B. The network operator enters the START command with the following start options: CONFIG=06,MAXSUBA=7,NETSOL=NO CONFIG=06 refers to ATCCON06, which contains a list of major node names. The MAXSUBA and NETSOL options override those options filed in ATCSTR00. The result is:



Example C. The network operator enters the START command with the start option LIST=ST. LIST=ST refers to ATCSTRST, which contains a sets of start options. The same configuration and start options are obtained as in Example B. The result is:

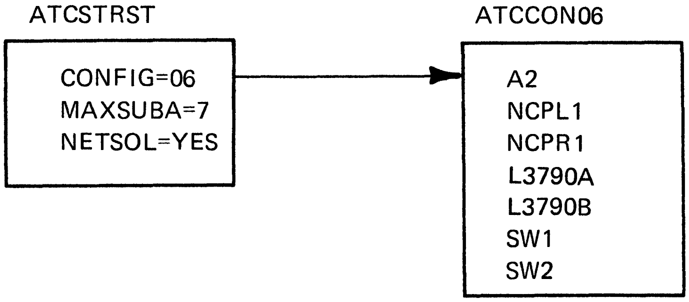


Figure 3-2. Examples of Using CONFIG and LIST Start Options

Chapter 4. Defining Connection And Disconnection Procedures

Before an application program and a terminal can communicate with each other, a connection must be established between them. Connection is the process by which ACF/VTAM establishes an association between the internal control blocks that represent a terminal and an application program. The existence of a connection means that a physical path is available for the transfer of data between the application program and the terminal and that ACF/VTAM is ready to handle the transfer. For terminals attached by switched lines, for example, connection includes a dial-in or dial-out operation. For SNA terminals, connection includes an exchange of commands between ACF/VTAM and the physical and logical units (in SNA terms, SSCP-physical unit, SSCP-logical unit, and logical unit-logical unit sessions are established).

A request by or on behalf of a terminal to be connected to an application program is called a logon. Four types of logons are defined and explained in *ACF/VTAM Concepts and Planning* in terms of the source of the logon:

Automatic logon, in which ACF/VTAM automatically logs a terminal on to an application program (the terminal's definition statements indicate whether this is to be done)

Terminal-initiated logon, in which the terminal (which can be an application program) enters a message that causes it to be logged on to a selected application program

Application program logon, in which one application program passes a terminal (that is already connected to it) to another application program, simulates a logon from a terminal, or acquires a terminal

Network operator logon, in which the network operator can cause a specified terminal to be logged on to a specified application program

The network operator VARY command overrides any automatic logon designation for a terminal. The new automatic logon specification remains in effect until overridden by another VARY command with the LOGON option, or until the major node containing the terminal is deactivated and reactivated with the COLD option specified. That is, if the terminal is deactivated and then reactivated (with no intervening deactivation of the major node), the new logon specification is still in effect. If a major node is deactivated and later reactivated with the COLD option specified, the automatic logon conditions specified at ACF/VTAM definition are in effect.

Logon Modes And Session Parameters

The immediate effect of a logon is to notify the application program that a terminal is "waiting" for the application program to establish connection and begin communicating with it. When an application program establishes connection with a logical unit (as contrasted with a BSC or start-stop terminal, discussed later), the application program tells ACF/VTAM and the logical unit how the communication session is to be conducted.

The information about the communication session is contained in a field that describes a set of session parameters. These parameters have meanings such as "the application program will send chained data" or "the logical unit will not send end-of-bracket commands to the application program". For more information on session parameters, see *ACF/VTAM Macro Language Guide* and *ACF/VTAM Macro Language Reference.*

You can define a logon mode name that identifies a set of session parameters. To associate a logon mode name with a particular set of session parameters, ACF/VTAM uses a logon mode table, as illustrated in Figure 4-1.

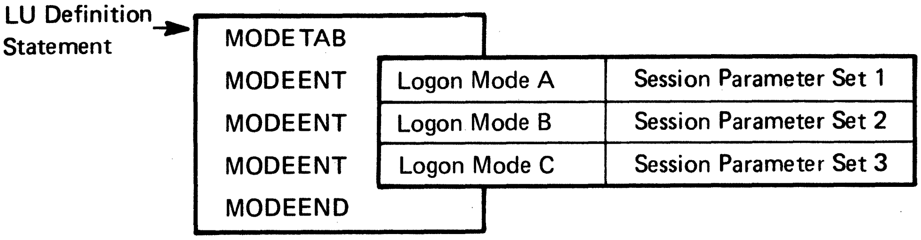


Figure 4-1. Logon Mode Table Macro Instructions

Each logon mode table can contain more than one logon mode name and associated set of session parameters.

ACF/VTAM cannot be started unless a logon mode table named ISTINALM (for OS/VS1 and OS/VS2 SVS) or ISTINCLM (for OS/VS2 MVS) is available to ACF/VTAM. To meet this requirement, a suitably named IBM-written logon mode table is supplied as part of ACF/VTAM.

Figure 4-2 shows the macro instructions that make up the IBM-supplied logon mode table.

Except for its name, this table is exactly the same for all OS/VS systems.

The IBM-supplied logon mode table can be modified or replaced by the user, provided that the modified table or replacement table has the same name as the IBM-supplied table and that the IBM-supplied table is deleted.

In addition to the required logon mode table just discussed, you can create and name optional logon mode tables with MODETAB, MODEENT, and MODEEND macro instructions and then assemble and link edit the tables to SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS). A logon mode table can be associated

```
ISTINCLM  MODETAB
IBM3767   MODEENT  LOGMODE=INTERACT,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B1',SECPROT=X'A0',COMPROT=X'3040'
IBM3770   MODEENT  LOGMODE=BATCH,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'A3',SECPROT=X'A3',COMPROT=X'7080'
IBMS3270  MODEENT  LOGMODE=S3270,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000'
IBM3600   MODEENT  LOGMODE=IBM3600,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'F1',SECPROT=X'F1',COMPROT=X'7000'
IBM3650I  MODEENT  LOGMODE=INTRACT,FMPROF=X'00',TSPROF=X'04',PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'6000'
IBM3650U  MODEENT  LOGMODE=INTRUSER,FMPROF=X'00',TSPROF=X'04',PRIPROT=X'31',SECPROT=X'30',COMPROT=X'6000'
IBMS3650  MODEENT  LOGMODE=IBMS3650,FMPROF=X'00',TSPROF=X'04',PRIPROT=XB0',SECPROT=X'90',COMPROT=X'4000'
IBM3650P  MODEENT  LOGMODE=PIPELINE,FMPROF=X'00',TSPROF=X'03',PRIPROT=X'30',SECPROT=X'10',COMPROT=X'0000'
IBM3660   MODEENT  LOGMODE=SMAPPL,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'A0',SECPROT=X'A0',COMPROT=X'0081'
IBM3660A  MODEENT  LOGMODE=SMSNA100,FMPROF=X'00',TSPROF=X'00',PRIPROT=X'00',SECPROT=X'00',COMPROT=X'0000'
          MODEEND
```

Figure 4-2. The IBM-Supplied Logon Mode Table

with one or more logical units by specifying the name of the table in the MODETAB operands of the logical units' definition statement. Many different logical units can specify the same logon mode table.

If you do not designate a logon mode table for a logical unit, the IBM-supplied table (or its user-written replacement) is used. A summary of the method used to obtain session parameters from a logon mode table is shown in Figure 4-3.

The source of the logon mode name in a logon during program execution is not necessarily the logical unit, but depends on the source of the logon itself. A logon mode name also can be supplied by the network operator, by ACF/VTAM (as part of the automatic logon process), or by an application program.

Regardless of the source of the logon, the disposition of the session parameters derived from the logon mode name is up to the application program. The application program can issue an INQUIRE macro instruction (1) to obtain the session parameters associated with the pending logon, or (2) to submit a logon mode name of its own selection and obtain the session parameters that are associated with that name in the logon mode table for that logical unit. When the application program issues an OPNDST macro instruction to establish connection with the logical unit, the application program indicates the logon mode name or the session parameters to be used during the communication session. Specifically, the application program issuing OPNDST can:

Supply a logon mode name whose session parameters are known

Directly supply the session parameters to be used

Use the session parameters associated with the pending logon (without ever examining the session parameters)

Use the default session parameters that you have specified for the logical unit

The *ACF/VTAM Macro Language Guide* and *ACF/VTAM Macro Language Reference* describe how the application program handles session parameters.

The MODETAB Macro Instruction

The MODETAB macro instruction indicates the beginning of a logon mode table and is coded as follows:

Name	Operation	Operand
[name]	MODETAB	

name

if coded, is used as a CSECT name for the logon mode table to be generated. This macro instruction has no operands.

The MODEENT Macro Instruction

A MODEENT macro instruction associates a logon mode name with a set of session parameters. To determine what values must be coded for this macro instruction's operands you must first be familiar with the features that are available on the devices in your network, and you must have decided how you want to use them. To find information on the features available for a particular device, see the appropriate system programmer's guide or component description for the device. This information will help you decide

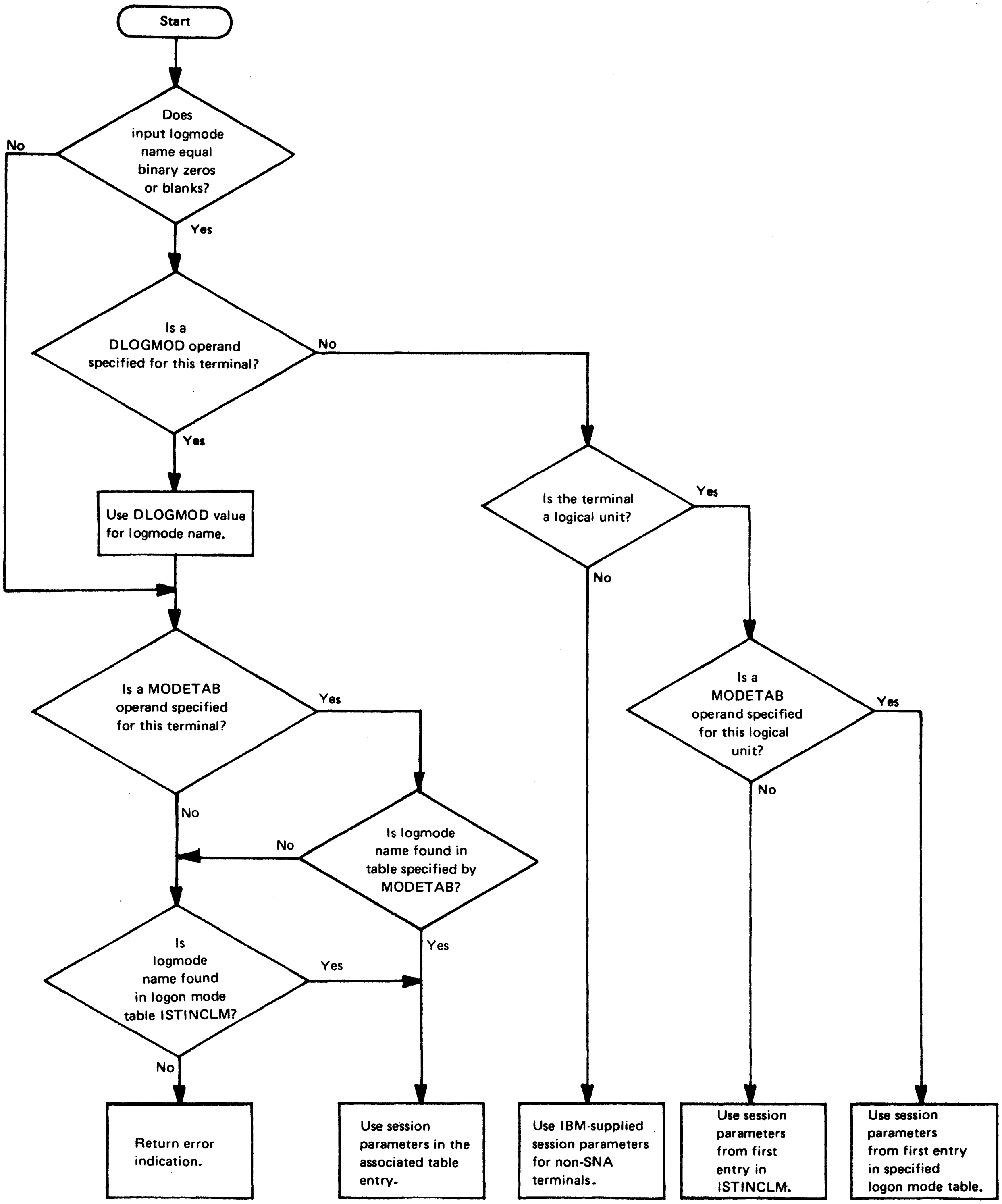

Start
Does input logmode name equal binary zeros or blanks?
No
Yes
Is a DLOGMOD operand specified for this terminal?
No
Yes
Use DLOGMOD value for logmode name.
Is the terminal a logical unit?
Yes
No
Is a MODETAB operand specified for this terminal?
Yes
No
Is a MODETAB operand specified for this logical unit?
Yes
No
Is logmode name found in table specified by MODETAB?
No
Yes
Is logmode name found in logon mode table ISTINCLM?
Yes
No
Return error indication.
Use session parameters in the associated table entry.
Use IBM-supplied session parameters for non-SNA terminals.
Use session parameters from first entry in ISTINCLM.
Use session parameters from first entry in specified logon mode table.

Figure 4-3. Search Logic for the Logon Mode Table

which session parameters are most appropriate to your needs. Then consult the section in the *ACF/VTAM Macro Language Reference* that describes the session parameter fields. It tells you what bits must be set to obtain the appropriate session parameters. If the *ACF/VTAM Macro Language Reference* identifies a bit as being reserved, set that bit to 0.

Once you have determined what bits must be set, convert the eight bits in each byte of the session parameter field into the two equivalent hexadecimal digits. The value for each operand is the hexadecimal equivalent of the bits in the byte with which the operand is associated. For example, the operand PRIPROT is associated with byte 3 of the session parameter field. If you decide (for example) that the appropriate bit settings for this byte are 1111 0001, then code PRIPROT=X'F1'.

For the FMPROF and TSPROF operands, 'value' can be specified as unframed decimal digits (for example, 5) or as framed hexadecimal digits (for example, X'04'). The other operands require hexadecimal values.

The MODEENT macro instruction is coded as shown below and follows the MODETAB or another MODEENT macro instruction.

Name	Operation	Operand
[name]	MODEENT	[LOGMODE=name] [,FMPROF=value] [,TSPROF=value] [,PRIPROT=value] [,SECPROT=value] [,COMPROT=value] [,RUSIZES=value] [,PSERVIC=value] [,PSNDPAC=value] [,SRCVPAC=value] [,SSNDPAC=value] [,ENCR=value] [1]

[1] For the ACF/VTAM Encrypt/Decrypt Feature only.

name
is optional; it has no function in the specification of a logon mode table.

LOGMODE=name
specifies the logon mode name to be used as a key for the session parameters in this table entry. If duplicate names appear in the table, the first occurrence of the name is used. If LOGMODE is omitted, eight blanks are used.

FMPROF=value
represents the function management profile (byte 1 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0 to FF (or its equivalent decimal value) can be specified for value. Values 2, 3, and 4 have defined meanings that are described in *ACF/VTAM Macro Language Reference.* The default value is 0.

TSPROF=value
represents the transmission services profile (byte 2 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0 to FF (or its equivalent

decimal value) can be specified for value. Values 2, 3, and 4 have defined meanings that are described in *ACF/VTAM Macro Language Reference.* The default value is 0.

PRIPROT=value
represents the primary logical unit protocol (byte 3 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0 to FF can be specified for value. The default value is 0.

SECPROT=value
represents the secondary logical unit protocol (byte 4 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0 to FF can be specified for value. The default value is 0.

COMPROT=value
represents the common logical unit protocols (bytes 5 and 6 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0 to FFFF can be specified for value. The default value is 0.

RUSIZES=value
represents a portion of the transmission services usage field (bytes 9 and 10 in the session parameter field) for this logon mode; it specifies the maximum length of data (request units) in bytes that can be sent by the primary logical unit and by the secondary logical unit with which the primary logical unit communicates in this session.

Specify RUSIZES as 4 hexadecimal digits. The leftmost 2 digits applies to the secondary logical unit, and the rightmost 2 digits applies to the primary logical unit. The format is the same for both sets of digits. The first digit is the mantissa (m) and the second digit is the exponent (n) in the formula m X 2^n, from which is calculated the maximum length of data that can be sent by the primary or secondary logical unit. For example, RUSIZES=X'96A8' specifies that the secondary logical unit can send a maximum length of 9 X 2^6 (or 576) bytes and that the primary logical unit can send a maximum of 10 X 2^8 (or 2560) bytes. The digit representing the mantissa must be in the range 8 to F while the digit representing the exponent must be in the range 0 to F. If both the mantissa and exponent are set to zero or if RUSIZES is not specified, the default size is used. For byte 9 (secondary logical unit send size), the default size is 6K bytes*. For byte 10 (primary logical unit send size), the default indicates there is no limit on the size of the RU that may be sent. **Note:** *This default size is set in the ACTVT by the ACF/VTAM module ISTTSCIA during ACF/VTAM start-up processing.*

PSERVIC=value
represents the logical unit presentation services profile (bytes 13 through 24 in the session parameter field) for this logon mode. Code a 24 digit hexadecimal number as the value, using the bit settings described in *ACF/VTAM Macro Language Reference* as a guide. If the PSERVIC operand is not coded, a value of X'0' is assumed.

PSNDPAC=value
specifies the primary send pacing count (byte 11 in the session parameter field). Any hexadecimal number in the range 0 through 3F can be coded. See "Pacing" in Chapter 2 for more information on primary send pacing. If the PSNDPAC operand is not coded, a value of X'0' is assumed.

SRCVPAC=value
specifies the secondary receive pacing count (byte 8 in the session parameter field). Any hexadecimal number in the range 0 through 3F can be coded. If the SRCVPAC operand is not coded, a value of X'0' is assumed.

*For OS/VS2 SVS only. For OS/VS2 MVS and OS/VS1, the default indicates there is no limit on the size of the RU that may be sent.

SSNDPAC=value

specifies the secondary send pacing count (byte 7 in the session parameter field). Any hexadecimal number in the range 0 through 3F can be coded. See "Pacing" in Chapter 2 for more information on secondary send pacing. If the SRCVPAC operand is not coded, a value of X'0' is assumed.

ENCR=value

can be specified only if the ACF/VTAM Encrypt/Decrypt Feature is installed. ENCR=value specifies what type of cryptography is expected by the logical unit. Any hexadecimal number in the range 0 through F (or an equivalent decimal value) can be specified. A 4-bit binary string (for example, ENCR=B'0001') can also be specified. The default value is 0.

The ENCR value is converted to a 4-bit string, which is treated as two 2-bit fields. The meanings of the bit settings are as follows:

xx . .	Private cryptography field
00 . .	No private cryptography
01 . .	Private cryptography used
. . xx	ACF/VTAM cryptography field
. . 00	No session-level cryptography
. . 01	Selective cryptography; the primary logical unit can encipher messages and the secondary logical unit must support cryptography.
. . 10	Reserved
. . 11	Mandatory cryptography; all messages on this session will be enciphered and deciphered.

This value is stored in the first four bits of byte 26 in the session parameter field.

The MODEEND Macro Instruction

The MODEEND macro instruction indicates the end of the logon mode table. It follows a MODEENT macro instruction and is coded as follows:

Name	Operation	Operand
[name]	MODEEND	

name

is optional; it has no function in the specification of the logon mode table. This macro instruction has no operands.

Logon And Logoff Commands

When a logical unit is ready to establish connection with a particular application program, it sends a logon to ACF/VTAM specifying the application program's name and (optionally) a logon mode name and some additional user data.

One form in which the logon can be sent is as an Initiate Self command whose format is fixed. Some SNA terminals (such as the 3600, 3650, 3660, and 3790 physical units) can automatically change their logical unit's logons into Initiate Self commands. ACF/VTAM, however, allows any logical unit to send a logon request in a variable form. Similarly, a logoff can be sent as a Terminate Self command, which causes the application program to break the connection between itself and the logical unit. ACF/VTAM also allows any logical unit to send a logoff request in a variable form. The "variable" logon or logoff is called an unformatted, or character-coded, command.

For logical units that are sources of character-coded commands, you can associate the logical unit's definition statement with a USS definition table that converts the command into either a logon or logoff command of the following format:

LOGON APPLID(name) LOGMODE(name) DATA(userdata)

APPLID(name)
specifies the name of the application program with which a session is to be established.

LOGMODE(name)
is used to select a set of session parameters for the session to be established.

DATA(userdata)
specifies user data to be made available to the application program's logon exit routine.

LOGOFF [APPLID(name)] [TYPE(COND | UNCOND)] [HOLD(YES | NO)]

APPLID(name)
specifies the name of the application program with which a session is to be terminated. If omitted, the application program with which a current session exists is assumed.

TYPE(COND | UNCOND)
specifies whether conditional or unconditional termination of the active session between the logical unit and the application program is requested. If conditional termination is requested, the logical unit is disconnected at the discretion of the application program.

HOLD(YES | NO)
specifies the action the logical unit expects ACF/VTAM to take in regard to physically disconnecting the physical unit after the logical unit has itself been disconnected. A HOLD value of YES corresponds to a NOT LAST indicator on a Terminate Self command. The effect of HOLD depends on the setting of the DISCNT operand specified on the PU statement. The relationship between HOLD and DISCNT is described in the description of the DISCNT operand of the PU (local) and PU (switched) statements in Chapter 2, and in the PU macro instruction description in *NCP Generation Manual.*

The USS Definition Table

To convert a character-coded command into a logon or logoff command, ACF/VTAM uses a USS definition table, as illustrated in Figure 4-4.

ACF/VTAM cannot be started unless a USS definition table named ISTINADT (for OS/VS1 and OS/VS2 SVS) or ISTINCDT (for OS/VS2 MVS) to meet this requirement a suitably named IBM-written USS definition table is supplied as part of ACF/VTAM. Included as part of the IBM-supplied USS definition table is an IBM-supplied translation table named STDTRANS. Except for their names, these tables are exactly the same for all OS/VS systems. Figure 4-5 shows the macro instructions that make up the IBM-supplied USS definition table.

The IBM-supplied USS definition table can be modified or replaced by the user, provided that the modified table or replacement table has the same name as the IBM-supplied table and that the IBM-supplied table is deleted.

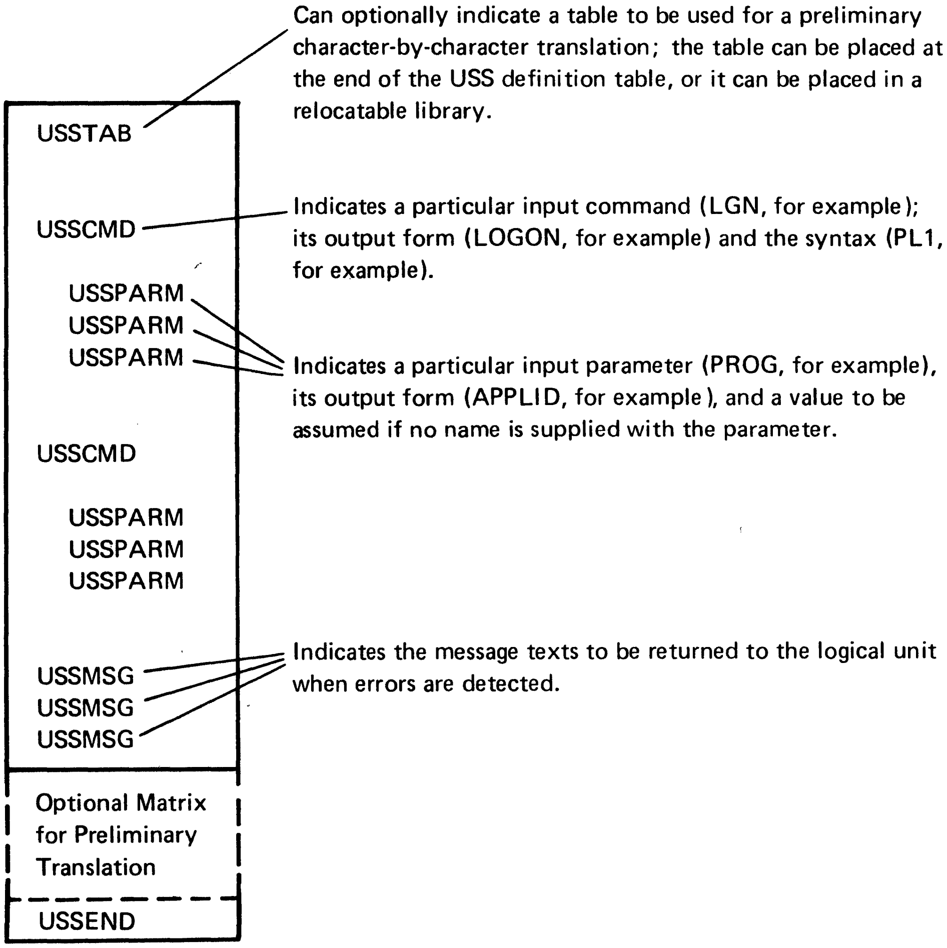

Can optionally indicate a table to be used for a preliminary character-by-character translation; the table can be placed at the end of the USS definition table, or it can be placed in a relocatable library.
USSTAB
USSCMD
Indicates a particular input command (LGN, for example); its output form (LOGON, for example) and the syntax (PL1, for example).
USSPARM
USSPARM
USSPARM
Indicates a particular input parameter (PROG, for example), its output form (APPLID, for example), and a value to be assumed if no name is supplied with the parameter.
USSCMD
USSPARM
USSPARM
USSPARM
USSMSG
USSMSG
USSMSG
Indicates the message texts to be returned to the logical unit when errors are detected.
Optional Matrix for Preliminary Translation
USSEND

Figure 4-4. USS Definition Table Macro Instructions

In addition to the required USS definition table just discussed, you can create and name optional USS definition tables with USSTAB, USSCMD, USSPARM, USSMSG, and USSEND macro instructions and then assemble and link edit the tables to SYS1.VTAM LIB. A USS definition table can be associated with a logical unit by specifying the table's name in the USSTAB operand of the logical unit's definition statement. If you do not designate a USS definition table for a logical unit, ACF/VTAM uses the IBM-supplied USS definition table.

ACF/VTAM returns messages to the logical unit if the logon or logoff commands are invalid. For example, if the logical unit's input sequence is not translatable by the USS definition table, ACF/VTAM informs the logical unit of that fact.

Creating a USS Definition Table

To create a USS definition table, assemble a USSTAB macro instruction, followed by a USSCMD macro instruction and its associated USSPARM macro instructions for each command to be defined. Code a USSMSG macro instruction for each message text to be defined. If a character translation table is to be specified, code the table using assembler DC statements. If the table is to be part of another module, code an EXTRN statement for the table name. Follow this with a USSEND macro instruction to indicate the end of the table definition.

```
ISTINCDT    USSTAB     TABLE=STDTRANS
LOGON       USSCMD     CMD=LOGON,FORMAT=PL1
            USSPARM    PARM=APPLID
            USSPARM    PARM=LOGMODE
            USSPARM    PARM=DATA
LOGOFF      USSCMD     CMD=LOGOFF,FORMAT=PL1
            USSPARM    PARM=APPLID
            USSPARM    PARM=TYPE,DEFAULT=UNCOND
            USSPARM    PARM=HOLD,DEFAULT=NO
MESSAGES    USSMSG     MSG=1,TEXT=’INVALID COMMAND SYNTAX’
            USSMSG     MSG=2,TEXT=’% COMMAND UNRECOGNIZED’
            USSMSG     MSG=3,TEXT=’% PARAMETER UNRECOGNIZED’
            USSMSG     MSG=4,TEXT=’% PARAMETER INVALID’
            USSMSG     MSG=5,TEXT=’UNSUPPORTED FUNCTION’
            USSMSG     MSG=6,TEXT=’SEQUENCE ERROR’
            USSMSG     MSG=7,TEXT=’SESSION NOT BOUND’
            USSMSG     MSG=8,TEXT=’INSUFFICIENT STORAGE’
            USSMSG     MSG=9,TEXT=’MAGNETIC CARD DATA ERROR’
STDTRANS    DC         X’000102030440060708090A0B0C0D0E0F’
            DC         X’101112131415161718191A1B1C1D1E1F’
            DC         X’202122232425262728292A2B2C2D2E2F’
            DC         X’303132333435363738393A3B3C3D3E3F’
            DC         X’404142434445464748494A4B4C4D4E4F’
            DC         X’505152535455565758595A5B5C5D5E5F’
            DC         X’606162636465666768696A6B6C6D6E6F’
            DC         X’707172737475767778797A7B7C7D7E7F’
            DC         X‘80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F’
            DC         X’90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F’
            DC         X’A0A1E2E3E4E5E6E7E8E9AAABACADAEAF’
            DC         X’B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF’
            DC         X’C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF’
            DC         X’D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF’
            DC         X’E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF’
            DC         X’F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF’
END         USSEND
```

Figure 4-5. The IBM-Supplied USS Definition Table

Link-edit the resultant object deck into SYS1.VTAMLIB as serially reusable and specify the module name in the USSTAB operand of the LU definition statement for each logical unit to be supported by this definition table.

The following macro instructions define a USS definition table. A description of the IBM-supplied USS definition table, an explanation of the command conversion process (including examples), and a description of character-coded command syntax follow the descriptions of these macro instructions.

The USSTAB Macro Instruction

The USSTAB macro instruction indicates the beginning of a USS definition table. It can optionally specify the table to be used for character translation. See "Conversion of Character-Coded Commands" in this chapter for a description of character translation.

Name	Operation	Operand
[name]	USSTAB	[TABLE=name]

name
if specified, is used as a CSECT name for the USS definition table to be generated; otherwise, no CSECT name is generated.

TABLE=name
specifies a translation table to be used by ACF/VTAM to translate each character-coded command received from a logical unit. If the table is not part of the module containing USSTAB, an EXTRN statement must be coded for the specified name.

If no translation table is specified, ACF/VTAM uses the translation table associated with the IBM-supplied table (or its user-written replacement).

The USSCMD Macro Instruction

USSCMD identifies a set of definition statements to be used to convert a user-defined command that can be received from a logical unit into a valid LOGON or LOGOFF command. It can also specify the syntax of the resulting LOGON or LOGOFF command.

Name	Operation	Operand
[name]	USSCMD	CMD=name [,REP=name] [,FORMAT=BAL | <u>PL1</u>]

name
is any valid symbol and is optional.

CMD=name
specifies the user-defined command name to which this USSCMD macro instruction applies. Ensure that no two CMD operands specify the same command name for a single USS definition table.

REP=name

specifies the valid command (LOGON or LOGOFF) that is to replace the user-defined command specified by the CMD operand. If REP is not coded, the value in CMD is used; CMD must specify LOGON or LOGOFF in this case.

FORMAT=BAL | PL1

indicates the syntax of the user-entered command. Code FORMAT=BAL to use the BAL assembler language syntax or FORMAT=PL1 to use the PL/I programming language syntax. If FORMAT is not specified or is specified incorrectly, FORMAT=PL1 is used. Sėe "Character-Coded Command Syntax" for a description of the syntax, the input character set, and PL/I and BAL value restrictions.

The USSPARM Macro Instruction

The USSPARM macro instruction identifies a user-defined keyword or positional parameter that can be coded in the user-defined command identified by the previous USSCMD definition. This macro instruction can also specify that this user-defined keyword is to be converted into a valid parameter for the LOGON command. A default value for this parameter can also be defined.

Name	Operation	Operand
[name]	USSPARM	{ PARM=Pn PARM=name } [,REP=name] [,DEFAULT=value]

name

is any valid symbol and is optional.

PARM=Pn

identifies a positional parameter where n is a decimal number between 1 and the maximum number of positional parameters for the command. Pn indicates the positional parameter in the user-entered command to which this USSPARM macro instruction applies. Do not code the same PARM parameter more than once for a single command.

PARM=name

specifies the keyword that identifies the parameter in the user-entered command to which this USSPARM macro instruction applies. Do not code the same PARM parameter more than once for a single command.

REP=name

specifies a keyword that is to appear in the LOGON or LOGOFF command that is to be generated from the user-defined command. The value of the keyword is assigned from the parameter specified by PARM. If PARM specifies a keyword parameter, its value is assigned to the keyword specified by REP. If PARM specifies a positional parameter, its value is treated as if it were a keyword value and it is assigned to the keyword specified by REP.

If REP is not coded, it takes the value of PARM. (That is, the user-entered parameter is used just as it is entered.)

Such positional parameters as P1,P2, . . . can also be used as keywords. For multiple specifications of the same parameter, the last value specified is used (as shown in Example 3 at the end of this section).

DEFAULT=value

specifies a default value to be used for the parameter identified by the PARM operand if the parameter is omitted. Single quotation marks in the default value must be specified as in the assembler DC statement for character (C-type) constants.

If a keyword parameter or a positional parameter is not entered for a user-defined command, the DEFAULT value from USSPARM is used; if DEFAULT is not specified, a null value is used.

For BAL assembler language syntax, if a keyword is specified without any value (KWD=” or KWD=, for example), a null value is used. If ” is specified for a positional parameter, a null value is used.

For PL/I programming language syntax, if a keyword is specified without any value (KWD ( ) or KWD, for example), a null value is used. If (”) is specified for a positional parameter, a null value is used.

The USSMSG Macro Instruction

The USSMSG macro instruction defines an alternate message text for a USS message. To determine under what circumstances a message is issued, refer to Appendix B.

Name	Operation	Operand
[name]	USSMSG	{MSG=n MSG=(n1,n2, . . .)} ,TEXT='message text'

name

is any valid symbol and is optional.

MSG=n or MSG=(n1,n2, . . .)

indicates the USS messages to be redefined. Enter decimal numbers in the range 0 through 10. The numbers 1 through 9 correspond to the USS messages below (these messages are explained in Appendix B):

MSG=1 corresponds to message INVALID COMMAND SYNTAX
MSG=2 corresponds to message command COMMAND UNRECOGNIZED
MSG=3 corresponds to message parameter PARAMETER UNRECOGNIZED
MSG=4 corresponds to message parameter PARAMETER INVALID
MSG=5 corresponds to message UNSUPPORTED FUNCTION
MSG=6 corresponds to message SEQUENCE ERROR
MSG=7 corresponds to message SESSION NOT BOUND
MSG=8 corresponds to message INSUFFICIENT STORAGE
MSG=9 corresponds to message MAGNETIC CARD DATA ERROR

The number 0 corresponds to a message indicating that the USS command has completed successfully. The number 10 corresponds to a message that is automatically sent to a logical unit (or a terminal for which PU=YES is specified) whenever the logical unit is available for use – such as when it is activated or powered-on, or whenever a user issues a logoff at the terminal. There are no such messages among the IBM-supplied messages, but you can create them using USSMSG.

TEXT='message text'
specifies the text to replace the USS messages identified by the MSG operand.

Single quotation marks in the message text must be specified as in the assembler DC statement for character (C-type) constants.

Variable data in the replacement text of a message is indicated by a percent sign (%). A percent sign in messages 3 or 4 is replaced by the keyword used to enter an erroneous keyword parameter or by Pn, where n is a decimal number corresponding to the sequential position of an erroneous positional parameter value. In any other messages, the percent sign is replaced by the verb entered by the user (if possible) or is deleted from the message text if the entered verb cannot be ascertained.

If more than one percent sign is contained in a message prototype, all such percent signs are replaced by the same character string.

The maximum length of a message after replacement of any percent signs is 255 characters. If a message exceeds 255 characters, only the first 255 characters are sent to a terminal.

The following characters can be used for USS messages:

26 uppercase letters: A-Z
3 national characters: $ # @
10 numeric digits: 0-9
12 special characters: blank ' = ( ) , + - * . / &
1 control character: New line (NL = X'15')

National characters (and any graphic or control characters not listed above) are sent to a terminal user only if present in user-specified message replacements.

The USSEND Macro Instruction

The USSEND macro instruction indicates the end of a USS definition table.

Name	Operation	Operand
[name]	USSEND	

name
is any valid symbol and is optional.

This macro instruction has no operands.

Order of USS Definition Table Use

When ACF/VTAM receives a character-coded command from a logical unit, it searches the IBM-supplied and user-supplied definition tables (if they exit) in a specific order. The order of table use is as follows:

For a translation table: ACF/VTAM uses the translation table specified by the TABLE operand of the USSTAB macro instruction associated with the logical unit (if such a table exists); otherwise ACF/VTAM uses the translation table associated with the IBM-supplied USS definition table (if such a translation table exists); otherwise ACF/VTAM does no translation.

For a verb: If the verb is in a user-defined USS definition table, ACF/VTAM uses the translation for the verb provided by that table, otherwise it searches the IBM-supplied USS definition table for the verb and if the verb is found, ACF/VTAM uses the translation for the verb that is provided there, otherwise ACF/VTAM uses the verb as entered.

For parameters on a verb: If the verb is in the IBM-supplied USS definition table or a user-defined USS definition table, ACF/VTAM searches that table for the parameter. If the parameter is found, ACF/VTAM uses the translation provided by the table. If the verb is not found or the parameter is not found, ACF/VTAM uses the parameter as entered.

For messages: If the message is in a user-supplied table, ACF/VTAM uses the text of the message provided by the table; otherwise ACF/VTAM searches the IBM-supplied table and if the message is found, ACF/VTAM uses the message text provided there, otherwise ACF/VTAM issues the message "MESSAGE NOT DEFINED".

Conversion Of Character-Coded Commands

As illustrated in Figure 4-6, when ACF/VTAM receives a character-coded command from a logical unit, (1) media control characters (cursor address, set buffer address characters, magnetic card reader longitudinal redundancy characters for 3270 device control, and new line characters for SCS device control) are deleted from the command. Then (2) the character string is translated according to the translation table specified by the TABLE operand of the USSTAB macro instruction for that logical unit. If no USS table is specified for the logical unit, or if no translation table is specified for the USS definition table, then ACF/VTAM uses the translation table in the IBM-supplied USS definition table. If the IBM-supplied USS definition table has been replaced by a user-written table and TABLE has been omitted, translation is not performed. If the IBM-supplied translation table (STDTRANS) is used, lowercase letters from a to z are replaced by the corresponding uppercase letters, horizontal tab characters are replaced with blanks, and all other characters remain unchanged.

Characters located between pairs of single quotation marks (for example, 'abc') are not translated. Operator ID card characters are not translated. An unpaired single quotation mark (') is identified by ACF/VTAM before character string translation and therefore cannot be translated or replaced. However, other characters can be translated into single quotation marks (if not between quotation marks). Care should be taken to prevent unpaired quotation marks resulting in the converted commands.

ACF/VTAM uses the translated character string to construct a reformatted USS command. It does this by (3) first using the verb (the first field) of the translated string to search the USS definition table for the associated entry built by the USSCMD macro instruction. If a replacement verb was specified on USSCMD, that verb is placed in the converted standard command being constructed.

Parameters in the character-coded command are replaced using information supplied by the USSPARM macro instructions associated with the USSCMD macro instruction. If a parameter was not supplied and a default was given on USSPARM, the default value is placed in the converted command; otherwise, the value on the character-coded command is used. If a parameter is specified more than once, the last specification of the parameter is used.

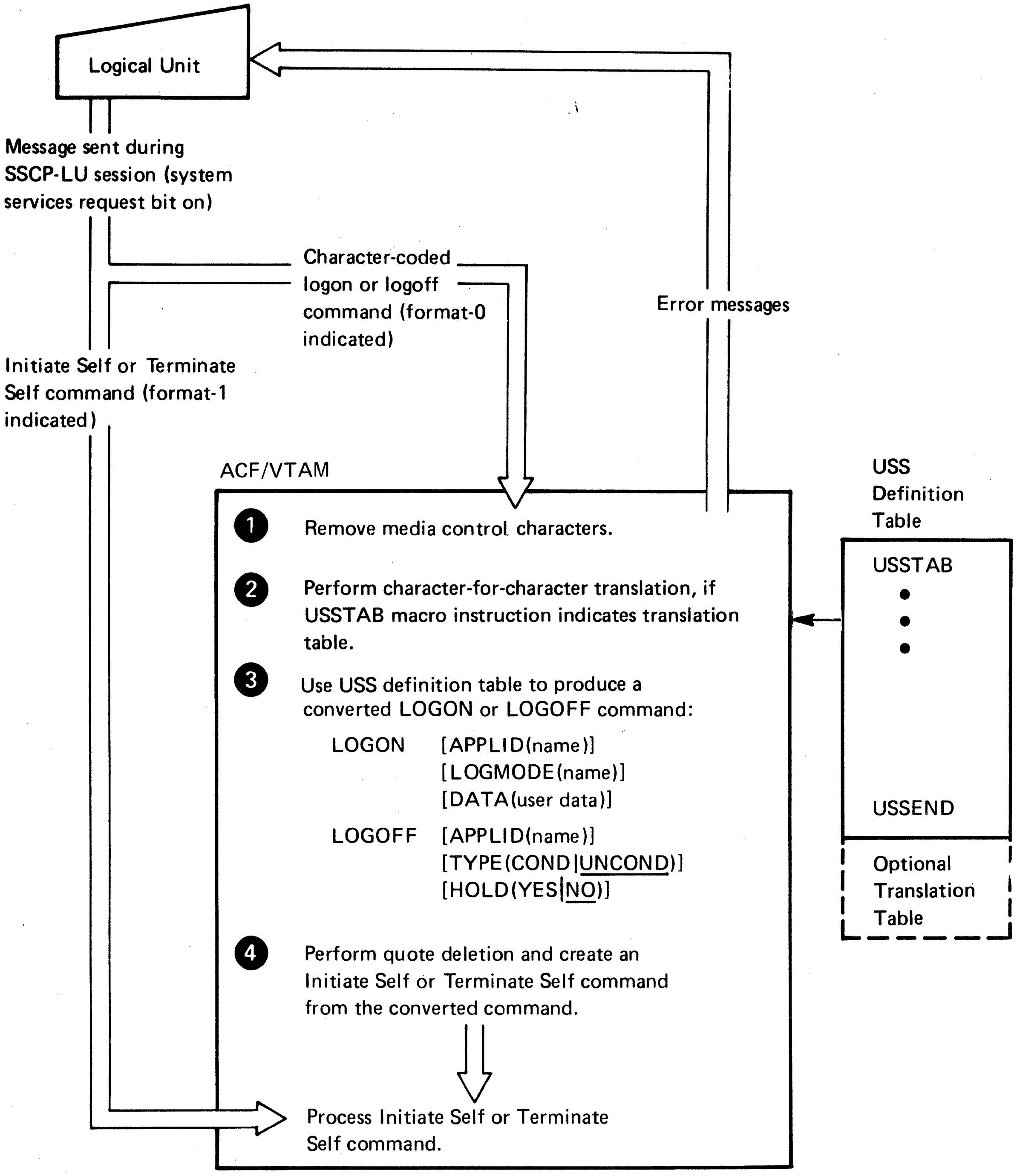

Logical Unit
Message sent during SSCP-LU session (system services request bit on)
Character-coded logon or logoff command (format-0 indicated)
Error messages
Initiate Self or Terminate Self command (format-1 indicated)
ACF/VTAM
1 Remove media control characters.
2 Perform character-for-character translation, if USSTAB macro instruction indicates translation table.
3 Use USS definition table to produce a converted LOGON or LOGOFF command:
LOGON [APPLID(name)] [LOGMODE(name)] [DATA(user data)]
LOGOFF [APPLID(name)] [TYPE(COND|UNCOND)] [HOLD(YES|NO)]
4 Perform quote deletion and create an Initiate Self or Terminate Self command from the converted command.
Process Initiate Self or Terminate Self command.
USS Definition Table
USSTAB
•
•
•
USSEND
Optional Translation Table

Figure 4-6. Conversion of Character-Coded Commands

(4) ACF/VTAM deletes quotation marks in converted logon and logoff commands, but only if both the first and last characters in a value are quotation marks and all intervening single quotation marks are paired (two adjacent single quotation marks). The first and last single quotation marks are deleted and each sequence of two adjacent single quotation marks is replaced by a single quotation mark. For example:

‘Don”t tread on me.’

results in

Don’t tread on me.

while:

X‘A4’(X‘3F’)

remains

X‘A4’(X‘3F’)

The deletion of quotation marks occurs for all parameters of the converted command, not just the DATA parameter.

Examples of Command Conversion

Example 1

The example of USS command conversion shown in Figure 4-7 uses this sample USS definition table:

```
USSTAB
*
* THE START COMMAND
*
USSCMD CMD=START,REP=LOGON,FORMAT=PLI
USSPARM PARM=P1,REP=APPLID,DEFAULT=SYSPROG
USSPARM PARM=MODE,REP=LOGMODE,DEFAULT=BATCH
USSPARM PARM=INPUT,REP=DATA,DEFAULT=XYZ
USSEND
```

In this example, MYPROG is a positional parameter that becomes a value when the command is converted. MODE is a keyword parameter that becomes LOGMODE after conversion.

The next two examples use the following USS definition table:

```
USSTAB
*
* THE RUN COMMAND
*
USSCMD CMD=RUN,REP=LOGON,FORMAT=BAL
USSPARM PARM=P1,REP=APPLID,DEFAULT=SYSPROG
USSPARM PARM=P2,REP=LOGMODE,DEFAULT=BATCH
USSPARM PARM3=P3,REP=DATA
USSEND
```

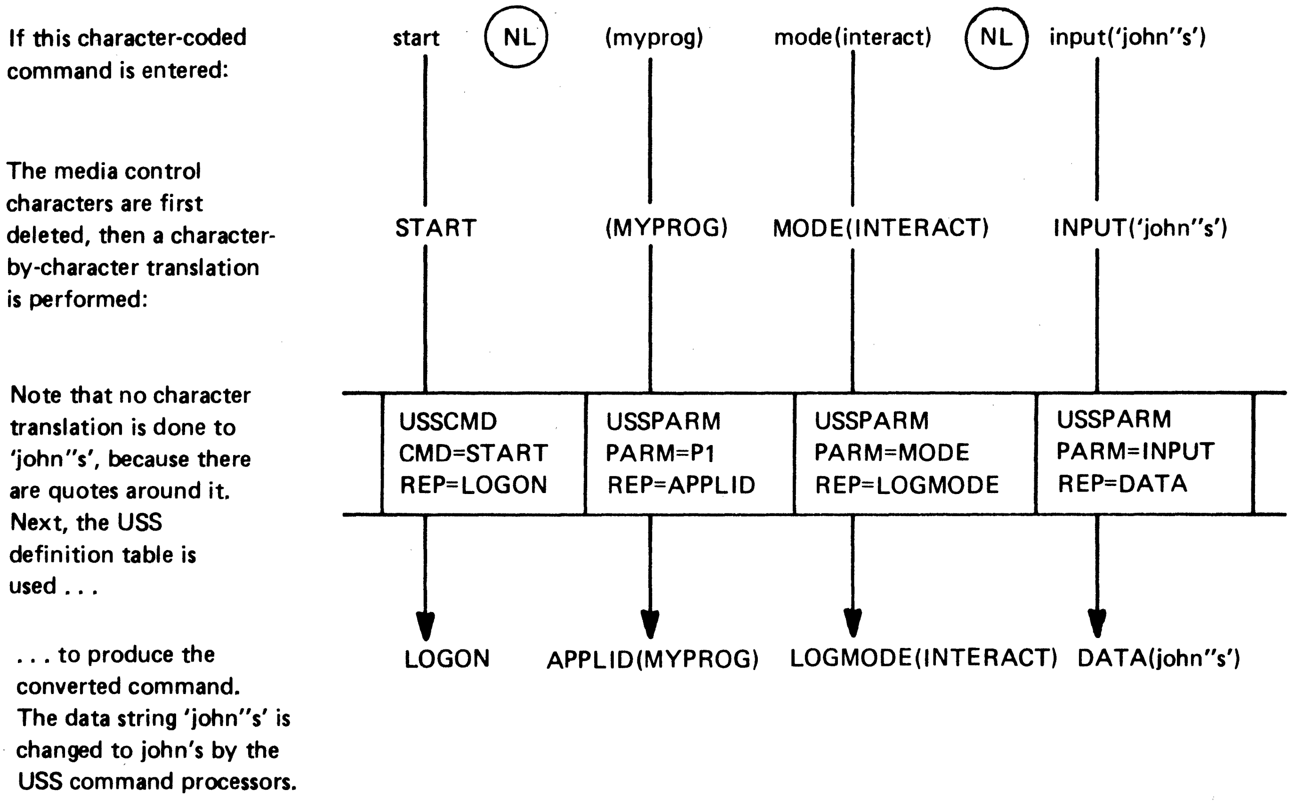

If this character-coded command is entered:
start
NL
(myprog)
mode(interact)
NL
input('john''s')
The media control characters are first deleted, then a character-by-character translation is performed:
START
(MYPROG)
MODE(INTERACT)
INPUT('john''s')
Note that no character translation is done to 'john''s', because there are quotes around it. Next, the USS definition table is used . . .
USSCMD
CMD=START
REP=LOGON
USSPARM
PARM=P1
REP=APPLID
USSPARM
PARM=MODE
REP=LOGMODE
USSPARM
PARM=INPUT
REP=DATA
. . . to produce the converted command. The data string 'john''s' is changed to john's by the USS command processors.
LOGON
APPLID(MYPROG)
LOGMODE(INTERACT)
DATA(john''s')

Figure 4-7. Example of Command Conversion

Example 2

All parameters are omitted and are therefore supplied default during conversion. The command

```
RUN
```

is converted to a command of the form

```
LOGON APPLID (SYSPROG) LOGMODE (BATCH) DATA ( )
```

Example 3

Parameter P2 is defined as both a positional and a keyword parameter. The command

```
RUN A,B,C,P2=Z
```

is converted to a command of the form

```
LOGON APPLID(A) LOGMODE(Z) DATA(C)
```

Because the last value specified for P2 was P2=Z, the value Z was used for the LOGMODE value.

The next two examples use the following USS definition table:

```
USSTAB
*
* THE LOFF COMMAND
*
USSCMD CMD=LOFF,REP=LOGOFF,FORMAT=PL1
USSPARM PARM=P1,REP=APPLID
USSPARM PARM=T,REP=TYPE,DEFAULT=COND
USSEND
```

Example 4

This example demonstrates the use of positional and keyword parameters when FORMAT=PL1. The command

```
Loff (PROG) T(COND);
```

is converted to a command of the form

```
LOGOFF APPLID(PROG) TYPE(COND)
```

Example 5

A null value is taken instead of a default value. The command

```
LOFF (PGM) T
```

is converted to a command of the form

```
LOGOFF APPLID(PGM) TYPE (  )
```

Because T was coded, the default value specified in the definition table is not used. If T had not been coded, TYPE(COND) would have resulted. TYPE ( ) causes unconditional termination.

The next example uses the following USS definition table:

```
USSTAB
*
* THE LON COMMAND
*
USSCMD CMD=LON,REP=LOGON,FORMAT=BAL
USSPARM PARM=P1,REP=APPLID,DEFAULT=TESTPROG
USSPARM PARM=MODE,REP=LOGMODE,DEFAULT=‘PROMPT’
USSPARM PARM=IN,REP=DATA
USSEND
```

Example 6

This example demonstrates the positional and keyword parameters when FORMAT=BAL. The command

```
LON PROGRAM,IN=‘7,3,JOhn’
```

results in a converted command of the form

```
LOGON APPLID(PROGRAM) LOGMODE(PROMPT) DATA(‘7,3,JOhn’)
```

Note that no character translation was performed on JOhn because there were single quotation marks around it. Note also that the single quotation marks around PROMPT in the default declaration have been deleted.

The next example uses the following USS definition table:

```
USSTAB
*
* THE ON COMMAND
*
USSCMD CMD=ON,REP=LOGON,FORMAT=BAL
USSPARM PARM=P1,REP=APPLID
USSPARM PARM=LMOD,REP=LOGMODE
USSEND
```

Example 7

This example uses a parameter (INPUT) that is not defined on the USS definition table above. The command

```
ON PGM76,LMOD=MODE,DATA=(A,B,C)
```

results in a converted command of the form

```
LOGON APPLID(PGM76) LOGMODE(MODE) DATA(A,B,C)
```

Because DATA is not defined by a USSPARM macro instruction in the table above, the keyword is not changed during conversion.

Character-Coded Command Syntax

The following characters can be used in a character-coded command entered by a logical unit:

All graphics characters (greater than or equal to X'40')
BS (Backspace: X'16')
HT (Horizontal Tab: X'05')
SSR (Start Secure Reader String: X'0450')
IRS (Interchange Record Separator: X'1E')
NL (new line: X'15'); deleted from the character-string before translation if SSCPFM=FSS is specified
3270 SBA (Set Buffer Address: X'11'); deleted from the character string before translation if SSCPFM=USS3270 is specified.
3270 AID (Attention Identifier); deleted from the character string before translation if SSCPFM=USS3270 is specified.

In character-coded commands, parameter values cannot contain blanks, horizontal tabs, or unpaired parentheses except between paired single quotation marks. A parameter value cannot contain an odd number of single quotation marks.

After translation, verbs can contain from 1 to 8 alphanumeric characters, the first of which must be alphabetic (A-Z, $, #, or @). Keywords can contain from 1 to 7 alphanumeric characters, the first of which must be alphabetic.

Values can contain any of the following characters:

All graphics characters (greater than or equal to X'40')
BS (backspace: X'16')
HT (horizontal tab: X'05')
Data entered from a magnetic card reader

Magnetic card data from a BSC or type 1 PU 3270 device is supported only if the card data is used as the last data in a value within quotation marks for the last parameter of a command. The CLEAR key should be pressed before entering a character-coded command containing magnetic card reader data.

If FORMAT=PL1 is specified or assumed by default in the USSCMD macro instruction, the following syntax for a character-coded command must be specified:

```
verb [(p1,p2, . . .)] [keyword[(value)] ] [keyword[(value)] ] [ . . . ] [ ; ]
```

verb
identifies the command. It is followed by one or more blanks or by a left parenthesis (that is, positional parameters).

(p1,p2, . . .)
is used to enter one or more positional parameters. Note that if used, the parentheses must be coded.

keyword[(value)]
is used to enter each keyword parameter. Each keyword must be followed by one or more blanks or by a value enclosed in parentheses.

If FORMAT=PL1 is specified or assumed by default, coded values cannot contain semicolons except between paired single quotation marks. A positional parameter value can not contain commas except between paired single quotation marks or parentheses.

If FORMAT=BAL is specified, the unformatted command must have the following syntax:

```
verb [p1,p2, . . .] [keyword=[value]] [,keyword=[value]] [ . . .]
```

verb
identifies the command. It is followed by one or more blanks.

p1,p2, . . .
is used to enter one or more positional parameters. Each parameter (unless it is the last in the command) is followed by a comma.

keyword=[value]
is used to enter each keyword parameter. Each parameter (unless it is the last in a command) is followed by a comma.

Blanks or horizontal tab characters are not permitted in a BAL command except between the verb and the first parameter or between paired single quotation marks. Values cannot contain commas except between paired parentheses or single quotation marks. A positional parameter can not contain equal signs except between paired parentheses or single quotation marks.

Automatic Logons

You can tailor ACF/VTAM to perform automatic logons and to manage terminal-initiated logons for application programs. An automatic logon causes ACF/VTAM to generate the logon for a terminal to a specific application program. (You also need automatic logon to permit terminal-initiated logons for devices that use the basic mode of data transfer.) A terminal-initiated logon causes ACF/VTAM to use the logon message (entered by a logical unit or terminal user) to establish connection with a specific application program.

To provide terminal-initiated logons, consider these steps:

Defining automatic logon capability (for terminals that use either the record or basic mode of data transfer)

Providing OS/VS logon capability (when no interpret table represents a terminal that uses basic mode of data transfer)

Defining interpret tables to contain valid logon messages (for terminals that use either record or basic mode)

Providing a logon monitor (network solicitor) facility (an option for terminals that use basic mode)

To enable ACF/VTAM to process automatic logons, you must specify an application program that is to be notified whenever a terminal designated for automatic logon is available for connection. This application program can be the program to which the terminal ultimately wishes to log on, or it can be an intermediary application program (such as the network solicitor) to which the terminal is temporarily assigned. If the application program and terminal are in different domains, you must provide an active path between them.

When an automatic logon is desired for a terminal, the name of the application program is specified in the LOGAPPL operand of the macro instruction or definition statement that represents this terminal.

If you are using the IBM-supplied network solicitor, the name in the LOGAPPL operand must be NETSOL (except for the LU statement; the IBM-supplied network solicitor does not monitor logical units). Otherwise, the name in the LOGAPPL operand must be the name assigned to the application program by APPL definition statement.

OS/VS Logon For Non-SNA Terminals

OS/VS logon is available for non-SNA terminals if an interpret table is not used. For OS/VS logon for SNA terminals, see "Logon and Logoff Commands" earlier in this chapter. Refer to *ACF/VTAM Installation Guide* for a summary of preparations required to use the network solicitor either with an interpret table or with OS/VS logon. The requirements for installing terminal-initiated logon for local non-SNA, BSC, and start-stop devices differ from the requirements for logical units in that the network solicitor facility must be present and automatic logon to the network solicitor must be specified for local non-SNA, BSC, and start-stop devices.

OS/VS logon provides a default logon capability to the user of a local non-SNA, BSC, or start-stop terminal if the terminal does not have access to an interpret table. The security of the teleprocessing system might be lessened if OS/VS logon is used in that a terminal can now request connection to any application program and is not limited to the application programs listed in the interpret table. However, you can prescribe that data be added to the logon message for use by a LOGON exit routine in the application program. The application program can then decide whether or not to accept the connection request. An authorization exit routine also can be used to decide whether or not to accept the connection request.

To enter the OS/VS logon, the terminal is assigned to the network solicitor, but an interpret table is not specified for the terminal. NETSOL is coded in the LOGAPPL operand, and the LOGTAB operand is not coded in the macro instructions or definition statements that are to represent this terminal. Therefore, if the network solicitor encounters a logon message from a terminal that does not have an interpret table specified, it assumes that the message is an OS/VS logon message.

When an OS/VS logon message is entered from a 3275 or 3277 terminal, the terminal's control unit inserts leading device control characters into the message. Because the IBM-supplied network solicitor deletes any leading (and trailing) device control characters from these logon messages, application programs should not expect these characters to be part of the logon message format. (For OS/VS2 MVS only, this function can be controlled with the STRPCNL operand of the NETSOL macro instruction.)

The format of the OS/VS logon message is:

```
LOGON[,other-data] [,APPLID(applname)] [,other-data]
```

The operands of the logon message can be separated with blanks instead of commas, if you wish.

LOGON

indicates that this is an OS/VS logon message. LOGON is required and must be the first word of the logon message.

other-data

indicates optional, user-defined information that can be made available to an application program.

APPLID(applname)

indicates the name of the application program to which the terminal is to be connected; this operand is optional.

If APPLID(applname) is specified, applname must be a 1 to 8 alphameric character name of an application program. The applname must also be the same as the name assigned to the application program by an APPL definition statement.

other-data

indicates optional, user-defined information that can be made available to an application program, such as accounting or security information that identifies the terminal user.

Interpret Tables

ACF/VTAM uses the interpret table to determine, for each logon message received from a terminal, which application program is to be notified of the connection request. (For local non-SNA, BSC, and start-stop devices, the interpret table notifies the network solicitor of the name of the application program.) The terminal is connected to the application program only when the application program accepts the request. When a terminal logs off (causing the application program to issue a CLSDST macro instruction), ACF/VTAM regains ownership of the terminal unless the terminal has an automatic logon specification. In this case, ACF/VTAM usually attempts to queue the terminal to the application program named in the specification.

Defining Logon Messages

The system programmer defines a valid logon message for each application program listed in a particular interpret table. The logon sequence must be entered by the logical unit or by the terminal user to initiate the connection request, and it may be followed by optional information. The optional information might be used, for example, by a user-written routine that restricts access to an application program, or by the LOGON exit routine in the application program. A user identifier or a password, for example, can be used in the routine to determine whether the logon will be accepted by the application program.

(Refer to the discussion of the INTRPRET and INQUIRE macro instructions in the *ACF/VTAM Macro Language Reference.)*

RESTRICTION: The length of the entire logon message (required information plus optional information) is limited to 255 characters.

If one or more blanks precede the logon message sent from the terminal, the blanks are ignored.

A single logon message entered from different terminals can initiate logon to the same or to different application programs. This is done by associating various terminals with the same or different interpret tables.

Establishing conventions for logon messages, including the choice of delimiters and the use of optional information in the logon message, is the user's responsibility.

Using An Interpret Table

Unless OS/VS logon is being used, the network solicitor requires the user to code one or more interpret tables before starting ACF/VTAM. The table is identified in NCP macro instructions and LOCAL, PU (local), LU (local), PU (switched), or LU (switched) statements by the name assigned to the table when it is link-edited.

Three ACF/VTAM macro instructions are used to construct an interpret table: INTAB, LOGCHAR, and ENDINTAB. Briefly, INTAB defines the beginning of an interpret table; LOGCHAR describes the logon requirements for each application program in the table, and ENDINTAB defines the end of the table (Figure 4-8).

For each application program, LOGCHAR specifies the logon message that initiates the attempt to connect the requesting terminal to an application program and the name of the application program to receive this logon. Instead of the application program name, a user-written routine can be specified. The routine can be coded to further restrict the application program to which the logon is to be passed.

Interpret tables are also used by ACF/VTAM during the processing of logons from local non-SNA, BSC, and start-stop terminals. Because these terminals are incapable of initiating a logon, an application program, the network solicitor, establishes connection with these terminals (if so specified by the user) and obtains data from each terminal. The network solicitor treats the data as a logon message, which it uses as input for the interpret table for that terminal. ACF/VTAM searches the table for a match and returns the 8-byte output sequence to the network solicitor. (The table can contain the output sequence, or it can contain the name of a user-written routine that provides the output sequence during program execution.) The network solicitor treats the output sequence as the name of an application program (as defined by the label of the application program's APPL definition statement) and logs the terminal on to that application program.

An IBM-written network solicitor is provided as part of ACF/VTAM. You can use, ignore, or replace the IBM-supplied network solicitor, as explained later in this chapter.

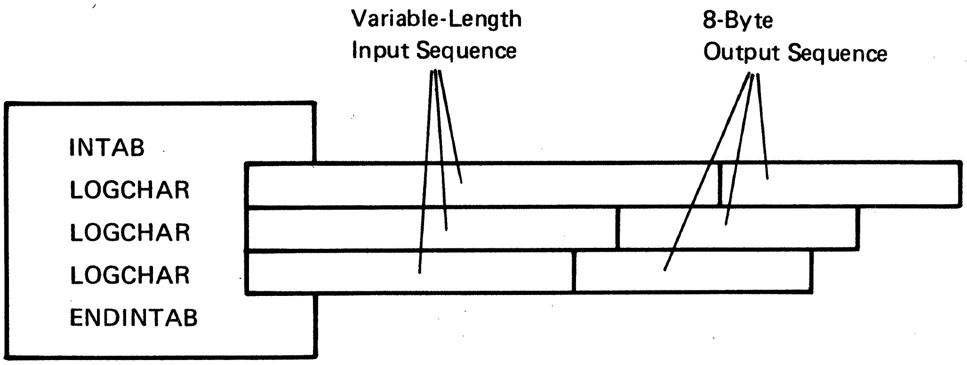


Figure 4-8. Interpret Table Macro Instructions

The interpret table function just described (resolving an input sequence from a terminal into an application program name) is included as part of the USS definition table function (resolving an input sequence into a logon mode name, user data, and an application program name). Therefore there is normally no need to combine the uses to which these tables are put; USS definition tables are typically used only for logical units, and interpret tables are typically used only for non-SNA terminals being serviced by the network solicitor or for general application program use. You can, however, use interpret tables to service the logons of logical units.

If the logical unit sends ACF/VTAM an Initiate Self command, ACF/VTAM uses the application program identifier portion of the command to search the interpret table and logs the logical unit on to the application program indicated by the table's output sequence. If no matching name is found in the interpret table, the application program name in the Initiate Self command is used. If there is no interpret table, the Initiate Self command must explicitly provide the application program name that would otherwise have been obtained from the table.

If the logical unit sends a character-coded logon, ACF/VTAM first attempts to use the interpret table, after removing media control characters, but before translating the command. If this is successful, the USSCMD and USSPARM portions of the USS definition table are not used to interpret the input sequence. You might choose to use this combined-table facility in order to maintain compatibility between logon messages to be sent by logical units that are not yet installed and logon messages that are already in use by non-SNA terminals. The following example explains why this combined use is necessary to achieve compatibility among logon messages:

You have defined an interpret table that allows the 2741 terminal operators to enter the following message:

request logon to application program 10

when they wish to become connected to an application program defined as APPL10. You want this identical logon message to be available to the operators of the newly installed 3767 terminals, which cannot build Initiate Self commands and therefore require a USS definition table.

As explained earlier in this chapter, a USS definition table is used to convert a character-coded logon into a formatted Initiate Self command only if the character-coded logon follows certain rules of syntax. Because the logon message above violates these rules, an interpret table must be used in conjunction with the USS definition table.

You use the same interpret tables to service both the 2741 and the 3767 terminals. When the character-coded logon is received, ACF/VTAM first uses the interpret table to convert "request logon to application program 10" into "APPL10" and then builds an Initiate Self command into which the name APPL10 is properly placed. This command cannot have a terminal-specified logon mode name or separate user data included in it, because these elements of an Initiate Self command can only be derived from the USS definition table, and the USS definition table cannot be used in this way due to the syntactically invalid logon message. Instead, a logon mode of blanks and a user data field consisting of the first 255 bytes of original input ('request logon to application program 10') is placed in the constructed Initiate Self command.

You do not have to use interpret tables. As explained above, interpret tables are normally not used for logons originating from logical units except when compatibility is needed between non-SNA and SNA logon messages, or when you do not want to insist that the application program names (as defined by APPL statements) be included as part of Initiate Self commands. All non-SNA terminals require an interpret table, however, if their logons are to be handled by the network solicitor, unless OS/VS logons are to be used.

Defining Interpret Tables

Each interpret table is defined by one INTAB macro instruction, followed by at least one LOGCHAR macro instruction, followed by one ENDINTAB macro instruction. Follow the ENDINTAB macro instruction with an assembler-language END statement unless the interpret table is to be followed by CSECTs containing one or more user-written APPLID routines, as described under "The LOGCHAR Macro Instruction." If two or more LOGCHAR macro instructions are used, they must be arranged so that their SEQNCE fields are in reverse collating order. For instance, if the two sequences MYPAY2 and MYPAY27 are to be defined, the LOGCHAR macro instruction specifying SEQNCE='MYPAY27' must precede that with SEQNCE='MYPAY2'.

The interpret table to be used for each terminal must be specified in the LOGTAB operand of the definition statement or macro instruction that defines the terminal. A unique interpret table can be defined for each terminal or the same table can be shared by several terminals. The way in which you relate terminals and interpret tables can be used to control access to application programs. A non-SNA terminal can log on only to application programs that are represented in the interpret table, for the terminal; furthermore, if two terminals use different interpret tables, the same logon message entered at both terminals can cause them to be logged on to different application programs.

Each interpret table must be link-edited to SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS).

The INTAB Macro Instruction

The INTAB (interpret table) macro instruction defines an interpret table that lists the ACF/VTAM application programs (including ISTOLTEP for TOLTEP) to which one or more terminals can establish connection. One INTAB macro instruction defines the name of the interpret table and a group of logon message definitions.

The format of the INTAB macro instruction is:

Name	Operation
[name]	INTAB

name

is 1 to 8 alphanumeric characters and must begin with an alphabetic character other than $ character. Specification of a name is optional. The name must be unique within the domain. This name should also be used as the operand for the assembler language END statement.

When the INTAB, LOGCHAR, and ENDINTAB macro instructions are assembled, this name is used to identify the entry point to the interpret table CSECT. This name can also be used as a member name for this interpret table when the linkage editor is being used to put the interpret table into SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS). This assignment prevents different names for the same interpret table (one for the entry point to the interpret table CSECT; and another as the name in the LOGTAB operand that other NCP macro instructions and ACF/VTAM definition statements use to refer to this interpret table).

The name assigned to the interpret table when it was put into SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS) by the linkage editor must be coded in the LOGTAB operand of the macro instructions and definition statements that define the terminals that will use this interpret table. (LOGTAB permits a terminal to initiate a logon and associates the specified interpret table with the terminal.)

The LOGCHAR Macro Instruction

Each LOGCHAR (logon characteristics) macro instruction defines a single logon message and either the name of an application program or the name of a logon-interpret routine. More than one LOGCHAR macro instruction can be included in an interpret table.

ACF/VTAM compares the logon message (character by character) with successive entries in the specified interpret table. If all the characters in the logon message correspond to characters in an entry in the interpret table, ACF/VTAM accepts the logon message as valid (even though the logon message can be longer than the corresponding entry in the interpret table). If the first character or characters of a logon message are identical, the LOGCHAR macro instructions should be arranged so that the logon sequences for the logon messages are from the most restrictive to the least restrictive. An example of this is:

```
SEQ1 LOGCHAR APPLID=(APPLICID,AP2),SEQNCE='LOG2'
SEQ2 LOGCHAR APPLID=(APPLICID,AP1),SEQNCE='LOG'
```

Otherwise, in the preceding example, if sequence LOG had preceded LOG2 in the interpret table, both logon messages LOG and LOG2 would be valid logons to application program API.

The format of the LOGCHAR macro instruction is:

Name	Operation	Operand

[name]	LOGCHAR	APPLID=(APPLICID,applname)
		APPLID=(ROUTINE,routinename)
		[,SEQNCE='characters']

name
is 1 to 8 alphameric characters and must not begin with a $ character; it is optional.

APPLID=(APPLICID,applname)
indicates the name of the application program, and is identical to the name assigned to the application program by an APPL definition statement.

APPLID=(ROUTINE,routinename)
indicates the routinename of the associated logon-interpret routine.

All logon-interpret routines specified in an interpret table must be assembled and link-edited with that interpret table.

SEQNCE='characters'
indicates either the required part of a terminal logon message or a logon from a Program Attention key on a terminal component of the IBM 3270 display system.

If 'characters' is a logon message, optional information, which is not specified in the LOGCHAR macro instruction, can be used by the logon-interpret routine (if the ROUTINE operand is specified), or by an application program's LOGON exit routine.

To continue the character string to another record in the SEQNCE operand, enter a nonblank character in column 72 and start the character string in column 16.

To specify an apostrophe or an ampersand within the logon message, code a double apostrophe or a double ampersand within the character string.

Do not specify a one-character logon message that is the same as an IBM 3270 AID (attention identification) character except to identify a logon message initiated by a 3270 Program Attention key. The AID characters are the numbers (1 to 9), the characters # @ = - Y ' _ % >, and the comma. Any other AID characters are treated as an Enter key.

To allow a terminal user to use lowercase letters in the logon message, the character string must be coded using the lowercase EBCDIC codes. This can be done on a cardpunch using multipunching.

Do not specify blanks as the first characters of a logon message because ACF/VTAM deletes leading blanks from logon messages entered by a terminal user. Therefore, logon messages with leading blanks would not match the logon message specified in the SEQNCE operand.

Do not specify leading and trailing device control characters within a character string that is to be validated by the IBM network solicitor, because the IBM network solicitor deletes these characters. (For OS/VS2 MVS only, this function can be controlled with the STRPCNL operand of the NETSOL macro instruction.)

Do not specify device control characters within a character string that is to be validated by the IBM-supplied network solicitor, which changes these characters in a logon message to blanks. Code a blank within the character string to represent each embedded device control character.

For non-SNA terminals, the length of the entire logon message (required information plus optional information) is limited to the number of characters that can be accommodated on one line of the terminal being logged on to a maximum of 255 characters. For SNA terminals, the length is limited to 255 characters (ACF/VTAM deletes new line [NL] characters before the INTRPRET macro instruction is completed). The maximum allowable logon sequence to NETSOL is 80 characters, including device control characters.

If "characters" is a logon message from one of the 3270 Program Attention keys (the Program Function [PF] keys and the Program Access [PA] keys, the character string is the one-character AID (attention identifier) character that corresponds to the appropriate PF or PA key pressed by the terminal user. The AID characters are the numbers (1 to 9), the characters # @ = Y ' _ % >, and the comma. Any other AID characters are treated as an Enter key.

If SEQNCE is not coded in one LOGCHAR macro instruction and if a logon message does not match the character string of the SEQNCE operand in a preceding LOGCHAR macro instruction in the interpret table, ACF/VTAM accepts this logon message and requests logon for this terminal to the application program specified in the LOGCHAR macro instruction (the one in which SEQNCE is not coded). Therefore, do not place a LOGCHAR macro instruction at the beginning of the interpret table (immediately following the INTAB macro instruction) without coding the SEQNCE operand. Otherwise, the remaining logon messages in the interpret table are not compared with the logon message entered by the terminal user.

The ENDINTAB Macro Instruction

The ENDINTAB (end interpret table) macro instruction defines the end of an interpret table. Code one ENDINTAB macro instruction after one or more LOGCHAR macro instructions to define the end of an interpret table. The ENDINTAB macro instruction can also be followed by an assembler language END statement or by CSECTs containing one or more user-written logon-interpret routines.

The format of the ENDINTAB macro instruction is:

Name	Operation
[name]	ENDINTAB

name

is 1 to 8 alphameric characters and must not begin with a $ character. The specification of a name is optional.

If an assembler language END statement is coded, it must be in the format:

END name

where name is the label of the INTAB macro instruction and specifies the main entry point.

Coding Logon-Interpret Routines (APPLID Routines)

You can code logon-interpret routines to validate logons and determine the name of the application program which is to receive the logons. ACF/VTAM uses the logon-interpret routine if the interpret table specifies the name of the logon-interpret routine. The routine's name must match the routinename specified in the APPLID=(ROUTINE, routinename) operand in the LOGCHAR macro instruction. All logon-interpret routines specified in an interpret table must be assembled and link-edited with that interpret table.

Because the logon-interpret routine receives the logon message as input from a terminal, the logon message can contain more data that is specified in the interpret table (the character string specified in the SEQNCE operand of the appropriate LOGCHAR macro instruction). This additional data can contain the name of the application program that is to receive the logon or a password for the logon-interpret routine to verify.

For example, if SEQNCE='CREDIT STATUS' is specified in the interpret table, and a terminal user enters logon message:

CREDIT STATUS MYID=KMG01

CREDIT STATUS is the logon message to initiate connection to the application (determined by the logon-interpret routine). MYID=KMG01 is additional data that can be a password for the logon-interpret routine or application program to verify.

Logon-Interpret Routine Requirements

Entry From: ACF/VTAM to entry point routinename.

Registers at Entry:

Register 0:	Length of logon message
Register 1:	Address of first byte of logon message
Register 13:	Address of save area provided by ACF/VTAM
Register 14:	Return address
Register 15:	Address of entry point of this routine

Operation: The logon interpret routine is executed synchronously under the control of ACF/VTAM and not under the control of an application program. For the application program to receive the logon, this routine must validate the logon, obtain the symbolic name of the application program to receive control, and provide this name to ACF/VTAM. Otherwise, the routine specifies that the logon is invalid or that the name of the application program was not found.

The logon-interpret routine must also:

Save and restore the contents of registers 2 through 14 when receiving and passing control

Use reenterable code (the routine must not store anything within itself or modify itself during execution)

Perform no I/O operations; otherwise an I/O request causes the routine to abnormally terminate.

Registers at Exit: Registers 0 and 1 contain the name of application program (in EBCDIC characters) to which the terminal is to be connected:

Register 0:	First 4 characters of name (left-justified).	
Register 1:	Last 4 characters of name (left-justified).	
Registers 2-14:	Restored to condition at entry	
Register 15:	Return code:	
	00	Application program was found and the name placed in registers 0 and 1
	Nonzero	Application program was not found and the name is not placed in registers 0 and 1

If the name of the application program contains fewer than 8 characters, use blanks to provide a name with 8 characters.

Installing And Changing Interpret Tables

Follow these steps to install each interpret table and any user-written routines:

1. Assemble the interpret table and the user-written routines referred to by the LOGCHAR macro instruction. (The user-written routines can be placed in a private call library as described in *OS/VS Linkage Editor and Loader.*)

2. Link-edit the interpret table with its associated routines, preferably assigning a module name that matches the name of the interpret table as specified with the INTAB macro instruction. (These changes must be made before ACF/VTAM is started.)

3. Code the assigned name in the LOGTAB operand of the appropriate macro instruction or definition statement to associate the terminal or terminals with this interpret table (see Chapter 2 for information on specifying the LOGTAB operand in macro instructions and definition statements).

To add a LOGCHAR macro instruction to an installed interpret table or to replace an installed interpret table, add or replace the LOGCHAR macro instruction in the source deck, assemble and link-edit the new interpret table, replacing the old module. (These changes must be made before ACF/VTAM is started.)

The control sections of the interpret table module consist of the interpret table itself and a CSECT for each user-written routine identified by the ROUTINE operand of the LOGCHAR macro instruction. Individual CSECTs can be extracted when the interpret table is link-edited. Because reprocessing a load module deletes the END statement, use the linkage editor ENTRY control statement to specify the entry point of the new load module. For linkage editor requirements, refer to *OS/VS Linkage Editor and Loader.*

To change the name of an installed interpret table:

1. Change the name specified on the INTAB macro instruction and follow the procedure outlined under "Changing an Interpret Table." When link-editing the interpret table, the module name and entry point must be made to match the new name of the interpret table.
2. Change the LOGTAB operand in all affected macro instructions and definition statements.
3. File the corrected NCP source deck and the corrected set of definition statements in SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS), replacing the old source decks. (The procedure is described in Chapter 2.) It is not necessary to do an NCP partial generation because the LOGTAB operand is used only by ACF/VTAM.

Example Of Defining An Interpret Table

In this example:

An interpret table named TABLE07 and its two associated logon-interpret routines are defined (coded), assembled, and link-edited.

Five logon messages are defined in the interpret table for four application programs.

Two logon messages that refer to two logon-interpret routines are defined in the interpret table.

The terminal that refers to any of the entries in interpret table TABLE07 has the operand LOGTAB=TABLE07 specified in the appropriate ACF/VTAM definition statement or NCP macro instruction.

```
//INTABASM JOB MSGLEVEL=1
//ASMSTEP EXEC ASMFCL,PARM.LKED='XREF,LIST,'
//ASM.SYSIN DD *
TABLE07 INTAB
MSG1 LOGCHAR APPLIC=(APPLICID,APPL1),SEQNCE='NEW ORDER'
MSG2 LOGCHAR APPLID=(APPLICID,APPL2),SEQNCE='ORDER INQUIRY'
MSG3 LOGCHAR APPLID=(APPLICID,APPL1),SEQNCE='CHANGE ORDER'
MSG4 LOGCHAR APPLID=(ROUTINE,CHKLOGON),SEQNCE='CREDIT STATUS'
MSG5 LOGCHAR APPLID=(ROUTINE,UPDCRDIT),SEQNCE='UPDATE CREDIT'
ENDINTAB
```

```
CHKLOGON CSECT
   .
   .
   .
(instructions in routine)
   .
   .
   .
UPDCRDIT CSECT
   .
   .
   .
(instructions in routine)
END TABLE07
/*
```

Control Statements For Example

The INTAB macro instruction defines the interpret table. The name of the macro instruction (TABLE07) represents (1) the name of the interpret table, (2) the name of the interpret table CSECT, and (3) the main entry point to the interpret table.

The LOGCHAR macro instruction, MSG1, defines the terminal logon message NEW ORDER to initiate connection to application program APPL1.

The LOGCHAR macro instruction, MSG2, defines the terminal logon message ORDER INQUIRY to initiate connection to application program APPL2.

The LOGCHAR macro instruction, MSG3, defines the terminal logon message CHANGE ORDER to also initiate connection to APPL1.

The LOGCHAR macro instruction, MSG4, defines the terminal logon message CREDIT STATUS to initiate connection to the application program determined by logon-interpret routine CHKLOGON.

The LOGCHAR macro instruction, MSG5, defines the terminal logon message UPDATE CREDIT to initiate connection to the application determined by logon-interpret routine UPDCRDIT.

The ENDINTAB macro instruction defines the end of interpret table TABLE07.

The logon-interpret routine CHKLOGON CSECT assembler language statement identifies the name of this logon-interpret routine and indicates that the following instructions are part of this control section.

The logon-interpret routine UPDCRDIT CSECT assembler language statement identifies the name of this logon-interpret routine and indicates that the following instructions are part of this control section.

The assembler language END statement is the last statement in the interpret table source program named TABLE07. (The assembler produces an object module and an END statement for the module. The assembler-produced END statement contains an entry point only if the source language END statement contained one.)

The Network Solicitor

A network solicitor is a logon monitor facility for interactive start-stop and BSC terminals (including local non-SNA terminals). The network solicitor can be used to permit terminal-initiated logons for these devices. Therefore, if a terminal assigned to the network solicitor is active, and not connected or queued for connection to an application program, the network solicitor can accept the terminal's logon. After accepting the logon, the network solicitor uses either an interpret table or OS/VS logon to validate the logon message and pass it to the appropriate application program. The LOGON sequence can be all uppercase, all lowercase, or a combination of both, but must agree with the interpret table.

The IBM-supplied network solicitor passes the logon message to the application program with the following modifications (the modifications are made in the order indicated):

Any characters in excess of 80 characters are stripped off.

Leading blanks are stripped off.

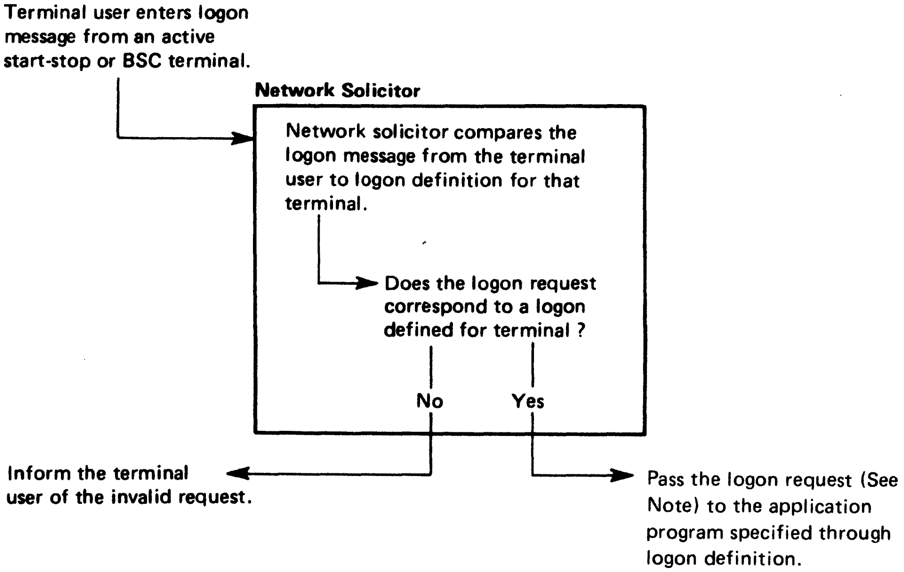
Leading and trailing device control characters are stripped off. (In OS/VS2 MVS only, this function can be suppressed.)

If the logon message is handled as an OS/VS logon message lowercase letters are converted to uppercase and all characters with a hexadecimal value less than X'4A' are converted to blanks.

The IBM network solicitor can be used to monitor any non-SNA terminal that ACF/VTAM supports, including a non-SNA 3270 that logs onto application programs that use record mode.

If the IBM-supplied network solicitor is monitoring a terminal and detects a hardware I/O error for the terminal, it issues a message to the network operator and stops monitoring the terminal (by issuing a CLSDST macro instruction with the RELEASE processing option specified to disconnect the terminal from the application program).

Figure 4-9 shows how the network solicitor processes a terminal-initiated logon message.


Terminal user enters logon message from an active start-stop or BSC terminal.
Network Solicitor
Network solicitor compares the logon message from the terminal user to logon definition for that terminal.
Does the logon request correspond to a logon defined for terminal ?
No
Yes
Inform the terminal user of the invalid request.
Pass the logon request (See Note) to the application program specified through logon definition.

Note: The network solicitor passes the logon request only if the application program has an open ACB with MACRF=LOGON specified and has issued a SETLOGON macro instruction with OPTCD=START specified.

Figure 4-9. Network Solicitor Functions

A system programmer who needs a network solicitor has these choices; use the IBM supplied-network solicitor, use the NETSOL macro instruction to generate a modified network solicitor, or write an application program to perform network solicitor functions.

IBM-Supplied Network Solicitor

When you include ACF/VTAM during system generation, it contains the IBM-supplied network solicitor. This network solicitor has the following values in effect, which are the default values of the operands of the NETSOL macro instruction:

NETSOL is the name of the network solicitor.

MSGCSECT is the name of the IBM-supplied message CSECT that contains network solicitor messages.

The IBM-supplied network solicitor messages are sent to the terminal user when an error is detected.

Application programs can acquire terminals from the network solicitor by using the SIMLOGON macro instruction.

No passwords can be verified by this network solicitor.

Ten RPLs are allocated for asynchronous operations.

The maximum allowable logon sequence to NETSOL is 80 characters, including device control characters.

To use the IBM-supplied network solicitor:

Specify the start option NETSOL=YES to load the network solicitor when starting ACF/VTAM or use the MODIFY command with NETSOL=YES to start the network solicitor after ACF/VTAM has been started.

Code LOGAPPL=NETSOL on the macro instructions and definition statements that define the devices to be controlled by this network solicitor.

If OS/VS logon is not to be used, code the appropriate interpret tables and put them in SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or in SYS1.LPALIB (for OS/VS2 MVS). Then specify the LOGTAB operand with the appropriate interpret tabe name in the definition statements and macro instructions the define the devices using this interpret table.

File or enter the start option NETSOL=YES to load the network solicitor when ACF/VTAM is started. Use the MODIFY command specifying NETSOL=YES to start the network solicitor at any time after ACF/VTAM is started.

Modifying The IBM-Supplied Network Solicitor

If you do not want to use the IBM-supplied network solicitor, you can generate a modified network solicitor by coding, assembling, and link-editing the NETSOL macro instruction. The modified network solicitor can either replace, or be used in addition to, the IBM-supplied network solicitor. If the modified network solicitor is a replacement, it must be link-edited into SYS1.VTAMLIB (for OS/VS1 or OS/VS2 SVS) or SYS1.LPALIB (for OS/VS2 MVS). If the modified network solicitor is to be executed concurrently with the IBM-supplied network solicitor, it can be link-edited into a user-defined data set.

For information on what operands can be modified, see the description of the NETSOL macro instruction in this chapter.

If the network solitor is to run in the same partition (OS/VS1) or region (OS/VS2) as ACF/VTAM, the macro instruction label must be coded as NETSOL. The PRTCT operand is ignored.

To install the network solicitor, assemble the macro instruction with the replacement message CSECT, if one was coded, and link-edit the object modules to SYS1.LINKLIB under the load module name ISTNSC00, replacing the IBM-supplied network solicitor module. Link-edit the network solicitor before starting ACF/VTAM.

Ensure that the network solicitor's TPEND exit routine and ACB closing mechanism remain intact. (The TPEND exit routine is scheduled when the MODIFY NETSOL=NO command is issued; until the network solicitor's ACB is closed, no further network operator commands can be issued.)

Other considerations are identical to those listed under the "IBM-Supplied Network Solicitor."

Using More Than One Network Solicitor

If you plan to use more than one network solicitor, whether one at a time or concurrently, only one version can be named NETSOL.

For each additional network solicitor, assemble the macro instruction with the replacement message CSECT, if one was coded, and link-edit the object modules to SYS1.LINKLIB under the user-assigned load module name which must not be ISTNSC00. Link-edit the network solicitor before starting ACF/VTAM.

Code the name of the network solicitor in the LOGAPPL operand in the configuration macro instructions (including LOCAL statements) that define devices that are to be logged on to the network solicitor when it is activated.

Code and file an APPL statement to define the additional network solicitor as an application program. Code the PRTCT operand of the APPL statement if the PRTCT operand is being coded in the NETSOL macro instruction. Code the AUTH operand with the PASS option and the ACQ option. (The ACQ option must be coded because the network solicitor used the OPNDST macro instruction with the ACQUIRE option.)

Unless OS/VS logon is to be used, code and file an interpret table as described earlier in this chapter; code the LOGTAB operand in the appropriate macro instructions.

Code the start option NETSOL=NO, or allow it to be assumed by default. The MODIFY command specifying NETSOL=YES or NO applies only to the network solicitor named NETSOL with load module name ISTNSC00.

The requirements for starting the alternate network solicitor are the same as for any application program. The APPL statement defining the network solicitor must be contained in a major node that has been activated. Then the network solicitor is started and stopped using network operator commands.

NETSOL Macro Instruction

Use the NETSOL macro instruction to modify or replace the IBM-supplied network network solicitor or to create additional network solicitors. The following explanation of the NETSOL operands describes the characteristics of the IBM-supplied network solicitor as well as the modifications available through coding the NETSOL macro instruction. Write the NETSOL macro instruction as follows:

Name	Operation	Operand
[NETSOL name]	NETSOL	SYSTEM=VS1 | VS2 | SVS [,ACQUIRE=YES | NO] [,MESSAGE=csectname | IBMMSG] [,NUMBER=n | 10] [,PRTCT=password] [,STRPCNL=YES | NO] (For OS/VS2 MVS only) [,TIME=n | 5]

NETSOL name

specifies the name of the network solicitor. Code NETSOL or omit the name unless this network solicitor is to be used with one or more additional network solicitors. In this case, only one of the network solicitors can be named NETSOL. If name is other than NETSOL, it can be any symbol valid in assembler language, but must be the same as the name specified on the APPL definition statement that defines the network solicitor application program.

Only a network solicitor named NETSOL can be started or stopped with network operator commands. If a network solicitor is to run in a partition (OS/VS1) or region (OS/VS2) other than the one in which ACF/VTAM is to run, that network solicitor must not be named NETSOL.

SYSTEM=VS1 | VS2 | SVS

Specifies the operating system in which the network solicitor is to run. This operand is required. VS1 refers to OS/VS1, VS2 refers to OS/VS2 MVS, and SVS refers to OS/VS2 SVS.

ACQUIRE=YES | NO

specifies whether the network solicitor is to be notified to release a terminal when the terminal is requested by an application program that issues the SIMLOGON macro instruction with the RELREQ option.

If YES is specified or the ACQUIRE operand is omitted, the network solicitor contains a RELREQ exit routine that is queued for execution to notify the network solicitor of the SIMLOGON request.

If the requested terminal is still owned by the network solicitor at the time that the RELREQ exit routine is entered, the terminal is released.

If NO is specified, the network solicitor is not notified when an application program attempts to acquire a terminal by using SIMLOGON.

For details of the RELREQ exit on the EXLST macro instruction and on the SIMLOGON macro instruction, refer to *ACF/VTAM Macro Language Reference.*

If application programs are authorized to use the SIMLOGON macro instruction, ACQUIRE=YES should be coded. (The application program authorization is provided in the APPL statement.)

ACQUIRE=YES permits a remote possibility of a terminal user's losing a terminal to one application program while attempting to log on to another application program.

If a terminal user enters a logon message after an application program issues a SIMLOGON macro instruction but before the network solicitor's RELREQ exit routine releases the terminal to the application program, the user's logon message will appear to be accepted. However, the terminal will be connected to the application program before the network solicitor can process the logon message. This situation can also occur if the user makes an error in a logon message, receives an error message from the network solicitor, and reenters a corrected logon message. That is, the terminal can be connected to an application program that issued a SIMLOGON macro instruction before the network solicitor can process the corrected logon message.

MESSAGE=csectname | IBMMSG

specifies the name of the control section (in the network solicitor load module) that contains error messages that can be sent to the terminal user when certain errors are detected.

Code MESSAGE=csectname when the named control section is to replace the message control section supplied by IBM. The messages in the replacement control section can be shortened, expanded, or translated versions of the messages in the IBM-supplied control section.

If this operand is omitted or if IBMMSG is specified, the IBM-supplied CSECT, named MSGCSECT, is assembled with the network solicitor. (A V-type address constant to the IBM-supplied CSECT, MSGCSECT, is created when the NETSOL macro instruction is assembled.) The twelve standard message texts of MSGCSECT are shown in Figure 4-10; refer to the twelve DC statements labeled AMSG01 through AMSG12.

For an explanation of the error conditions that produce the messages, refer to Appendix B.

The message control section contains a table consisting of the lengths and offsets of all the messages, followed by the text for each message. The network solicitor uses indexing to refer to the table for each of the situations requiring a message.

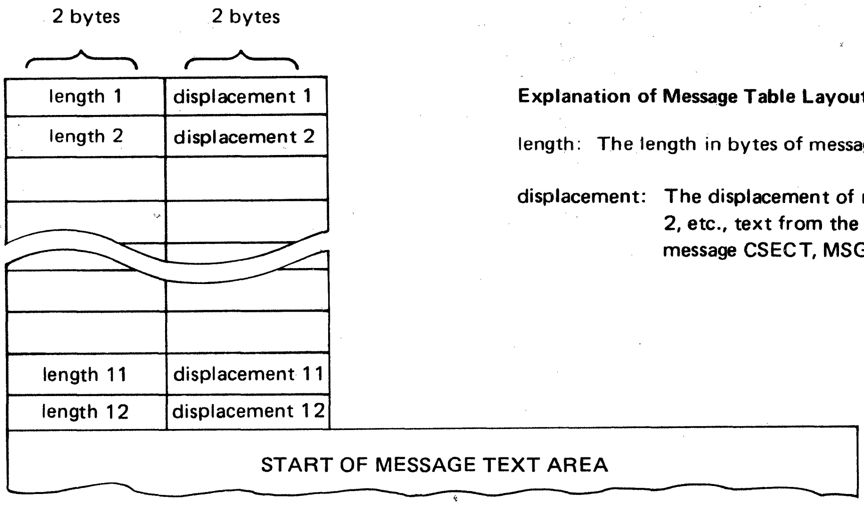
When replacing MSGCSECT, the name MSGCSECT or any other user-selected name (except IBMMSG) can be used for the csectname coded with the MESSAGE operand. (The name MSGCSECT is recommended in order to reduce coding changes in the IBM-supplied CSECT.) The format of the table of lengths and offsets must be identical to the format shown in Figure 4-10. (However, if the name of the CSECT is changed from MSGCSECT, the new name should be substituted in the first twelve DC statements for the name MSGCSECT.) Likewise, each message statement label must be coded as shown in Figure 4-10. Only the message text within the single quotation marks should be changed.

RESTRICTION: The maximum recommended message length (between single quotation marks of the DC statement) is 100 characters. However, fewer characters might have to be used if the message is to be sent to a terminal that has a line length of less than approximately 128 characters. (If all messages are limited to 47 characters, buffer-size and line-length restrictions for all NETSOL-supported devices will be satisfied.)

Assemble the replacement control section together with the NETSOL macro instruction and then link-edit the resulting object module to SYS1.VTAMLIB.

NUMBER=n | 10

specifies the number of terminal operators whose logon processing can be done concurrently. The specified number becomes the maximum number of asynchronous



Explanation of Message Table Layout

length: The length in bytes of message 1, 2, etc.

displacement: The displacement of message 1, 2, etc., text from the start of the message CSECT, MSGSECT.

The control section that defines the IBM-supplied message texts is shown below:

```
MSGCSECT CSECT
         DC    AL2(L'AMSG01,AMSG01-MSGCSECT)
         DC    AL2(L'AMSG02,AMSG02-MSGCSECT)
         DC    AL2(L'AMSG03,AMSG03-MSGCSECT)
         DC    AL2(L'AMSG04,AMSG04-MSGCSECT)
         DC    AL2(L'AMSG05,AMSG05-MSGCSECT)
         DC    AL2(L'AMSG06,AMSG06-MSGCSECT)
         DC    AL2(L'AMSG07,AMSG07-MSGCSECT)
         DC    AL2(L'AMSG08,AMSG08-MSGCSECT)
         DC    AL2(L'AMSG09,AMSG09-MSGCSECT)
         DC    AL2(L'AMSG10,AMSG10-MSGCSECT)
         DC    AL2(L'AMSG11,AMSG11-MSGCSECT)
         DC    AL2(L'AMSG12,AMSG12-MSGCSECT)
AMSG01   DC    C'ERROR WHEN READING LOGON MESSAGE'
AMSG02   DC    C'INPUT NOT RECOGNIZED'
AMSG03   DC    C'NO INTERPRET TABLE'
AMSG04   DC    C'NO ROUTINE TO RECOGNIZE APPLICATION'
AMSG05   DC    C'USER UNAUTHORIZED FOR THIS APPLICATION'
AMSG06   DC    C'APPLICATION UNKNOWN TO VTAM'
AMSG07   DC    C'APPLICATION DEACTIVATED BY NETWORK OPERATOR'
AMSG08   DC    C'APPLICATION IS INACTIVE'
AMSG09   DC    C'APPLICATION IS CLOSING DOWN'
AMSG10   DC    C'APPLICATION NEVER ACCEPTS LOGONS'
AMSG11   DC    C'APPLICATION IS NOT ACCEPTING LOGONS'
AMSG12   DC    C'THIS TERMINAL IS LOGGED ON TO THE NETWORK SOLICITOR'
         END
```

Figure 4-10. The IBM-Supplied Message CSECT

RPLs and associated work areas that the network solicitor can use.

Specify as an decimal integer in the range 2 through 255.

PRTCT=password

Specifies a password of up to eight characters that will be compared with the password for a network solicitor defined by an APPL statement. Matching of the passwords establishes that the network solicitor has authority to run as the application program defined in the APPL statement.

Omit this operand if NETSOL is the name of the network solicitor. The password is meaningful only if an APPL statement is filed to define the network solicitor and a password is specified in the APPL statement. (The APPL statement is required if a name other than NETSOL is assigned to the network solicitor.) This operand must be coded to match the password coded in the PRTCT operand of the APPL statement.

STRPCNL=YES | NO (For OS/VS2 MVS only)

indicates whether or not the network solicitor is to strip device-control characters from logon messages before it passes the messages to the application programs.

STRPCNL=NO is useful when all of the application programs to be handled by the network solicitor need the device control characters.

TIME=n | 5

Indicates the time (in minutes) between when the network solicitor is initiated and when ACF/VTAM begins to send the network solicitor logon messages to the 3270 devices that have been automatically logged onto the network solicitor. This option, which delays the sending of the logon messages, can be used to prevent shortages of system resources during network solicitor initialization.

User-Written Network Solicitor

Using ACF/VTAM macro instructions, you can write an application program to perform the services of a network solicitor. These macro instructions are described in *ACF/VTAM Macro Language Reference.*

The IBM-supplied network solicitor can be used for guidance. A copy can be obtained by listing or assembling the macro instruction.

The user-written program can be named NETSOL and the load module can be named ISTNSC00 if the user wishes to replace the IBM-supplied network solicitor and control the network solicitor using the START or MODIFY commands in the same way they are used for the IBM-supplied network solicitor. You must ensure that the user-written program named NETSOL has a TPEND exit routine, and that this exit routine promptly causes the program's ACB to be closed. (The TPEND exit routine is scheduled when the MODIFY NETSOL=NO command is issued; until the program's ACB is closed, no further network operator commands can be issued.)

If the user-written program will be used concurrently with the IBM-supplied network solicitor, the user-written network solicitor CSECT must not be NETSOL and the load module name must not be ISTINSC00. Then, the user-written program is controlled as an application program.

Example Of Adding A Network Solicitor To SYS1.VTAMLIB

In this example, IBM-supplied cataloged procedures are used to assemble and link edit additional network solicitor into SYS1.VTAMLIB (as load module MYNETSOL).

```
  //ADDSOL JOB MSGLEVEL=1
  //ASMSTEP EXEC ASMFCL,PARM.LKED='XREF,LIST'
  //ASM.SYSIN DD *
MYNETSOL NETSOL MESSAGE=IBMMSG
  /*
  //LKED.SYSLMOD DD DSN=SYS1.VTAMLIB(MYNETSOL),DISP=OLD, . . .
```

Control Statements For Example

The ACF/VTAM NETSOL macro instruction (1) defines the name of the additional network solicitor (MYNETSOL), (2) specifies that the IBM-supplied CSECT (MSGCSECT) is to be used, creating a V-type address constant to MSGCSECT, (3) indicates by default that the network solicitor contains a RELREQ exit routine, and (4) has no password because it is not defined by an APPL definition statement.

Chapter 5. ACF/VTAM Services

ACF/VTAM provides services that allow the user to execute exit routines to provide interfaces to user-written authorization and accounting exit routines.

Exit Routines

You can provide exit routines for authorization and accounting administration. These routines can be installed at any time before ACF/VTAM is started. For an overall discussion of ACF/VTAM integrity facilities, see *ACF/VTAM Concepts and Planning.*

If you do not provide exit routines, ACF/VTAM-supplied dummy modules are invoked. These dummy modules simply return control to the caller (ACF/VTAM), except that a successful return code is set before the return in the case of the authorization routine. This section describes the conventions to be followed in preparing and installing an authorization or accounting exit routine to replace the IBM-supplied dummy modules.

Authorization Exit Routines

You can write an authorization exit routine to be invoked before ACF/VTAM services a connection request. The purpose of the routine is to check on or restrict the use of an application program or a terminal.

ACF/VTAM makes available in a parameter list the type of I/O request to be authorized and the names of the terminal and the application program to be connected or queued. This exit routine can contain a table of valid or invalid requests. The information in the table can be compared with each connection or logon request as recorded in the parameter list. If the request is valid, the routine must place zero in register 15 before returning to ACF/VTAM. Any nonzero return code indicates that the request is not authorized for completion.

For example, an application program can be permitted to connect to the first available terminal (OPNDST with ACQUIRE and CONANY options). The authorization exit routine can compare the available terminal with a table provided in the routine to determine if authorization will be granted for the available terminal, or for a particular available terminal at a particular time of day.

Invoking The Authorization Exit Routine

The authorization exit routine is invoked before ACF/VTAM services any connection request and whenever a terminal is to be queued as the result of a logon request. For example, this routine is invoked as a result of an OPNDST or SIMLOGON macro instruction in an application program; as a result of TOLTEP attempting to make a terminal connection by using OPNDST with the ACQUIRE option, or as a result of an automatic logon or network operator logon.

Initial Register And Parameter List Contents

When this user-written routine gains control, register contents are as follows:

Register 1:	Address of a parameter list (described in Figure 5-1)
Register 13:	Address of the save area for use by this routine
Register 14:	Return address
Register 15:	Address of the entry point of this routine

Offset Dec Hex	Size (bytes) Alignment (bits)	Description
0 (0)	1	Reserved
1 (1)	1	Contains a decimal number from 1 through 6 that identifies the request: **Number** **Request** 1 SIMLOGON or OPNDST macro instruction with ACQUIRE option. (See offsets 2 and 3 to determine if CONALL or CONANY option was also specified.) 2 Reserved. 3 OPNDST macro instruction with ACCEPT option. (See offset 2 to determine if ANY option was also specified.) 4 Logon request initiated by the network operator by issuing a VARY ACT command or VARY LOGON command; or a connection request for a CDRSC. (See offset 8 for the name of the application program or the CDRSC for which the logon is requested.) 5 CLSDST macro instruction with PASS option. (See offset 4 for pointer to name of application program issuing request and offset 8 for pointer to name of application program for which logon is requested.) 6 OPNSEC macro instruction.
2 (2)	1 1	If offset 1 is 1, CONALL or CONANY option is specified in either OPNDST macro instruction with ACQUIRE processing option or SIMLOGON macro instruction. **or** If offset 1 is 3, ANY option is specified on OPNDST macro instruction with ACCEPT processing option.
	. x.	Reserved.
	. . 1.	If offset 1 is 1, SIMLOGON macro instruction request.
	. . . x xxxx	Reserved.
3 (3)	1	
	x.	Reserved.
	. 1.	If offset 1 is 1, CONALL option, not CONANY option, is specified in either OPNDST macro instruction with ACQUIRE processing option or SIMLOGON macro instruction. (See offset 2 to determine which macro instruction is specified.)
	. . xx xxxx	Reserved.
4 (4)	4	0 or address of doubleword with name of application program that issued request. (Not applicable if offset 1 is 4 or 7.)
8 (8)	4	0 or address of doubleword with name of application program for logon requests. (Valid logon request if offset 1 is 1 (SIMLOGON macro instruction), 4, or 5.)
12 (C)	4	Identifies terminals associated with request: • If field is 0, request is from OPNDST macro instruction with ACCEPT and ANY options; no terminals associated with this request. • If field is not 0, this field contains address of doubleword with a terminal name.
16 (10)	2	Hexadecimal number of terminals.

Figure 5-1. Parameter List for Authorization Exit Routine

Full explanations of the macro instructions and options referred to in Figure 5-1 appear in *ACF/VTAM Macro Language Reference.*

Design Restrictions And Suggestions

Follow these restrictions and suggestions when writing the routine:

The name of the module must be ISTAUAAT (for OS/VS1 and OS/VS2 SVS) or ISTAUCAT (for OS/VS2 MVS).

Use standard linkage.

Save registers 0-14.

The routine operates as an internal ACF/VTAM subroutine. Therefore, performance is degraded if the routine requires lengthy processing time. While this routine is being executed, no new connection, disconnection logon, or activation requests are processed by ACF/VTAM for any application program. System waits, including implied waits for I/O operations, should be avoided if possible.

This routine operates enabled in pageable storage. Because the routine operates at the current task's dispatching priority, there is a possibility of lockout if a wait requires other task action. The routine gets control in supervisor state and with an ACF/VTAM protection key of 0 (for OS/VS2 MVS, the protection key is 6) so errors within the routine could cause damage to ACF/VTAM or system control blocks and modules. The routine is executed under the invoking program's task, which can be the ACF/VTAM task. In the case of a VARY command, the routine executes under the ACF/VTAM task.

ACF/VTAM macro instructions must not be used in the routine.

The parameter list pointed to by register 1 (the 24-byte area described in Figure 5-1) must not be modified by this routine. Neither can any field pointed to from the parameter list be modified.

The routine must supply a return code to ACF/VTAM in register 15. A return code of zero authorizes the connection to be made. Any nonzero return code means that the request is not authorized. If the request is not authorized, ACF/VTAM informs the application program by setting the RTCND and FDBK2 fields in the RPL.

RTNCD is set to X'14'.
FDBK2 is set to X'55'.

This routine is notified of requests made by the network solicitor. Therefore, if you are planning to use the network solicitor to monitor terminals for logons, the authorization routine should authorize all requests from the network solicitor, including pass requests and acquire requests. ACF/VTAM's network solicitor uses the PASS option of the CLSDST macro instruction to pass valid terminal-initiated logons to active application program. If ACQUIRE=YES is specified in the NETSOL macro instruction, giving the network solicitor the capability to release upon request terminals that it is monitoring, this authorization exit routine should authorize the network solicitor to issue a CLSDST macro instruction with the RELEASE option.

The routine is notified if ACF/VTAM's terminal online test program (TOLTEP) attempts to acquire a terminal. This routine should authorize all TOLTEP requests.

Failure of the authorization exit routine to honor requests of the type discussed here limits the available ACF/VTAM services accordingly.

ACF/VTAM invokes the exit routine before servicing all pending connection requests. This means that the exit routine is invoked twice before a connection is made, except for OPNDST ACQUIRE and REQSESS, for which the exit routine is invoked only once.

The authorization routine can be interrupted if a user task is abnormally terminated (for example, cancelled by the operator).

Final Register Contents

The routine must leave the register status as follows:

Registers 0-14:	Restore these registers.
Register 15:	General return code of zero if the request is authorized. Any nonzero return code if the request is not authorized.

Refer to the preceding section, "Design Restrictions and Suggestions," for information about RTNCD and FDBK2 return codes set in the RPL of the application program.

Accounting Exit Routines

You can write an accounting exit routine to be invoked whenever a connection is made or broken. A separate invocation is made for each session. The purpose of the routine is to keep statistics on the ACF/VTAM connections so that the users of the application programs and terminals can be charged appropriately.

ACF/VTAM makes available the name of the application program, the name of the terminal, and whether the request is a connection or a disconnection. The exit routine can record the time a connection is initiated. When the connection is being broken, the routine can again record the time of the disconnection. The difference between these two times is the connection time for the terminal and that application program. To reduce the effect on ACF/VTAM's performance, the information collected should be analyzed later by another program.

Initial Register Contents

When this user-written routine gains control, register contents are as follows:

Register 0:	Positive if a connection has been made; negative if a disconnection has occurred
Register 7:	Pointer to the doubleword containing the name of the primary logical unit
Register 11:	Pointer to the doubleword containing the name of the secondary logical unit
Register 13:	Address of the save area for use by this routine
Register 14:	Return address
Register 15:	Address of the entry point of this routine

Design Restrictions And Suggestions

Follow these restrictions and suggestions when writing the routine:

The name of the module must be ISTAUAAG (for OS/VS1 and OS/VS2 SVS) or ISTAUCAG (for OS/VS2 MVS).

Use standard linkage.

Save registers 0-14.

This routine operates enabled in pageable storage. Because the routine operates at the current task's dispatching priority, there is a possibility of lockout if a wait requires other task action. The routine gets control in supervisor state with an ACF/VTAM protection key of 0 (For OS/VS2 MVS, the protection key is 6) so that errors within the routine could cause damage to ACF/VTAM or system control blocks and modules. The routine is executed under the invoking program's task, in some cases this is the ACF/VTAM task.

The routine operates as an internal ACF/VTAM subroutine. Therefore, performance is degraded if the routine requires lengthy processing time. While this routine is being executed no new connection, disconnection, logon, or activation requests are processed by ACF/VTAM for any application program. System waits, including implied waits for I/O operations, should be avoided if possible.

The routine is notified of connection and disconnection requests involving terminals, TOLTEP, and the network solicitor. The routine should be designed to process requests involving these facilities.

ACF/VTAM macro instructions must not be used in the routine.

Fields pointed to by registers 7 and 11 must not be modified.

The accounting routine can be interrupted if a user task is abnormally terminated (for example, cancelled by the operator).

Final Register Contents

All general purpose registers except register 15 must be restored. No return code is expected by ACF/VTAM.

Installing Exit Routines

Follow these steps to install the authorization and accounting exit routines:

1. Assemble the routines. (The routines can be placed in a private call library as described in *OS/VS Linkage Editor and Loader.*)
2. Link-edit the routines to SYS1.VTAMLIB (for OS/VS1 and OS/VS2 SVS) or SYS1.LINKLIB (for OS/VS2 MVS), replacing the IBM-supplied dummy module or modules. These changes must be made before ACF/VTAM is started.

The names of the replacement modules must be the same as the names of the IBM-supplied modules. For OS/VS1 and OS/VS2 SVS, they are the authorization exit routine, ISTAUAAT, and the accounting exit routine, ISTAUAAG. For OS/VS2 MVS, they are the authorization exit routine, ISTAUCAT, and the accounting exit routine, ISTAUCAG.

The IBM modules can be replaced by the linkage editor in either of two ways:

Use the NAME control statement with the replace function. An example is:

```
NAME ISTAUAAT(R).
```

Specify the disposition field of the SYSLMOD DD statement as OLD. Some examples are:

```
//SYSLMOD DD DSNAME=SYS1.VTAMLIB(ISTAUAAT),DISP=(OLD,KEEP), . . .
//SYSLMOD DD DSNAME=SYS1.LINKLIB(ISTAUCAT),DISP=(OLD,KEEP), . . .
```

Chapter 6. Reliability, Availability, And Serviceability Facilities

The reliability, availability, and serviceability (RAS) facilities are:

Configuration restart

Teleprocessing online test executive program (TOLTEP)

The traces for buffers, I/O, NCP lines, and internal events

The network control program (NCP) dump

Error recording and recovery features

Configuration Restart

Configuration restart is an ACF/VTAM facility that reactivates the network after it is deactivated or a failure occurs in it. Configuration restart can be immediate or delayed. Immediate configuration restart automatically reestablishes the status of:

A local or remote NCP when restart requires reloading the NCP after a failure occurs

A physical unit (and its associated logical units) when it loses contact with ACF/VTAM

Delayed configuration restart is begun by a network operator command after:

An ACF/VTAM failure

A host operating system or host computer failure

A communications controller or an NCP failure from which ACF/VTAM did not immediately recover

Deactivation of the network (or any part of it) by the network operator

Figure 6-1 illustrates how ACF/VTAM records changes to the network configuration.

When the host operating system is generated, ACF/VTAM definition statements are filed in SYS1.VTAMLST. The first time a major node is activated, activation information is recorded in the resource definition table (RDT) and in the NODELST data set, if one exists. Any further activations or deactivations of major nodes are also recorded in the RDT and in the NODELST data set (if any). Every time the network operator changes the status of a minor node, the change is recorded in the RDT and (optionally) in a configuration restart data set.

Information from configuration restart and NODELST data sets is used for a delayed configuration restart. Information from the RDT is used for an immediate configuration restart.

Immediate Configuration Restart

Status information for an immediate restart of an NCP, or a physical unit and its logical unit is available in the RDT.

Immediate Restart Of The NCP

When the NCP is first activated, ACF/VTAM builds the RDT from the NCP generation deck to represent the NCP's configuration. As the network operator changes the NCP's configuration, ACF/VTAM records the new status in the RDT.

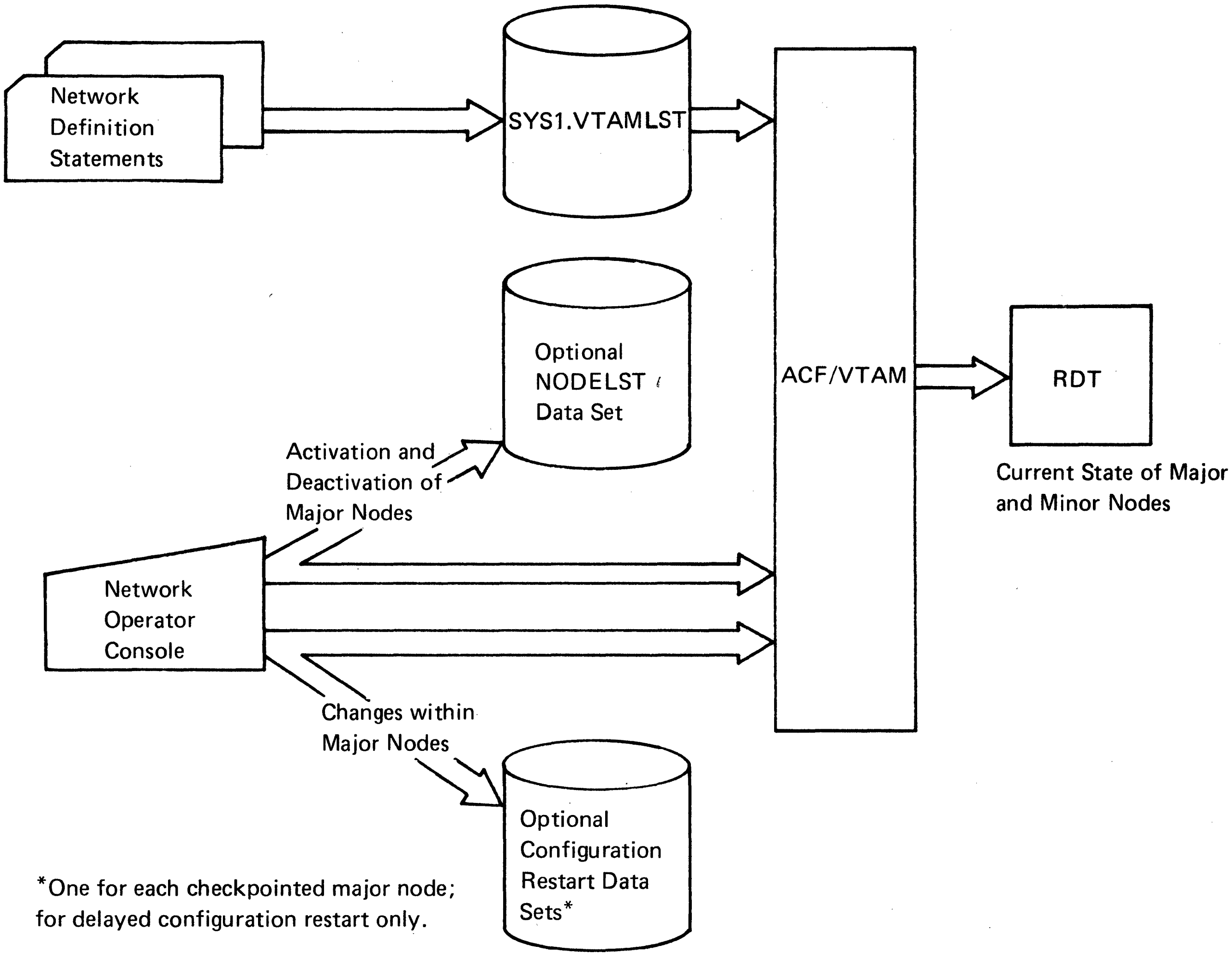

Network Definition Statements
SYS1.VTAMLST
ACF/VTAM
RDT
Current State of Major and Minor Nodes
Optional NODELST Data Set
Activation and Deactivation of Major Nodes
Network Operator Console
Changes within Major Nodes
Optional Configuration Restart Data Sets*
*One for each checkpointed major node; for delayed configuration restart only.

Figure 6-1. Recording Changes to the Network Configuration

For an NCP or communications controller failure:

If the error is not a channel (link) error and if AUTODMP=YES was specified on the PCCU definition statement, or if the network operator requests a dump, ACF/VTAM dumps the contents of the communications controller.

If AUTOIPL=YES was specified on the PCCU definition statement, or if the network operator requests a reloading of the NCP, ACF/VTAM loads a fresh copy of the NCP into the communications controller.

If the communications controller is successfully reloaded, ACF/VTAM:

Reactivates links that were active at the time of failure.

Restarts any remote BSC cluster controllers or physical units attached to the communications controller. A remote communications controller is restarted but is not reloaded unless it also failed.

Determines whether the network operator has used MODIFY commands to change any of the line-scheduling parameters from the system definition status. For polled and nonswitched non-SDLC links, ACF/VTAM resets values that existed when the NCP failed. The MODIFY command is explained in *ACF/VTAM Network Operating Procedures.*

Resets the date and time of day in the communications controller.

Allocates PEP lines to the partition with which they were associated at the time of the failure.

The I/O trace and buffer content trace are reestablished but the NCP line trace is not. The buffer pool utilization trace is not affected.

Uncompleted data-transfer requests for start-stop and BSC terminals are returned to the application programs with notification of the status of the communications controller. If the restart is successful, the data-transfer requests can be retried. See "Immediate Restart of a Physical Unit" below for a description of how ACF/VTAM notifies application programs in session with logical units about data-transfer requests.

After a communications controller failure, the network operator is notified of configuration restart processing. If the attempt to reload the communications controller fails, ACF/VTAM deactivates the failing part of the network.

Immediate Restart Of A Physical Unit

Although ACF/VTAM attempts to reactivate physical units and logical units that were active at the time of failure, it does not reload the physical unit. Application programs in session with logical units are notified through their LOSTERM exit routines when a loss of contact with the logical units occurs.

If the physical unit is successfully restarted, the application program is notified, and it must reestablish its session with the logical unit by issuing CLSDST and OPNDST macro instructions.

If a logical unit is restarted and if that logical unit's definition statement contains a LOGAPPL operand, ACF/VTAM automatically generates a logon for that logical unit to the application program named the LOGAPPL operand, whether or not the logical unit was connected to that application program when the failure occurred in the network. See "The LOCAL Definition Statement" in Chapter 2 for a description of the LOGAPPL operand.

Switched physical units and locally attached physical units are not reconnected until a request is issued for a session between logical units. If the restart is unsuccessful or is not attempted because of a channel or SDLC link failure, the physical unit is deactivated. Application programs can continue processing with the sessions that were not affected. The network operator is informed of the status of the physical unit.

Delayed Configuration Restart

Delayed configuration restart is begun by a network operator command after an ACF/VTAM failure, a host operating system or host computer failure, a communications controller or an NCP failure from which ACF/VTAM did not immediately recover, or deactivation of the network (or any part of it) by the network operator.

If, in response to an attempted restart after a system or ACF/VTAM failure (including operator cancellation of ACF/VTAM), ACF/VTAM issues the message IST180I (indicating that an OPEN failure occurred), the operator must use the VERIFY command (see *OS/VS Access Method Services User's Guide,* GC26-3836) to ensure that the named VSAM configuration restart or NODELST data set is properly closed. If the operator retries the restart without first using VERIFY, this message is not repeated, but the information in the data set might be damaged.

After ensuring that the data sets are properly closed, the operator can then use the VSAM data sets to restore the network configuration (as shown in Figure 6-2).

The status to which the network configuration is restored depends on whether NODELST and configuration restart data sets were defined and on how the network operator uses the CONFIG operand and the optional COLD or WARM operand when restarting the network.

Configuration Restart Data Sets and NODELST Data Sets

The user can define VSAM configuration restart data sets for the NCP major nodes, local SNA major nodes, switched SNA major nodes, and local non-SNA major nodes in the network. In addition, if the Multisystem Networking Facility is installed, configuration restart data sets can be defined for CDRM and CDRSC major nodes. ACF/VTAM uses these data sets to record changes to the minor nodes within these major nodes. If the VSAM data sets are defined, the network operator can specify how ACF/VTAM is to use the data sets for a delayed configuration restart.

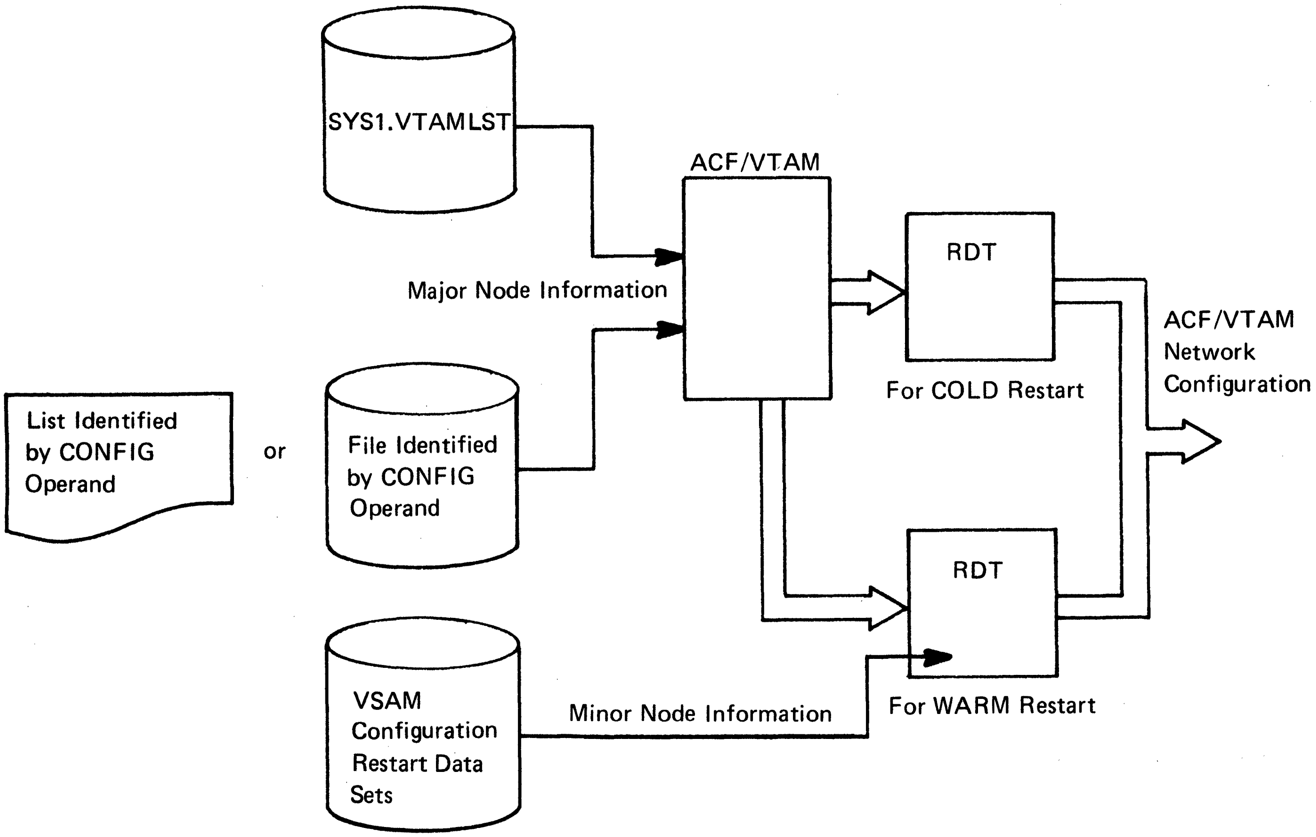


Figure 6-2. Delayed Configuration Restart

To associate VSAM configuration restart data sets with individual major nodes, the CONFGDS operand is coded on:

The PCCU definition statement for an NCP major node

The LBUILD definition statement for a local non-SNA major node (see Chapter 2)

The VBUILD definition statement for a local SNA or switched SNA major node (see Chapter 2)

The VBUILD definition statement for a CDRM or CDRSC major node (Only if the Multisystem Networking Facility is installed, see Chapter 2)

A NODELST data set can be defined to record which major nodes are active in the network, and if the Multisystem Networking Feature is installed, which path table definitions are active. A NODELST data set is specified with the NODELST start option. For more information on specifying a NODELST data set, see Chapter 3 in this book and *ACF/VTAM Network Operating Procedures.*

The network operator can use either or both of these data set types when performing a delayed configuration restart.

If the user defines configuration restart data sets, ACF/VTAM provides two additional forms of recovery: manual switching to a backup CPU and manual switching to a backup communications controller. (More information is provided later in this section.)

The CONFIG start option specifies the major nodes to be reactivated and the COLD or WARM start options specify how the minor nodes are to be restored. For a delayed configuration restart, ACF/VTAM builds a new RDT from data in SYS1.VTAMLST and in the list or VSAM data set specified in the CONFIG operand. If a NODELST data set was specified when the network was last started, then the major nodes that were active when the failure or deactivation occurred can be reactivated by specifying that same data set name with the CONFIG operand. The major nodes that were initially active (when ACF/VTAM was last started) can be reactivated by specifying (with the CONFIG operand) the list or VSAM data set that was specified in the CONFIG operand when the network was last started.

If the COLD operand is specified or assumed by default, ACF/VTAM restores the minor nodes of each major node specified by the CONFIG operand to its initial status. The records in any configuration restart data sets for these major nodes are deleted. If the WARM option is specified, ACF/VTAM restores each major node specified by the CONFIG operand to the status recorded for them in the configuration restart data set (provided a configuration restart data set is defined for it). The data in this VSAM data set is used to update the RDT entry for each minor mode within that major node.

The following information is recorded for an NCP major node:

The channel-unit address (if local) or the RNAME (if remote)

If the Multisystem Networking Facility is installed, for a local 3704 or 3705 controller, whether a remote connection to another 3704 or 3705 is,a cross-domain connection

If the Multisystem Networking Facility is installed, for a local 3704 or 3705 controller with a backup connection for a primary cross-domain connection, the RNAME of the primary connection

For each line, whether it is active or inactive, its line scheduling parameters, and for dial SDLC lines, the Answer Mode (whether the line is dial-in or not)

For each nonswitched non-SNA device, its device transmission limit, the name specified in the LOGAPPL operand, and whether it is active or inactive

For each UTERM entry, the name specified in the LOGAPPL operand, and whether it is active or inactive

For each BSC or start-stop port, whether it is active or inactive

For each BSC general-poll cluster, whether it is active or inactive

For each logical unit, the names specified in the LOGAPPL and LOGMODE operands, and whether it is active or inactive

For each physical unit, whether it is active or inactive

If the ACF/VTAM Encrypt/Decrypt Feature is installed, ACF/VTAM records the cryptographic status (REQD, SEL, OPT, or NONE) for each logical unit.

The following information is recorded for a local SNA major node:

For each channel-attached controller, its channel-unit address, and whether it is active or inactive

For each logical unit, the names specified in the LOGAPPL and LOGMODE operands, and whether it is active or inactive

For each physical unit, whether it is active or inactive

If the ACF/VTAM Encrypt/Decrypt Feature is installed, ACF/VTAM records the cryptographic status (REQD, SEL, OPT, or NONE) for each logical unit.

The following information is recorded for a switched SNA major node:

For each logical unit, the names specified in the LOGAPPL and LOGMODE operands, and whether it is active or inactive

For each physical unit, whether it is active or inactive

For a physical unit with dial-out capability, the PATH=USE | NOUSE setting

If the ACF/VTAM Encrypt/Decrypt Feature is installed, ACF/VTAM records the cryptographic status (REQD, SEL, OPT, or NONE) for each logical unit.

The following information is recorded for a local non-SNA major node:

For each minor node, its LOGAPPL operand value, and whether it is active or inactive

If the Multisystem Networking Facility is installed, the following information is recorded for a CDRSC major node:

For each minor node, whether it is active or inactive

If the Multisystem Networking Facility is installed, the following information is recorded for a CDRM major node:

For each minor node, whether it is active or inactive

If a major node's configuration restart data set is empty or if the major node has no configuration restart data set, the minor node status information for that major node is set to the initial status. (This applies only when WARM is specified in the start procedure. If WARM is specified in a VARY command to activate a major node that has no configuration restart data set or has an empty data set, ACF/VTAM rejects the VARY command.)

Restart To Initial Status

ACF/VTAM restores the configuration to its initial status as defined by the user if the network operator specifies the COLD operand or allows the COLD operand to be assumed when using the start procedure or when issuing a VARY ACT command for each major node.

For an NCP generated with the automatic network shutdown feature (ANS=YES in the BUILD macro instruction), the NCP is not reloaded if the communications controller already contains the correct NCP. Because the COLD start option was specified, the network operator must reissue the network operator commands to restore the configuration to the prior state, if desired.

Restart To Status Before Failure Or Deactivation

If the network operator specifies the WARM operand in the start procedure or in a VARY ACT command for a major node for which there is an associated configuration restart data set, ACF/VTAM updates its RDT from information in the configuration restart data sets. ACF/VTAM then restores the configuration to its status prior to the failure or deactivation.

If the network operator specifies the WARM operand when starting ACF/VTAM, any major node identified in the CONFIG operand that does not have an associated configuration restart data set or that has an empty configuration restart data set is reactivated to its initial status. An informational message is issued to the network operator for each major node.

The VARY ACT command fails if the network operator specifies the WARM operand for a major node that does not have an associated configuration restart data set or that has an empty configuration restart data set.

For an example of how NODELST and configuration restart data sets are used, see Figure 6-3.

Restart Considerations

Because the status of any ACF/VTAM commands being processed at the time of failure cannot be predicted, the network operator should use DISPLAY commands to determine whether any commands being processed at the time of failure need to be reissued. For a discussion of the network operator commands, refer to *ACF/VTAM Network Operating Procedures.* After a failure in ACF/VTAM, the host computer, or the host operating system, and the subsequent restart of the network, communication might not be possible with lines or terminals associated with the NCP that were being tested with online test programs (OLTs) when the failure occurred. After the status of the network is reestablished, TOLTEP should be run again for those lines or terminals.

If a logical unit is restarted and if that logical unit has a LOGAPPL value associated with it (either in it's definition statement or through the VARY LOGON command, ACF/VTAM automatically generates a logon for that logical unit to the application program named by the LOGAPPL operand, whether or not the logical unit was connected to that application program when the failure occurred in the network. See "The LOCAL Definition Statement" in Chapter 2 for a description of the LOGAPPL operand.

In this example, major nodes A, B, and C have configuration restart data sets (CONFGA for major node A, CONFGB for major node B, and so on). Each major node comprises 4 minor nodes (for example, major node A comprises the minor nodes A1, A2, A3, and A4). When the minor nodes are defined, the initial status (active or inactive, for example) can be specified. In this case, the minor nodes B3 and B4 are to be initially active and the other minor nodes inactive. This configuration can be represented as follows:

	Major Node A	Major Node B	Major Node C
Minor Nodes	A1 A2 A3 A4	B1 B2 B3* B4*	C1 C2 C3 C4

*Defined as initially active

Configuration Restart Data Sets

CONFGA	CONFGB	CONFGC
empty	empty	empty

When ACF/VTAM is started, the network operator can enter start options. In this example, the network operator uses the CONFIG operand to specify the list XX, which lists major nodes A and C as initially active. The operator uses the NODELST operand to specify that the VSAM data set ABC is to be used to record which major nodes are active in the network:

CONFIG=XX,NODELST=ABC

Because of these start options:

ACF/VTAM activates major nodes A and C.
ACF/VTAM records the activation of major nodes A and C in data set ABC.

These actions can be represented as:

List XX			VSAM Data Set ABC
A and C initially active	→ ACF/VTAM activates major nodes A and C	→ ACF/VTAM records the activation of A and C in data set ABC →	A and C currently active

Then, to activate major node B, the operator enters:

VARY ACT,ID=B

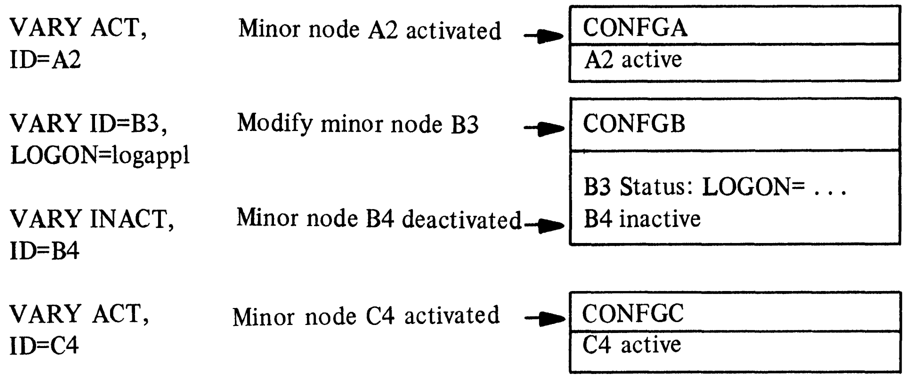
ACF/VTAM activates major node B and records its activation in ABC:

		VSAM Data Set ABC
VARY ACT,ID=B	B activated →	A, B, and C currently active.

ACF/VTAM activates minor nodes B3 and B4 because B3 and B4 are indicated as initially active in the definition statements for major node B. Because B3 and B4 are defined as initially active, ACF/VTAM does *not* record the activation of B3 and B4 in CONFGB.

Figure 6-3 (Part 1 of 2). Example of How NODELST and Configuration Restart Data Sets Are Used

After the major nodes are active, the operator activates, deactivates, or modifies the status of minor nodes with VARY commands. Every time ACF/VTAM activates, deactivates, or modifies the status of a minor node, it records that action in the configuration restart data set (if one exists) for the major node of which the minor node is a part.



VARY ACT,
ID=A2 — Minor node A2 activated → CONFGA: A2 active

VARY ID=B3,
LOGON=logappl — Modify minor node B3 → CONFGB: B3 Status: LOGON= . . .

VARY INACT,
ID=B4 — Minor node B4 deactivated → CONFGB: B4 inactive

VARY ACT,
ID=C4 — Minor node C4 activated → CONFGC: C4 active

The status information stored in the configuration restart data sets can be represented as:

Major Nodes

List XX	Data Set ABC
A and C initially active	A, B, and C currently active

Minor Nodes

Configuration Restart Data Sets		
CONFGA	CONFGB	CONFGC
A2 active A2 status	B3 status: LOGON= . . . B4 inactive	C4 active C4 status

If the network is now halted or if failure occurs in it, ACF/VTAM deactivates all the active major and minor nodes and closes the configuration restart data sets. The information needed to restore the network is in the ACF/VTAM definition statements and these data sets. When the network operator restarts the network, start options can again be entered.

If the operator enters the start options below	These major nodes are restarted	These minor nodes are restarted
CONFIG=XX,COLD	A and C	None
CONFIG=XX,WARM	A and C	A2 and C4
CONFIG=ABC,COLD	A, B, and C	B3 and B4
CONFIG=ABC,WARM	A, B, and C	A2, B3, and C4

Specifying CONFIG=XX,COLD restores the network to its initial status, while CONFIG=ABC,WARM restores the network to the status it had when it failed or was deactivated. Specifying only CONFIG=XX is the same as specifying CONFIG=XX,COLD because if neither COLD nor WARM is coded, COLD is assumed.

The operator can also specify another NODELST data set:

CONFIG=ABC,WARM,NODELST=QRS

Or, the operator can continue using the one used before:

CONFIG=ABC,WARM,NODELST=ABC

Figure 6-3 (Part 2 of 2). Example of How NODELST and Configuration Restart Data Sets Are Used

Switching To A Backup Computer

If configuration restart data sets are defined, ACF/VTAM provides some support for switching one or more major nodes to a backup computer. This kind of backup involves moving the VSAM configuration restart data set for each major node to be switched. The primary and backup computers must meet these conditions:

The same levels of ACF/VTAM and VSAM must be installed on both systems, although they need not have the same operating system.

The ACF/VTAM major node definitions and their corresponding hardware configurations must be the same in both systems. (Physically switching the hardware connections from one system to another would satisfy the requirement for matching hardware configurations.)

The MAXSUBA value must be the same in both ACF/VTAM systems.

The actual moving of the VSAM data sets is done by creating the data sets and a user catalog on the primary system and then using the VSAM Access Method Services to 'IMPORT CONNECT' the user catalog (and all of its cataloged data sets) in the backup system. (For details of VSAM catalog and data set portability, see *OS/VS Access Method Services User's Guide.*) The backup system can then gain access to the primary system's configuration restart data sets through either a shared DASD arrangement or by physically moving the VSAM volume containing the user catalog and data sets.

If ACF/VTAM is restarted in the backup system after moving the VSAM volume, the appropriate job control language for the VSAM user catalog and data sets can be included in the ACF/VTAM start procedure at that time. If, however, one or more major nodes are to be switched without restarting ACF/VTAM in the backup system, the job control language must be included when ACF/VTAM *is* started, sometime before the switch is made.

Switching To A Backup Communications Controller

ACF/VTAM uses configuration restart data sets to reconstruct the NCP's configuration when the network operator reactivates an NCP in a communications controller. This occurs when:

The channel to a local communications controller fails and the network operator switches to an alternate channel. If an NCP is equipped with a type 3 channel adapter, the switch to an alternate channel is automatically completed.

A failure occurs in a local or remote communications controller and the network operator switches the outboard lines of the communications controller to an alternate communications controller

A failure occurs in local communications controller and the network operator switches the link connecting the local communications controller with a remote communications controller to a backup local communications controller.

The network operator then issues the VARY ACT command, specifying the WARM operand and either:

A new channel unit address for the U operand if the communications controller is local

An alternate name for the RNAME operand if the communications controller is remote

ACF/VTAM restores the network configuration to its prefailure status. If the network operator reactivates a remote NCP after switching a local NCP to an alternate communications controller, ACF/VTAM ensures that the status of the remote NCP network configuration is synchronized with the local NCP.

Backup and Reconfiguration for a Multiprocessor

In a suitably prepared multiprocessor one processor can backup the other. The procedure required to do the necessary reconfiguration varies, depending on whether the attached devices are symmetric (devices that are attached to both processing units in a multiprocessor) or asymmetric (devices that are attached to only one processing unit in a multiprocessor). The following guidelines might be useful when developing operator procedures for reconfiguring a specific multiprocessor:

- In most cases, recovery for symmetric devices is provided automatically. For a symmetric 3705 Communications Controller equipped with a type 3 channel adapter, use the system VARY command to bring the channel paths offline from the alternate processing unit that has been reconfigured. No other action is required. Note that for a recovery sequence, the paths from an alternate processor will already be offline.
- Asymmetric devices must be equipped with a two channel switch to allow backup by an alternate processing unit. For a communications controller of local device:

  If a communications controller has not been deactivated by the ACF/VTAM ERPs, use the ACF/VTAM VARY INACT, I or VARY INACT, F command to deactivate the NCP.

  If a device has not already been deactivated by the ACF/VTAM ERPs, use the ACF/VTAM VARY INACT, I command to deactivate the device, or to deactivate the major node for that device.

  Switch the device or communications controller to the alternate processing unit.

  For each device affected by the physical switching operation, use the system VARY command to bring the channel paths for each device online, and, if necessary, to bring previous channel paths offline.

  Use the ACF/VTAM VARY ACT command to reactivate each minor node or the entire major node. Configuration restart data sets can be used as described elsewhere in this chapter.

  Restart user sessions as appropriate.

Switched Network Backup

ACF/VTAM allows you to define a physical unit and its associated logical units in such a way that you can use a switched line to back up a nonswitched line. To do this, you must define 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 units and its logical units are active as part of an NCP major node and the nonswitched line fails, the network operator can release the physical unit and its logical units with the VARY REL command and then activate the backup switched line, the physical unit, and its logical units, as part of the switched SNA major node.

See Chapter 2 for information on how to define NCP major nodes and switched SNA major nodes.

The Teleprocessing Online Test Executive Program (TOLTEP)

TOLTEP is the interface between ACF/VTAM and the online test programs that exercise and evaluate the hardware capabilities of some of the devices supported by ACF/VTAM. TOLTEP controls the selection, configuration, and use of the online test programs.

Before ACF/VTAM is installed, IBM Field Engineering uses configuration information and consults with users to make preparations for running TOLTEP. TOLTEP is fully described in *DOS/VS and OS/VS TOLTEP for ACF/VTAM.*

ACF/VTAM Traces

ACF/VTAM provides a buffer content trace, a buffer use trace, an I/O trace, an NCP line trace, and an internal trace. These traces are controlled by the TRACE start option when ACF/VTAM is started and by the MODIFY command while ACF/VTAM is running. A complete description of trace operands is given in *ACF/VTAM Network Operating Procedures.*

The buffer content, buffer use, I/O, and NCP line traces are debugging aids that enable the user to monitor the data flow in the network and aid the user in problem determination. The ACF/VTAM internal trace is a debugging aid that enables the system programmer or an IBM program systems representative to locate the cause of an ACF/VTAM failure. The ACF/VTAM internal trace is described in the *ACF/VTAM Debugging Guide* and is not discussed further in this book.

The ACF/VTAM traces are available through use of the generalized trace facility (GTF), an OS/VS service aid. A full range of system events, other than the ACF/VTAM traces, can be traced by use of GTF. GTF is fully described in the appropriate *Service Aids* manual.

GTF output, including ACF/VTAM trace data, is directed to a trace data set, SYS1.TRACE. Trace output is formatted and printed using the PRDMP service and with the Edit control statement. Edit also is described in the appropriate *Service Aids* manual.

The Buffer Content Trace

The buffer content trace records buffer contents (up to 224 bytes for an input message for an output message) for any of these types of nodes: application program, logical unit, component, terminal, SDLC or local cluster controller, and NCP. The buffer trace records the origin and destination node names, the direction of I/O flow, and message data.

Data flow to and from the SSCP can be traced by specifying ID=VTAM as a trace parameter when starting the trace.

There are two types of buffer traces, user traces and ACF/VTAM traces. The user trace consists of the data recorded at the application program buffer at the time of a request or completion of a request. The ACF/VTAM traces consists of the contents of the user trace, plus the TH and RH. ACF/VTAM traces are recorded at the time of transmission by ACF/VTAM.

The format of a trace record is described in *ACF/VTAM Debugging Guide.*

The Buffer Use Trace

The buffer use trace aids in determining a reasonable size for each of the buffer pools. Chapter 7 describes a method of using the buffer use trace to adjust the pool values to accurately represent the user's requirements.

The buffer use trace records the usage of all ACF/VTAM pools. By keeping the buffer use trace active throughout the ACF/VTAM session, you can collect trace records that show the time periods when requests for buffers are low, moderate, and high (as determined by the user).

When both GTF (USR trace option) and the ACF/VTAM buffer use trace are active, the trace information is collected periodically, depending on an IBM-supplied or user-supplied value.

The IBM-supplied value produces one record for approximately every 250 transactions. A transaction is defined here as one inbound message and one outbound message. If the value is reduced to 1, some trace output is certain. Once a high transaction rate is reached, the count should be increased to reduce system overhead. For information on how to change this value, refer to *ACF/VTAM Debugging Guide.*

The ACF/VTAM buffer use trace collects the following information for each of the ACF/VTAM buffer pools:

Pool name

The current number of buffers in use

The greatest number of buffers used since the last trace record for the pool

The number of requests queued since the last trace record for the pool, plus the number of requests still unsatisfied from the previous record.

The number of times the pool has been expanded

The greatest number of buffers contained in the pool (including expansions) since the last trace record for the pool

The current number of buffers in the pool

Date and timestamp (Greenwich Mean Time) if TIME=YES was specified in the GTF START command

For details on starting and stopping the ACF/VTAM buffer use trace, see *ACF/VTAM Network Operating Procedures.* For an explanation of the format of the buffer use trace output, see *ACF/VTAM Debugging Guide.*

The I/O Trace

The I/O trace records I/O activity for any of these types of nodes: component, terminal, SDLC cluster controller, and NCP. This trace records: the node identification, the direction of the data flow, the transmission header (TH), the request/response header (RH), the basic data unit (BDU) (if the device is BSC or start-stop), and sense information (if the device is SDLC).

The format of a trace record is described in *ACF/VTAM Debugging Guide.*

The NCP Line Trace

The line trace uses the NCP line trace facility to record operating parameters of a line each time a level-2 interruption occurs for that line. The NCP line trace only traces lines operating in network control mode. Level-2 is the program interruption level at which bit service or character service for the communication line is performed.

Up to eight lines at a time can be traced for an NCP. If more than eight line traces are requested, the additional requests are rejected until one of the previous traces has ended.

The trace facility can hurt the performance of the network. Therefore, the facility should be used with discretion.

Host System Preparations

GTF is included automatically in the operating system during system generation. The IBM-supplied cataloged procedure for GTF defines the trace output data set, SYS1.TRACE, and directs the output to an access device.

GTF trace options can be stored in SYS1.PARMLIB prior to starting GTF, or the options can be entered when GTF is started. The ACF/VTAM trace facility requires the following GTF options:

RNIO, so that the I/O trace can function for an NCP or a remote device attached to the NCP

IO or IOP, so that the I/O trace can function for a local device

USR, so that the buffer content, NCP line, and storage pool traces can function

If trace options are to be stored as members in SYS1.PARMLIB, a SYSLIB DD statement must be added to the IBM-supplied cataloged procedure. It is also desirable to change the EXEC statement in the IBM-supplied cataloged procedure to specify TIME=YES (see "Starting GTF").

Starting GTF

GTF must be started with the GTF START command before a trace can be activated from ACF/VTAM. The command, including information about overriding the parameters in the IBM-supplied cataloged procedure, is described in the appropriate *Service Aids* manual.

The network operator must specify TIME=YES when entering the GTF START command to timestamp each logical record as it is recorded. (As an alternative, the EXEC statement in the IBM-supplied GTF cataloged procedure can be changed to specify TIME=YES, and the procedure replaced on SYS1.PROCLIB.)

If GTF options were not previously stored in SYS1.PARMLIB, the network operator is prompted to enter the options when GTF is started.

The IBM-supplied cataloged procedure for GTF specifies MODE=EXT, meaning that the trace output is to be directed to a direct access device. For the I/O trace (RNIO trace option only), GTF trace output can be maintained in main storage, but less data is gathered for option RNIO. To do this, enter MODE=INT on the GTF START command.

Using Start And Modify Commands

Using ACF/VTAM start options entered at the network operator's console, or using a start option list, a network operator can:

Turn on the ACF/VTAM trace facility when ACF/VTAM is started, specify the node to be traced, and specify the type of trace to be performed

Use the NOTRACE operand to override the TRACE operand in the event TRACE was filed in a start option list

Using operands of the MODIFY command, the network operator can dynamically turn any of the four types of trace on or off for any traceable node.

For information about the use of the START and MODIFY commands, see *ACF/VTAM Network Operating Procedures.* The other trace options are described in Chapter 3.

A trace is turned off by using the NOTRACE operand in the START or MODIFY command. Additionally, the NCP line trace is terminated by ACF/VTAM for either of these reasons:

NCP entering slowdown mode

Loss of contact with the communications controller even if ACF/VTAM's configuration restart is successful

The network operator is notified if the NCP line trace is terminated. The network operator must then turn off the line trace with the MODIFY command, or all subsequent requests to trace that line will be rejected. If additional line tracing is needed on the same line, the trace can be started again by using the MODIFY command.

Printed Output From Traces

The PRDMP service aid is used to format and print GTF trace data recorded under MODE=EXT, including the buffer content, I/O, NCP line, and buffer use traces. The trace data can be edited by specifying special keywords in the EDIT control statement. The following keywords are used to format the trace data:

IO formats all I/O trace records of local devices.

IO = (cuu1,cuu2, . . . cuu50) formats I/O trace records of local devices with unit addresses cuu1, cuu2, etc., to a maximum of 50 devices.

RNIO formats the trace records of a local or remote NCP or devices attached to a local or remote communications controller.

USR=(CL01,CL02,LINE,TPIO) or USR=ALL formats buffer trace records from ACF/VTAM's buffers (CL01), from the ACF/VTAM buffer pool utilization records (CL02), from ACF/VTAM's I/O component (TPIO), from the NCP line trace (LINE), or all buffer, line trace, and buffer pool records (ALL). For example: USR=(CL01) or USR=(TPIO,LINE). Refer to the "Buffer Content Trace" section in this chapter for a full explanation of the CL01 and TPIO subparameters of the USR key word. If more than one subparameter is coded, separate the subparameters with commas.

GTF trace data recorded internally (MODE=INT) is dumped by the ABEND/SNAP dump. The formats of the I/O, buffer content, NCP line, and buffer use trace records are shown in the *ACF/VTAM Debugging Guide.*

OS/VS Error Recording

ACF/VTAM uses the system error recording facilities described in the *SYS1.LOGREC Error Recording* publication for your system. ACF/VTAM error recovery procedures interface with the system error recording facilities to format and write the error records to the SYS1.LOGREC data set. This data set is a system data set that resides on the system residence volume. The following records are written to the SYS1.LOGREC data set:

Short outboard recording (OBR) records for local 3270 terminals and the local communications controller in event of counter overflow or ACF/VTAM closedown

Long OBR records for local 3270 terminals and the local communications controller in the event of permanent errors

Software records when an abend occurs while ACF/VTAM is running

Miscellaneous data recorder (MDR) records from the NCP for line errors, local and remote communications controller errors, and errors on remotely attached devices

ACF/VTAM supplies information for the OBR and software records, and the NCP supplies communications controller or line information for MDR records.

OBR, software, and MDR records on the SYS1.LOGREC data set are edited and printed by using the IFCEREP0 service aid as described in the *SYS1.LOGREC Error Recording* publication for your system. The format of each printed record, except for information dependent on the communications controller, appears in the SYS1.LOGREC publication. For software records refer to the appropriate *System Data Areas* manual and for MDR records refer to *IBM 3704 and 3705 Program Reference Handbook,* for information in the record format dependent on the communications controller.

OBR Records

Short OBR records are written to SYS1.LOGREC whenever:

The Start I/O counter, temporary error counter, or a counter in the statistics table reaches its maximum setting (before counter overflow occurs).

The end-of-day (EOD) condition occurs. EOD occurs when the network operator takes the local device offline with the VARY command or the HALT command.

Long OBR records are written when a permanent error occurs.

Short and long OBR records contain this information:

Dates and time at which the counters overflowed (short OBR) or at which the permanent error occurred (long OBR)

Program identification

Unit address

Device types

Channel status word (CSW) in the long OBR record

Device-dependent information appended to the record

The device-dependent information includes:

Reason for the record being written (permanent error, counter overflow, or EOD)

Contents of the Start I/O (SIO) counter (that is, the number of SIO commands issued to the device)

Contents of the temporary error counter

CCW command code of the first failure of the local communications controller or 3270

CCW command code of the last retry of the local local communications controller or 3270

Node name of the device

Contents of the statistical data recorder

Sense bytes (long OBR record only) for the first and last command failure

The SIO counter and temporary error counter are maintained by ACF/VTAM for each locally attached device.

Software Records

ACF/VTAM creates a recovery environment through the STAE macro instruction. If an abend occurs while ACF/VTAM is running, an ACF/VTAM STAE exit routine gains control. One function of the routine is to write a software record to SYS1.LOGREC if ACF/VTAM was in control when the abend occurred.

The software record contains four areas:

The software record header as described in the SYS1.LOGREC publication.

SDWA, the STAE work area maintained by the OS/VS supervisor and described in the appropriate *System Data Areas* manual. SDWA includes the name of the abending program, or the request block, (RB) of the abending program if in supervisor mode, contents of the general-purpose registers, name of the module and control section in which the error occurred, and the PSW.

An area of zeros (for OS/VS1 and OS/VS2 SVS only, to provide compatibility with the OS/VS2 MVS software record).

A 100-byte area containing information from ACF/VTAM's component recovery area (CRA). Figure 6-4 is a general description of the CRA fields recorded in the software record.

MDR Records

The error recovery procedures in the communications controller build MDR records that show:

Permanent line errors

Statistical data on line errors

Errors on devices attached to a communications controller

Communications controller errors:

Channel adapter

Communications scanner

Program checks

Miscellaneous level-1 and level-3 interruption requests

The MDR record for a remote communications controller includes the information that appears in an OBR record for a local communications controller. The device-dependent portions of these records are documented in *IBM 3704 and 3705 Program Reference Handbook.*

Offset		Size		
Decimal	Hex	(Bytes)	Field Name	Description*
0	(0)	4	CRACRR	Address of the CRR being used when the abend occurred.
4	(4)	36	CRALKACT	ACF/VTAM lock information: the level of the locks held and the addresses of the lockwords.
40	(28)	20	CRAPSS	*First 16 bytes:* prefix (control block ISTPFCRR) including CRR identification and pointer to old CRR. *Last 4 bytes:* for the PSS component, trace flags for the current module and modules entered.
60	(3C)	20	CRAPROCR	*First 16 bytes:* prefix (control block ISTPFCRR) including CRR identification and pointer to old CRR. *Last 4 bytes:* for the control layer or TPIOS component, trace flags for the current module and modules entered.
80	(50)	20	CRASVC	*First 16 bytes:* prefix (control block ISTPFCRR) including CRR identification and pointer to old CRR. *Last 4 bytes:* for the CID handling component, trace flags for the current module and modules entered.

*ACF/VTAM control blocks are described in *ACF/VTAM Data Areas.*

Figure 6-4. Format of ACF/VTAM CRA Information in the SYS1.LOGREC Software Record

Processing The SYS1.LOGREC Data Set

Records on the SYS1.LOGREC data set can be selected, formatted, and printed by using the environment recording, editing, and printing service aid, IFCEREP0. IFCEREP0 can perform these functions:

Edit or summarize records on the SYS1.LOGREC data set and write the edited or summarized records to a user-specified output device.

Collect records from the SYS1.LOGREC data set and write the collected records onto an accumulation data set to provide comprehensive error statistics. The records on the accumulation data set can be further edited and written by IFCEREP0 to a user-specified output device.

Collect records from the SYS1.LOGREC data set and write the collected records onto a measurement data set. The measurement data set is used as input for the reliability data extractor (RDE) summary function of IFCEREP0.

Use the RDE summary function of IFCEREP0 to process the records on the measurement data set to produce an IPL report and a system error report. The IPL report contains comprehensive information on the reasons for system initializations, system idle time, and system unavailability time. The system error report contains error statistics for device and program failures.

Process the records on the SYS1.LOGREC or accumulation data set to produce a report that gives an overall description of the records in the input data set.

Chapter 7. Tuning ACF/VTAM

You can tune ACF/VTAM to meet your requirements using information described in this chapter:

ACF/VTAM Buffer Pools: Explains the structure and content of the various pools and how you can use the default buffer pool values or calculate your own values, tune these values, and put pools in fixed storage.

ACF/VTAM Tuning Statistics: Describes the ACF/VTAM tuning statistics facility, lists some of the variables that can be adjusted to tune ACF/VTAM, and briefly explains how the tuning statistics can be used to decide how to set those variables.

When tuning ACF/VTAM, you must also consider its effect on your requirements for all of your operating system and the hardware configuration it supports.

ACF/VTAM Buffer Pools

ACF/VTAM has 11 buffer pools to control the buffering of data. ACF/VTAM dynamically allocates and deallocates space in these buffer pools for the ACF/VTAM control blocks, I/O buffers, and channel programs that control the transmitting of this data. For information on how these pools are used, refer to the *ACF/VTAM Debugging Guide.* For information on ACF/VTAM control blocks and which pools they are obtained from, see Appendix E.

Types Of Buffer Pool Allocation

ACF/VTAM provides two types of buffer pool storage allocations. One type, basic allocation, is made for each buffer pool when ACF/VTAM is started. The other type, dynamic allocation, is a process by which ACF/VTAM temporarily increases the size of a buffer pool when there are heavy demands for space in that pool. Dynamic allocation, which takes place only if the user asks for it, allows the system programmer to reduce the amount of storage that must be permanently allocated for ACF/VTAM buffer pools. It also enables the system programmer to provide for temporary peak demands or for unexpectedly high demands for buffers, a feature that is useful when initializing a system.

Basic Allocation

The basic allocation is the amount of space reserved for the buffer pool when ACF/VTAM is started.

This basic allocation is specified by the *baseno, bufsize, slowpt,* and F start-option parameters (see Chapter 3). The meanings of these parameters are:

baseno
indicates the initial number of buffers to be provided in the buffer pool. After ACF/VTAM is started, the pool always contains at least this number of buffers.

bufsize
indicates the size in bytes of each buffer in the buffer pool.

slowpt

indicates the point at which the buffer pool is to enter slowdown processing. The pool enters slowdown processing whenever the number of buffers currently not in use in the pool is less than or equal to *slowpt.* Do not confuse *slowpt* with a VTAM Level 2 start parameter, *bth,* which referred to the number of buffers in use. Any buffer pool size calculations must be recalculated, with *slowpt* replacing *bth.*

F

indicates that a buffer pool that is normally in pageable storage is to be put in fixed storage.

These parameters establish the base size and the characteristics of the buffer pool. Figure 7-1 shows the structure of an ACF/VTAM buffer pool after basic allocation.

The *baseno* parameter should be specified so that it allocates a buffer pool large enough to meet the maximum demand for buffers (or, if dynamic allocation is used, the normal demand for buffers).

Values for the basic allocation for each buffer pool must be available to ACF/VTAM when it is started. If the user fails to specify a start-option parameter for a buffer pool, ACF/VTAM uses an IBM-supplied value for the missing parameter. The IBM-supplied values are not necessarily the most efficient values for your system, nor are they necessarily compatible with any dynamic allocation specifications you might take. For more information on IBM-supplied values for start-option parameters, see "Using IBM-Supplied Values For Basic Allocation" in this chapter.

Effect Of The Slowdown Point

When the number of buffers remaining available in a pool is equal to or less than the slowdown point (*slowpt*), the pool enters slowdown processing. During slowdown processing, buffers are allocated only for priority requests. (Priority requests are those requests for storage that must be satisfied to prevent system interlocking.) Nonpriority requests are not honored if doing so would cause the pool to enter slowdown processing. Nonpriority requests are queued or are rejected with a return code. Slowdown processing ends as soon as the number of available buffers becomes equal to or greater than *slowpt* and there are no queued requests for storage.

Dynamic Allocation

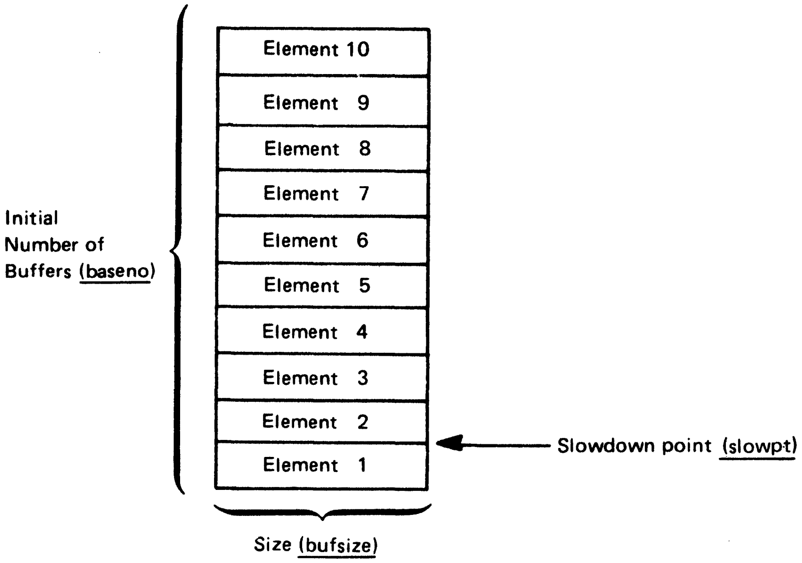
Two additional buffer pool parameters are provided to enable the user to specify that during heavy demands on the buffer pool, the buffer pool is to be temporarily enlarged. The parameters and their meanings are:

xpanno

indicates the number of buffers to be added to the buffer pool whenever dynamic allocation is needed. Whenever the buffer pool is to be expanded, ACF/VTAM acquires the smallest number of whole pages of storage that are sufficient to provide the number of buffers specified in xpanno. (For example, if 5 buffers will fit on one page of storage, and if *xpanno* is specified as 6, ACF/VTAM acquires two pages of storage whenever the buffer pool must be expanded, and expands the pool by 10 buffers.

xpanpt

is a decimal integer that specifies the expansion point for this buffer pool. When the number of buffers not in use in the buffer pool falls to a value that is equal to or less than *xpanpt,* ACF/VTAM schedules an asynchronous routine to expand the buffer pool by the number of buffers specified by *xpanno.* The value of *xpanpt* must be



This example shows a pool for which the start options were specified as poolname=(10,bufsize,1). The initial number of buffers in the pool is 10, the slowdown point is set at 1, and the bufsize is the length in bytes of each buffer in the pool.

Figure 7-1. A Buffer Pool with No Expansion Parameter Specified

greater than the value of *slowpt,* but less than the value of *baseno* minus *adjval,* where *adjval* is an adjustment value for this buffer pool. (See Figure 7-4 for the values of the adjustment value for each pool.) If you specify an *xpanpt* value, but omit the *slowpt* value, make sure that the *xpanpt* value is greater than the default *slowpt* value for the pool. If *xpanpt* is not specified, no buffer pool expansion occurs.

Dynamic expansion takes place only when the user specifies both the *xpanno* and *xpanpt* parameters for the pool. If *xpanno* and *xpanpt* are not both specified, the pool always remains the size specified by the *baseno* and *bufsize* parameters.

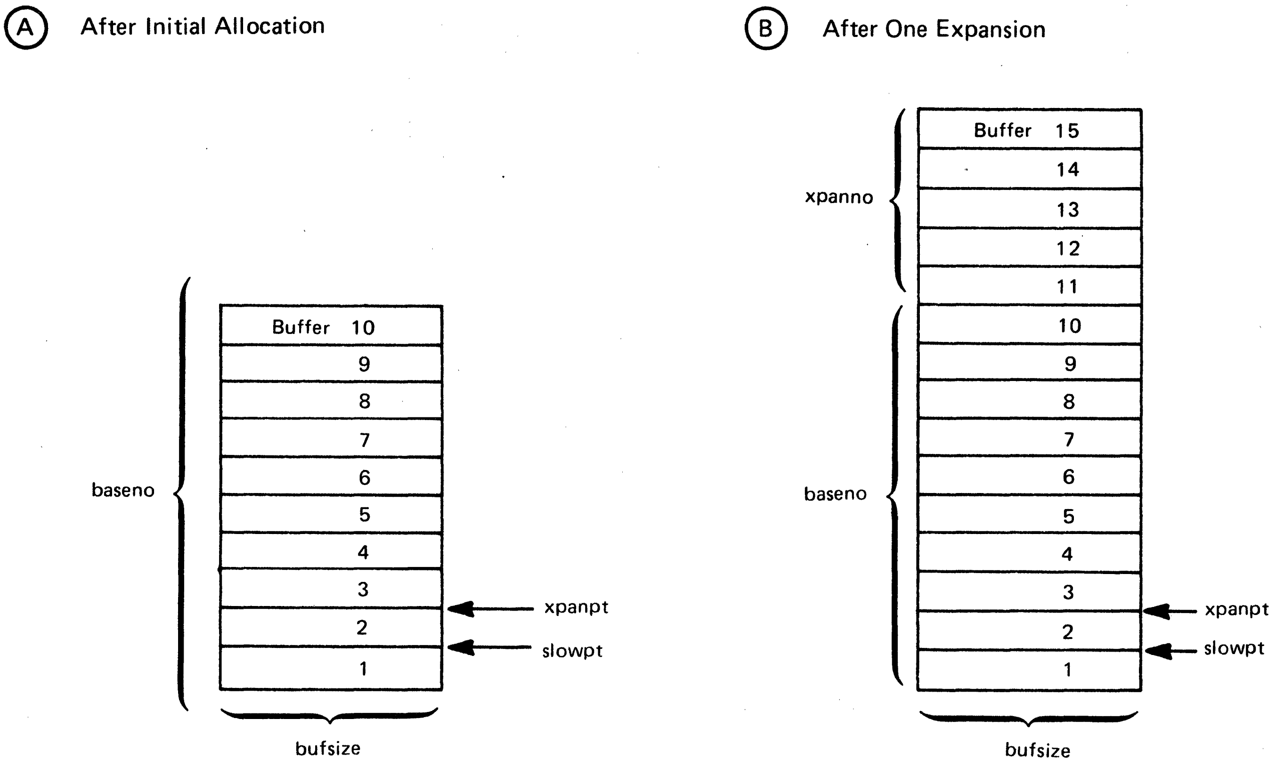
The buffers acquired by dynamic expansion are functionally the same as the buffers provided by the base allocation.

Figure 7-2 shows the structure of a pool (A) after basic allocation, and (B) after one dynamic expansion of the pool.

Purpose Of Dynamic Expansion

Without dynamic expansion of a pool, you would have to specify basic allocation parameters large enough to meet the greatest possible demands on the pool. With dynamic expansion, smaller basic allocation values can be specified and peak demands on the pool can be met with dynamic expansion.

Dynamic expansion is not intended to be used frequently; it is intended only to meet peak demands on the pool. For example, if a user experiences peak demands at certain times of the day, dynamic expansion could be used to meet those periods of peak demand. The basic allocation parameters would be specified to provide enough buffers for the periods of normal activity.



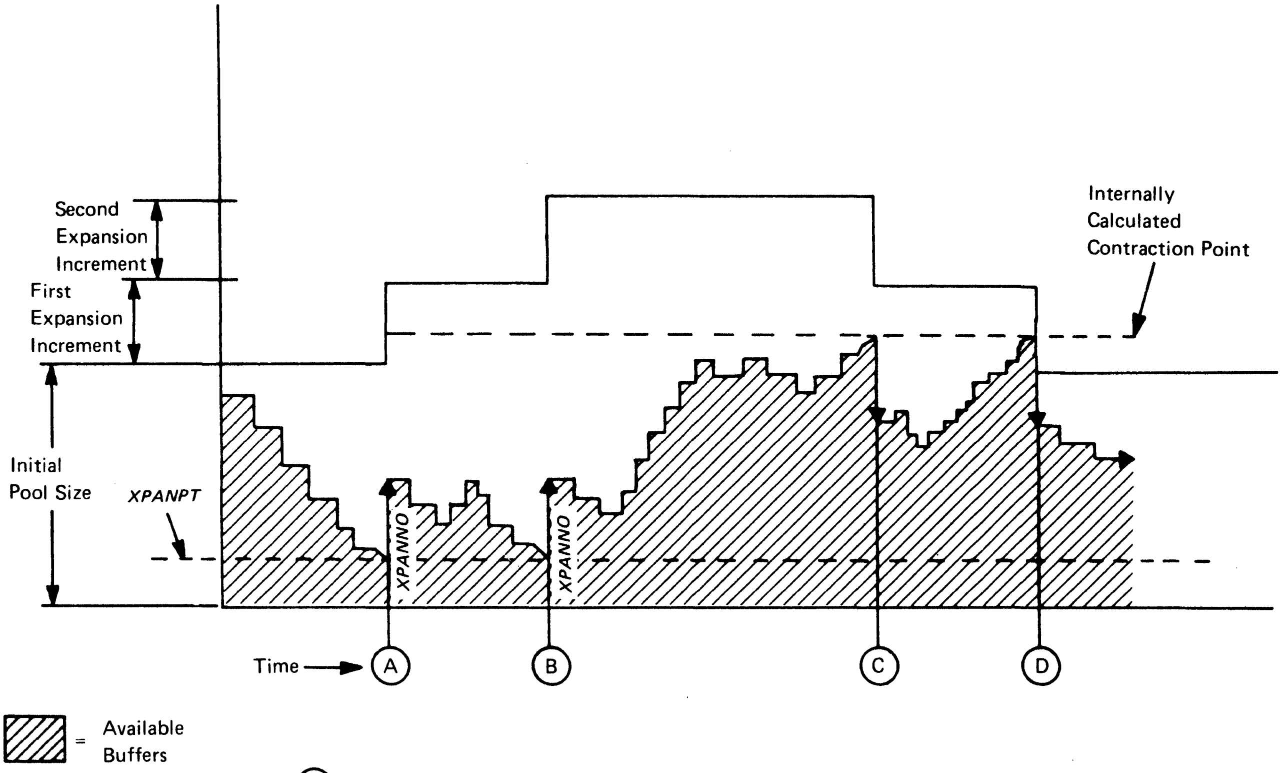
A This example shows a buffer pool for which the start options were specified as *poolname=((10,bufsize,,1,5,2).* After initial allocation, the pool contains 10 buffers (baseno=10), the length in bytes of each buffer is *bufsize,* the slowdown point is 1, the expansion size is 5 buffers (assume that 5 buffers fill one page of storage), and the expansion point is 1.

B After one expansion, there are 15 buffers in the pool. Each of the 5 additional buffers has a length of *bufsize* and the same expansion point and slowdown point as before.

Figure 7-2. A Buffer Pool after Initial Allocation and after One Expansion

The user should consider carefully whether dynamic expansion is appropriate for the type of demands the system makes on each pool. A large basic allocation for the pool means that pool processing is more efficient, but more storage is tied up for that pool. Dynamic expansion provides more efficient use of storage, but reduces processing efficiency.

The pool is expanded when the number of buffers not currently in use in the pool is less than or equal to *xpanpt*. The pool goes into slowdown mode when the number of buffers not currently in use in the pool is less than or equal to *slowpt*. When necessary, a pool is expanded repeatedly, with the pool growing larger and larger. The limitation on the size of each pool is 32767 times bufsize. When the number of buffers not currently in use is equal to or greater than (2 X *xpanno*) + *xpanpt*, ACF/VTAM checks to see if the buffers it acquired in the previous expansions of the pool are not in use, and if so, ACF/VTAM releases those buffers. Buffers are released in blocks, just as they are acquired. If any of the buffers in a block are in use, ACF/VTAM does not release any of the buffers. This process of expansion and contraction is shown in Figure 7-3.


Second Expansion Increment
First Expansion Increment
Initial Pool Size
XPANPT
XPANNO
XPANNO
Internally Calculated Contraction Point
Time
A
B
C
D
▨ = Available Buffers

(A) When the number of available buffers becomes equal to or less than xpanpt, the pool is enlarged by the number of buffers specified in xpanno (rounded up to the nearest page size).

(B) When the number of elements again slips below xpanpt, the pool is again enlarged.

(C) When the number of available buffers becomes equal to or greater than an internally calculated contraction point, and when all of the buffers in an expansion increment become available, the buffers in that expansion increment are released and the size of the pool is reduced.

(D) The pool is again contracted when the number of available buffers exceeds the contraction point and all of the buffers in the expansion increment become available. This contraction returns the pool to its initial size.

Figure 7-3. Expansion and Contraction for a Buffer Pool

Using IBM-Supplied Values For Basic Allocation

When the user does not specify a base allocation parameter for a pool, (*baseno, bufsize,* or *slowpt*), an IBM-supplied value is used to construct the pool. The IBM-supplied values for the pools should be large enough to meet the network configuration requirements and workload of many users. These values should help eliminate problems caused by inadequate pool sizes. However, if the IBM-supplied values are inadequate or inappropriate, you can calculate your own values.

The IBM-supplied pool values are shown in Figure 7-4.

For OS/VS1 and OS/VS2 SVS

	IBM-Supplied Values			
Buffer Pool Name	*baseno*	*bufsize*	*slowpt*	*adjval*
APBUF	25	60	3	0
CRPLBUF	20	120	2	0
IOBUF	5	64*	1	0
LFBUF	10	120	0	1
LPBUF	15	1016	2	5
NPBUF	15	200	1	0
PPBUF	5	64*	1	0
SFBUF	10	72	0	1
SPBUF	5	104	0	0
UECBUF	10	112	1	0
WPBUF	15	168	0	0

For OS/VS2 MVS

	IBM-Supplied Values			
Buffer Pool Name	*baseno*	*bufsize*	*slowpt*	*adjval*
APBUF	129	60	13	0
CRPLBUF	208	120	15	0
IOBUF	100	64*	19	0
LFBUF	104	120	0	0
LPBUF	64	1016	0	4
NPBUF	192	200	16	0
PPBUF	175	64*	18	0
SFBUF	163	72	0	5
SPBUF	3	104	0	0
UECBUF	34	112	4	0
WPBUF	78	168	0	0

*This value must be overriden to equal the value specified for the UNITSZ operand in the NCP HOST macro instruction.

Figure 7-4. IBM-Supplied Buffer Pool Values

Calculating Values For Basic Allocation

You can calculate your own basic allocation values if the IBM-supplied values are either inadequate or inappropriate. Appendix C shows how to calculate the base-buffer-count (*baseno*) value for a specific pool. The IBM-supplied *bufsize* and *slowpt* values should be used for all pools except for IOBUF and PPBUF. In general, the values for basic allocation for a pool should be somewhat above the amounts needed for the greatest possible usage of the pool. However, if dynamic expansion is specified, the values should be somewhat above the amounts needed for the normal (fairly constant) usage of the pool. This allows for some increased usage of the pool to occur without the pool being expanded. The values, however, should not be set so high as to defeat the purpose of dynamic expansion; the values should allow expansion to occur for any significant increase in the use of the pool.

Although the general formulas in Appendix C usually yield values that exceed most user's requirements, exceptions can occur. If a value for a specific pool is too low, it can be adjusted by using the information in the topic "Tuning Pool Values" in this chapter.

Considerations In Specifying Dynamic Expansion Values

Several factors should be considered in deciding on the values to be specified for *xpanno* and *xpanpt* for a particular pool.

One consideration is the frequency and magnitude of the peak demands for buffers in the pool. If peak demands occur at well-separated points in the day and the demand causes a sharp increase in the number of buffers being used, a relatively large *xpanno* value should be used. This causes each expansion of the pool to be relatively large and reduces the number of expansions that take place (thereby reducing the processing time used for the expansions). If, in contrast, the rise to the peak periods are more gradual or if smaller peaks appear more frequently, the *xpanno* value should be relatively smaller, allowing the expansion to be done gradually, as it is needed.

Another consideration involves the *xpanpt* value. If dynamic expansion values are specified, the *xpanpt* value must be larger than or equal to the *slowpt* value, thereby ensuring that the pool is expanded before slowdown processing occurs.

Still another consideration involves a decision about which are to be dynamically expanded. This decision might be based on information gathered by using the buffer use trace facility described later in this chapter. If the usage of a pool remains fairly stable throughout the day, just the basic allocation values can be set to cover that stable usage (thereby saving expansion time with little wastage of storage). If, however, the usage of a pool fluctuates significantly, dynamic expansion should be specified for the pool.

Tuning Pool Values

You can use the ACF/VTAM buffer use trace (described in Chapter 6) to adjust the ACF/VTAM pool values to accurately represent your requirements (such as network configuration and maximum transaction rate). One procedure for doing this is to (1) initially operate ACF/VTAM using the IBM-supplied or the user-calculated pool values, (2) fix additional and optional pageable pools (if any) in storage, (3) activate the buffer use trace, and (4) adjust the pool values as indicated by the trace data.

Adjusting Buffer Pool Values

When analyzing the ACF/VTAM buffer use trace data and adjusting the buffer pool values for initial allocation, consider these guidelines:

ACF/VTAM should be operated using the user's requirements for application programs and workload, for the network configuration, and for the maximum transaction rate.

If a specific pool often goes into slowdown mode or runs out of buffers, that pool's slowpt value should be decreased or its baseno value should be increased.

If a pool has a low number of requests, storage can be saved by reducing its baseno value.

For IOBUF, NPBUF, PPBUF, and UECBUF, the baseno value should be a multiple of the number of buffers for each page.

For basic mode, if an application program stops accepting input data, PPBUF must be large enough to hold all the data that ACF/VTAM can receive from the terminals connected to the application program (until the program begins accepting input data again). Therefore, do not assume that low utilization figures (from the buffer use trace) indicate a need to change the *slowpt* and *baseno* values for PPBUF.

The size of PPBUF is based on the assumption that BUFFACT=1 for all application programs (APPL definition statements) and BUFLIM=2 for all basic mode terminals (LOCAL definition statements and TERMINAL, VTERM, or COMP macro instructions). Therefore, the *baseno* value should allow two buffers for each basic mode terminal.

Putting Buffer Pools In Fixed Storage

By using the F attribute (value) for ACF/VTAM buffer pools, you can selectively fix pageable pools in storage. This can help you meet response time requirements. (For example, the transaction rate might be so small that pageable pool elements are paged out between transactions.) Fixing one or more buffer pools can also reduce the required number of buffers in each of the pools that support a specific network configuration or transaction rate.

For OS/VS1: If no location is specified, Subpool 0 is used by default for each ACF/VTAM buffer pool, except UECBUF. The default location for UECBUF is Subpool 241.

For OS/VS2 MVS: If no location is specified, Subpool 231 of common storage is used by default for each ACF/VTAM buffer pool, except UECBUF. The default location for UECBUF is Subpool 241.

For OS/VS2 SVS: If no location is specified, Subpool 250 is used by default for each ACF/VTAM buffer pool, except UECBUF. The default location for UECBUF is Subpool 252.

Guidelines for Fixing Buffer Pools

When assigning the F attribute to ACF/VTAM buffer pools, these guidelines can be used:

The F attribute can be used to fix the WPBUF buffer pool because it contains terminal-related control blocks (one for each terminal). Although the pages containing these control blocks are probably not referred to often enough to remain fixed in storage, ACF/VTAM frequently refers to them when processing transactions from that terminal.

The F attribute can be used to fix the CRPLBUF and LPBUF buffer pools (containing transaction-related control blocks) if the transaction rate is low or if a large amount of batch work is being processed. Otherwise, if ACF/VTAM is processing many transactions, the transaction rate can be large enough to keep the pages in real storage.

The APBUF, NPBUF, and SPBUF buffer pools should not be fixed unless there are a large number of logons, logoffs, VARY commands, and OPNDST requests. These pools contain control blocks to which ACF/VTAM does not refer when processing transactions.

Using Tuning Statistics

Some parameters specified in the NCP's HOST macro instruction and in ACF/VTAM definition statements affect the efficiency of data transfer in both directions between ACF/VTAM and a local SNA controller. (A local SNA controller can be a 3704 or 3705 communications controller or an SNA cluster controller.) This section discusses how ACF/VTAM tuning statistics can be used to gather information on operations between ACF/VTAM and the SNA controllers, and how that information can be used to adjust important operands in the NCP and ACF/VTAM definition statements.

General Way in Which ACF/VTAM Reads and Writes Data

Each channel program used by ACF/VTAM to write data to a SNA controller consists of a write channel program followed by a read channel program. If the controller has data ready to go to ACF/VTAM when finishes a write operation, ACF/VTAM immediately begins read that data without any prompting from the controller. If, however, the SNA controller has data to send to ACF/VTAM, and ACF/VTAM has not attempted to write or read during a specified time interval, or if the controller has reached a predefined buffer limit, the controller sends an attention to ACF/VTAM requesting that it start a read operation. If ACF/VTAM is able to accept the data, ACF/VTAM starts a read channel program to satisfy the request.

Therefore, ACF/VTAM can read data in either of two ways: as an immediate sequel to a write operation (which is fast and efficient) or as a separate operation initiated by an attention interruption from the SNA controller (which is less efficient).

The amount of data that ACF/VTAM can read in one operation depends on the number of buffers used by a read channel program and on the size of each buffer.

Basic Tuning Objectives

The basic objectives of tuning ACF/VTAM data-transfer operations are:

To read data from the controller as often as possible as an immediate sequel to an ACF/VTAM write operation, thereby reducing the number of attention interruptions that ACF/VTAM must process.

To read more than one path information unit (PIU) on each read operation.

These objectives can be met by adjusting parameters in the ACF/VTAM and NCP macro instructions.

Specifying Tuning Statistics

Tuning statistics can be specified with the TNSTAT start option (see Chapter 3), and this specification can be changed with the MODIFY network operator command (see *ACF/VTAM Network Operating Procedures).* Among the items that can be regulated are how often records are to be written, and whether the records are to be written only to the system management facility (SMF) file, or to that file and to the network operator's console.

Tuning Statistics Output

Each tuning statistics record contains information about the state of data-transfer operations between ACF/VTAM and one local SNA controller. Each record contains statistics that cover the time period since the last tuning statistics record was written for that controller.

This is the format of the tuning statistics report that appears (if requested) at the network operator's console:

```
IST440I   TIME=12402308   DATE=78079   LOCAL PC NAME=NCPLOC
IST441I   DLRMAX=1        CHWR=14      CHRD=15
IST442I   ATTN=15         RDATN=0      IPIU=15
IST443I   OPIU=14         RDBUF=15     SLODN=0
```

TIME

indicates the time (in hours, minutes, seconds, and hundredths of seconds) at which the record was recorded. In the previous example,12402308 means that the record was recorded at the 12th hour, 40th minute, 23rd second, and 8 one-hundredths of a second of the day.

DATE

is the date on which the tuning statistics report was recorded. The date is in the form yyddd, where yy are the last two digits of the numeric year and ddd is the numeric day of the year. In the example, 78079 means the record was recorded on the 79th day of 1978.

LOCAL PC NAME

is the name of the local SNA controller for which the statistics were gathered. A local SNA controller can be a 3704 or 3705 communications controller or an SNA cluster controller.

DLRMAX

a decimal value that indicates the maximum number of dump-load-restart requests that were awaiting processing or were being processed at one time during the interval. This number refers to the entire domain, not to the SNA controller named in the report. The dump-load-restart subtask processes these types of requests:

Dumping, loading, or restarting of an NCP

Some ACF/VTAM messages to the operator that require a reply

Connection and disconnection processing for a local subarea

Any I/O to a configuration restart or NODELST data set

This value can be used to determine the proper setting for the DLRTCB start option, which determines how many dump-load-restart requests can be processed concurrently. If DLRMAX consistently exceeds DLRTCB, it indicates that ACF/VTAM are serializing on the available TCBs and performance might be affected.

CHWR

indicates the total number of write channel programs issued.

CHRD

indicates the total number of read channel programs issued.

ATTN

indicates the total number of attentions (asynchronous I/O interruptions) received from the controller.

RDATN

indicates the total number of times that the attention was included in the ending status on a read channel program (that is, the number of times that ACF/VTAM, after reading data, was requested with an attention to read more data).

IPIU

indicates the total number of inbound (to ACF/VTAM) PIUs received from this controller.

OPIU

indicates the total number of outbound (from ACF/VTAM) PIUs sent to this controller.

RDBUF

indicates the total number of ACF/VTAM buffers used for read operations.

SLODN

indicates the total number of times the controller went into slowdown mode.

Effects of the DELAY Operand

The DELAY operand controls the length of time that a 3704 or 3705 communications controller holds data before it requests ACF/VTAM to read the data. The delay operand thereby affects the frequency with which the controller sends attentions to ACF/VTAM. ACF/VTAM operates more efficiently when it processes fewer attentions, so from an ACF/VTAM viewpoint a longer attention delay is desirable. A longer attention delay, however, increases the amount of data that the controller must hold in its buffers and also lengthens overall response time.

The tuning characteristics of the DELAY operand are:

If the DELAY time is too long, the response time can be poor.

If the DELAY time is too short, ACF/VTAM must process too many attentions.

Effects of the MAXBFRU Operand

For ACF/VTAM, MAXBFRU specifies the maximum number of buffers that can be used in one read operation when reading data from a controller. For the controller, MAXBFRU specifies how many buffers the controller must reserve for holding data awaiting transmission into ACF/VTAM.

ACF/VTAM operates more efficiently if MAXBFRU is set to a higher value because more buffers are available for each read operation. The controller, however, must be able to buffer both the current read channel program and the last read channel program; therefore, the higher the MAXBFRU value, the greater the demand on the controller's buffer resources.

The tuning characteristics of the MAXBFRU operand are:

If MAXBFRU is too low, many more ACF/VTAM read operations are required and, consequently, the number of attentions occurring on a read operation is higher.

If MAXBFRU is too high, the NCP enters slowdown mode frequently.

Effects of IOBUF *bufsize* Parameter

The IOBUF *bufsize* parameter affects the number of buffers that ACF/VTAM must use for each PIU to be transmitted.

If the *bufsize* is much larger than the average size of a PIU, storage is wasted because ACF/VTAM puts only one PIU into each buffer, even when more PIUs would fit in the buffer. On the other hand, if *bufsize* is smaller than the average PIU, ACF/VTAM breaks the PIU into blocks just large enough to fill one buffer and chains the buffers together. Therefore, when *bufsize* is too small, ACF/VTAM must do extra processing to handle the chaining and ACF/VTAM's I/O operations become less efficient.

The best results are obtained when *bufsize* is such that, on the average, slightly more than one buffer is used for each inbound PIU. If the average number of buffers used for each inbound PIU (found by dividing RDBUF by IPIU) is exactly one, the *bufsize* value is too large, and if it is greater than two, the *bufsize* value is too small.

Effects of the VPACING Operand

VPACING controls the amount of data that ACF/VTAM can send to a controller in one write operation. Indirectly, it influences the frequency with which ACF/VTAM can read data at the end of a write operation.

When the n parameter of VPACING is high, more writes can be done to the controller on each write operation, thereby reducing the amount of processing that ACF/VTAM must do for write operations. When the n parameter is low, fewer writes are done on each operation, thereby increasing the amount of processing that ACF/VTAM must do for write operations.

While a high n value enables ACF/VTAM to write more data, it also places a heavier demand on the controller's buffer resources. The amount of storage available in the controller for buffering its inbound and outbound data is limited. When more buffers must be devoted to outbound data, less storage is available for inbound data. Therefore, when the demands on the controller for both inbound and outbound buffers is heavy, the controller tends to enter slowdown processing more frequently. (When the controller is in slowdown processing, it does not read any data; it only sends data out.)

In terms of ACF/VTAM tuning statistics, the effects of the VPACING operands are:

If the n parameter of VPACING is too low, ACF/VTAM write operations occur less frequently and the controller must issue read attentions more often.

If the n parameter of VPACING is too high, there tends to be a greater demand on the controller's buffer resources, and the controller tends to enter slowdown processing more frequently.

Tuning Considerations

Tuning is a process of adjusting variables until processing meets the requirements of the user and the network. Figure 7-5, which shows some of the symptoms and causes of tuning inbalances, can help in the tuning process.

Other considerations for using the tuning statistics output are:

The value for SLODN should be low, but not zero. If the controller never goes into slowdown mode, it indicates that a substantial fraction of the controller's resources are never used, and therefore wasted. Generally, the ratio of controller slowdowns to the total number of PIUs processed should be about the same as the ratio of the number of hours of peak load to the total number of hours of operation.

The RDATN value should be as small as possible. A large RDATN value indicates that there are not enough ACF/VTAM read buffers.

The ATTN value should usually be less than the CHRD value. The smaller that ATTN is in proportion to CHRD, the greater the proportion of reads that were done as part of a write operation. If ATTN is about the same as CHRD, the DELAY value is too low.

An Approach To Tuning

Here is one approach to tuning the main parameters.

Set the DELAY, MAXBFRU, VPACING, and IOBUF bufsize values all to their maximum values, and after specifying tuning statistics, operate the network under both normal and peak loads. Using the output of the tuning statistics as a guide, do the following:

Decrease the DELAY value until the number of controller slowdowns (SLODN) becomes reasonable, or until the number of read attentions (RDATN) becomes significant.

Decrease the VPACING value until ACF/VTAM's inbound data rate (IPIU) becomes acceptable, or until ACF/VTAM's outbound data rate (OPIU) becomes too low.

Decrease the IOBUF bufsize value until the RDBUF value is about two or three times the IPIU value.

It might be necessary to adjust each of these variables several times before ACF/VTAM's I/O processing is satisfactorily tuned.

Sympton	Possible Causes			
	DELAY time	MAXBFRU value	VPACING value	IOBUF bufsize
Too many attentions (ATTN is high)	Too low		Too low	
Poor response time at low data rate	Too high			
Too many NCP slowdowns (SLODN is high)		Too high	Too high	
Too many attentions for read operation (RDATN is high)		Too low		Too small
ACF/VTAM's inbound data rate is poor (IPIU is low)			Too high	
ACF/VTAM's outbound data rate is poor (OPIU is low)			Too low	
RDBUF about the same as IPIU				Too large
IPIU much less than RDBUF				Too small
OPIU much less than CHWR			Too low	
No NCP slowdowns occur (SLODN is always 0)		Too low		
RDBUF less than (MAXBFRU x CHRD)		Too high		

Figure 7-5. Symptoms and Causes of Tuning Imbalance

Appendix A. Device Considerations

This appendix lists MTA line considerations and device-dependent information that affects the devices that are assigned to ACF/VTAM and the NCP and use basic mode. Some of the dependencies affect the coding of the macro instructions used by application programs, and some of the dependencies affect the coding of NCP macro instructions used by the NCP and ACF/VTAM.

For devices that use record mode, the device dependencies in the installation publication for the appropriate IBM data communication subsystem should be reviewed.

The device considerations for a specific device type in this appendix should be reviewed before coding the NCP macro instructions that also affect that device type; then this appendix should be used in conjunction with the device considerations provided in the *ACF/VTAM Macro Language Reference* for ACF/VTAM macro instructions, and in the *NCP Generation Manual* for NCP macro instructions.

SNA Devices

For a complete list of SNA devices supported by ACF/VTAM, see *ACF/VTAM Concepts and Planning.*

For device considerations for each SNA device, see the appropriate installation publication for the specific device.

For a discussion of the communication protocols that are applicable for a specific SNA terminal, refer to *ACF/VTAM Macro Language Reference.*

Non-SNA Terminals

IBM 1050 Data Communications System

Translation from lowercase to uppercase characters is not provided by either the NCP or ACF/VTAM.

TERMINAL And COMP Macro Instructions

If only one input and one output component are attached to the 1050 system, or if all components are used in group mode, one TERMINAL macro instruction can be used to define the 1050 system. Also, COMP macro instructions can be used for components with different characteristics, or for a 1050 system that uses specific polling or addressing characters.

If a 1050 system is on a switched line, the application program must insert the appropriate component selection character in the data transmitted to the terminal.

If a 1050 system is on a nonswitched line, the component selection character must be defined in the ADDR operand of the COMP macro instruction that represents each 1050 component. This definition allows the application program to establish connection and to transmit data to the terminal without having to insert a component selection character.

If COMP is used, the application program should issue an OPNDST macro instruction for each component and for the symbolic name representing the entire 1050 system, to prevent another application program from doing I/O with these 1050 components.

The DEVICE operand should be coded on the TERMINAL and COMP macro instructions to allow application programs to use the INQUIRE macro instruction with OPTCD= DEVCHAR. INQUIRE can then obtain the device characteristics for a particular 1050 component.

If CONV=YES is coded for output data to be sent in response to input from the terminal, the application program should initiate simultaneous WRITE requests without waiting for the pending READ to complete.

IBM 2740 Communication Terminal, Models 1 And 2

Translation from lowercase to uppercase characters is not provided by either the NCP or ACF/VTAM.

For Model 2 only, leading graphics are not translated but are returned to the application program as the image of the expected bit positions.

TERMINAL and COMP Macro Instructions

If CONV=YES is coded for output data to be sent in response to input from the terminal, the application program should initiate simultaneous WRITE requests without waiting for the pending READ to complete.

If only one output and one input component are attached to the 2740 terminal or if all components are used in group mode, one TERMINAL macro instruction can be used to define the 2740 terminal. COMP macro instructions can be used for components with different characteristics, or for a 2740 terminal that uses specific polling or addressing characters. If COMP is used, the application program should issue an OPNDST macro instruction for each component and for the symbolic name representing the entire 2740 system to prevent another application program from doing I/O with these 2740 components.

IBM World Trade Teletypewriter Terminals (WTTY)

Identification characters cannot be verified by either ACF/VTAM or the NCP.

GROUP Macro Instruction

If an application program specifies a NIB macro instruction with PROC=MSG, the WTTYEOB operand in the GROUP macro instruction must define the end-of-block sequence (if read-ahead is not desired for input operations).

If an application program specifies a NIB macro instruction with PROC=TRANS (the default value), the WTTYEOT operand in the GROUP macro instruction must define the end-of-transmission sequence. This sequence should not contain the FIGS or LTRS character if a WTT Adapter is attached.

TERMINAL Macro Instruction

If the terminal is equipped to send who-are-you sequences (WRU), the WRU sequence (any valid character except FIGS or LTRS) should be specified in the ADDR operand.

AT&T 83B3 Selective Calling Station and Western Union Plan 115A Outstation

If the terminal is used in group mode, one TERMINAL macro instruction can be used to define the group. COMP macro instructions can be used if the terminals are to be used in specific mode. If COMP is used, the application program should issue an OPNDST macro instruction for each component and for the symbolic name representing the entire AT&T 83B3 or Western Union Plan 115A system to prevent another application program from doing I/O with these components.

CPT/TWX (Model 33/35) Line Control Type

For ACF/VTAM to verify the ID sequence received from a dial-up TWX terminal, the VIDLIST macro instruction must be coded. For the NCP to verify the ID sequence, the IDLIST macro instruction must be coded.

MULTIPLE-TERMINAL-ACCESS (MTA) Line Considerations

The VTERM macro instruction can be used to permit independent connection to a dial-up terminal on an MTA line.

IBM 2770 Data Communication System

If a 2770 system is on a switched line and has more than one component, one TERMINAL macro instruction can be used to define the 2770 system. COMP macro instructions cannot be used to define the components. The application program must insert the appropriate component selection character in the data transmitted to the terminal.

If a 2770 system is on a nonswitched line and has more than one component, one TERMINAL macro instruction can be used to define the 2770 system. COMP macro instructions can be used to define each component. The ADDR operand of each COMP macro instruction defines the component selection character for each component. This definition allows the application program to establish connection and to transmit data to the terminal without having to insert a component selection character.

IBM 2972 Station Control Unit, Models 8 And 11

The CLUSTER macro instruction is coded if the 2972 is on a line that is a nonswitched multipoint line.

TERMINAL Macro Instruction

The NCP assigns a name to the terminal that corresponds to the symbol on the TERMINAL macro instruction representing the same terminal. The NCP also uses this assigned name to refer to the printer/keyboard. ACF/VTAM assigns a symbolic name to refer to the passbook printer or auditor key to allow connection to a passbook printer on a 2980 Model 1 or 4 Teller Station or an auditor key on a 2980 Model 2 Administrative Station. The name is formed by placing a dollar sign ($) to the left of the specified name and by deleting the last character. For example:

```
A2980TRM TERMINAL ADDR=(818140,8181F4),TERM=2980,. . .
```

A2980TRM is the NCP-assigned name for the printer/keyboard, and $A2980TR is the ACF/VTAM-assigned name for the passbook printer. The user must ensure that the ACF/VTAM-assigned name of the passbook printer or auditor key is unique.

IBM 3270 Information Display System (BSC And Locally Attached)

The CLUSTER macro instruction is coded if the 3271 or 3275 is on a line that is a nonswitched multipoint line.

IBM 3735 Programmable Buffered Terminal

The VIDLIST macro instruction can be used to verify IDs.

IDLIST Macro Instruction

If the NCP does not verify IDs using the IDLIST macro instruction, the BHSET and processing options (defined at NCP generation) are not associated with the first block of input received from a dial-up terminal.

LINE Macro Instruction

The WAIT option is specified in the POLIMIT operand when defining a line to which a 3735 is attached.

Nonswitched 3735s can coexist with other 3735s on a multipoint line. However, 3735s cannot coexist with other BSC terminals on a multipoint line because the POLIMIT requirements for 3735s is different from those for other BSC terminals.

IBM 3740 Data Entry System

The VIDLIST macro instruction can be used to verify IDs.

If the NCP does not verify IDs using the IDLIST macro instruction, the BHSET and processing options (defined at NCP generation) are not associated with the first block of input received from a dial-up terminal.

If a CPU ID sequence is to be sent to a 3740 system on a switched line from the NCP, the sequence is specified in the CUID operand (in the BUILD macro instruction) and in the CUIDLEN operand (in the TERMINAL macro instruction).

IBM System/3 CPU, IBM System/32, Or IBM System/370 CPU

If the System/3, IBM System/32, or System/370 CPU is to be the primary station on a nonswitched, point-to-point (contention) line, the LINE macro instruction is coded with the POLLED=NO, TADDR=NONE, and YIELD=YES operands. This coding makes the communications controller the secondary station and yields control to the primary station when contention occurs on the line.

Appendix B. Terminal-User Messages Issued By ACF/VTAM

This appendix describes the messages that might be displayed or printed at a terminal operated by a telecommunications user (as opposed to the system operator or network operator). The first half of Appendix B consists of the messages ACF/VTAM issues in response to LOGON or LOGOFF commands issued by logical units. The balance of Appendix B consists of the messages issued by the IBM-supplied network solicitor (NETSOL).

USS Messages

ACF/VTAM writes the messages described below in response to a character-coded logon or logoff command set by the logical unit. Each message shown indicates an error condition; no message shown in this appendix is issued when the command has been successfully processed.

The texts of the messages shown below are those contained in the IBM-supplied USS definition table. The description of the USSMSG macro instruction in Chapter 4 explains how the MSG operand is used to replace the tests of these messages. The messages are arranged in the order in which they are designated by the MSG operand.

The description of the MSG operand in Chapter 4 indicates that nine messages can be replaced, but ten messages are described below. The last message, MESSAGE NOT DEFINED, cannot be modified because it is issued only if the USS definition table has been improperly constructed or modified by the user. In addition to the nine messages you can replace, you can also create messages of your own to be issued after the successful completion of a USS command or when a logical unit is available for new user sessions. These messages, with message numbers 0 and 10, are created with the USSMSG macro instruction.

The issuance of any of these messages (other than 0 or 10) means that the command just entered has been ignored. Except where otherwise noted, the proper terminal operator response is to reenter the command.

MSG=1

INVALID COMMAND SYNTAX

Explanation: Some part of the command violates the rules described under "Character-coded Command Syntax" in Chapter 4.

System Action: The command is ignored.

Terminal-User Response: Determine the correct form and reenter the command.

MSG=2

command COMMAND UNRECOGNIZED

Explanation: The entered command (or its replacement) is neither LOGON nor LOGOFF.

System Action: The command is ignored.

Terminal-User Response: Determine the correct command and enter it.

MSG=3 parameter PARAMETER UNRECOGNIZED

Explanation: The entered parameter (or its replacement) is not valid for the command with which it is entered. If the parameter is a keyword parameter, the keyword parameter appears in the message text; if the parameter is a positional parameter, that parameter's position appears in the message text (for example, P3 PARAMETER UNRECOGNIZED).

System Action: The command is ignored.

Terminal-User Response: Determine the correct parameter and reenter the command.

MSG=4 parameter PARAMETER INVALID

Explanation: The variable value associated with a parameter is invalid. Either (1) the application program name or CDRSC name is unknown to ACF/VTAM, (2) ACF/VTAM cannot find the logon mode name in the appropriate logon mode tables, or (3) either the TYPE parameter (for LOGOFF) is neither COND nor UNCOND, or the HOLD parameter is neither YES nor NO.

System Action: The command is ignored.

Terminal-User Response: Determine the correct value (name) for the parameter and reenter the command.

MSG=5 UNSUPPORTED FUNCTION

Explanation: The logical unit sent the command to ACF/VTAM in an improper manner. If SSCPFM=USS3270 is specified for a logical unit, input can be entered from it only through the Enter key, the Clear key, System Request key, or a magnetic card reader. If SSCPFM=USSSCS is specified for a logical unit, it must not issue a Clear, Cancel, or Signal command; nor can it send a zero-length command. All logical units must send the character-coded command as a single-request chain.

System Action: The command is ignored.

Terminal-User Response: Reenter the command. Some of the invalid commands mentioned above are sent when program function keys are pressed; avoid pressing these keys.

MSG=6 SEQUENCE ERROR

Explanation: If the command just entered is a logon command, this message means that an application program is already connected to the logical unit.

System Action: The command is ignored.

Terminal-User Response: None. The command is already in effect.

MSG=7 SESSION NOT BOUND

Explanation: A valid logon request has been entered and forwarded to the application program, but the session is not established because (1) the application program has rejected the logon (by issuing a CLSDST instead of an OPNDST macro instruction), (2) the logical unit has rejected the application program's OPNDST macro instruction (by returning a negative response to the Bind command sent by the OPNDST), or (3) logon to be processed by logon-interpret routines was found to be invalid.

In addition, for cross-domain requests, this message may be issued because (1) communication cannot be established with CDRM that controls the desired cross-domain resource,

(2) the format of the CDINIT response is incorrect, or (3) a duplication of network addresses was discovered.

The logon command, as entered, cannot be used to establish a connection with the application program.

System Action: The command is ignored and no session is established between the application program and the logical unit.

Terminal-User Response: It is possible that the application program cannot accept the session parameters specified by the logon mode name in the logon command or the logical unit cannot accept the application program's substituted session parameters; in this situation, a logon command specifying a different logon mode name might work. This message is generally evidence of improper design of the application program, the USS definition table, or the logical unit's application program.

MSG=8

INSUFFICIENT STORAGE

Explanation: ACF/VTAM was unable to obtain enough storage to service the request. If the user has allocated an optimum partition size for ACF/VTAM, this condition is temporary and rare. If too little storage has been allocated, this condition occurs frequently and can be corrected only by increasing the size of the ACF/VTAM partition. (An identical situation occurs when application programs issue ACF/VTAM requests; see the description of error return codes in *ACF/VTAM Macro Language Reference.*)

System Action: The command is ignored.

Terminal-User Response: Reissue the command.

MSG=9

MAGNETIC CARD DATA ERROR

Explanation: A character-coded command from a logical unit for which SSCPFM= USS3270 is coded contains invalid magnetic card data. Either the card was entered into a field that was too small, or a parity error occurred.

System Action: The command is ignored.

Terminal-User Response: Reenter the command. If the magnetic card has been entered into a field that was too small, press the CLEAR key and reenter the command, entering the card into a larger field. If parity errors are suspected, notify IBM maintenance personnel.

Undefined Message

MESSAGE NOT DEFINED

Explanation: An error condition associated with one of the preceding messages occurred, but ACF/VTAM cannot find an USSMSG macro instruction in the logical unit's USS definition table that corresponds to the error condition. For example, an insufficient storage condition occurred, but neither the logical unit's USS definition table nor the IBM-supplied table contains an USSMSG macro instruction with MSG=8 specified. (Since the IBM-supplied table defines all messages, the user has deleted the message.) This message is evidence that the user has improperly defined or installed the USS definition tables.

System Action: The command is ignored.

Terminal-User Response: Check the command for any obvious errors and reissue the command. There is no way to determine which of the above nine error conditions apply.

Network Solicitor Messages

The messages described below can be issued to any BSC or start-stop terminal serviced by the network solicitor. The message texts as shown here are those issued by the IBM-supplied network solicitor.

The messages from the network solicitor do not contain identifying numbers or message identifiers. The text of the messages can be shortened, extended, or translated by following the procedures in the "Defining Connection and Disconnection Procedures" in this publication.

The information in this appendix should be made available to the terminal users who can receive the messages and to the network operator. However, before making this information available, consider these guidelines:

Rewrite the message text if it was changed when generating a network solicitor besides the IBM-supplied network solicitor.

Change "Terminal-User Response" to reflect the user's procedures or policies.

Delete "Programmer Response" for each message. This information can assist an application or system programmer (not the terminal user or network operator) to recognize the programming considerations that cause the message to be issued.

The messages are listed alphabetically below.

APPLICATION DEACTIVATED BY NETWORK OPERATOR

Explanation: The network operator has issued a command to deactivate the application program that you require. Your logon request is ignored.

System Action: The logon request is not accepted.

Terminal-User Response: Tell the network operator which application program you want to use. When the network operator tells you that the application program is active, reenter your logon request.

Programmer Response: The application program appeared to be active and accepting logon requests; however, the network solicitor issued a CLSDST macro instruction (with OPTCD=PASS) that failed. This occurred because the network operator deactivated the application program after the network solicitor issued the INQUIRE macro instruction (with OPTCD=APPSTAT), but before the network solicitor issued the CLSDST macro instruction (with OPTCD=PASS).

APPLICATION IS CLOSING DOWN

Explanation: The application program you requested is disconnecting itself from ACF/VTAM. Your logon request is not accepted.

System Action: The logon request is not accepted.

Terminal-User Response: Tell the network operator which application program you want to use. When the network operator tells you that the application program is active, reenter your logon request.

Programmer Response: The application program has issued a SETLOGON macro instruction (with OPTCD=QUIESCE). The application program's logon request queue is permanently closed (the application program is in the process of closing its ACB).

APPLICATION IS INACTIVE

Explanation: The application program you requested is not currently active. Your logon request is ignored.

System Action: The logon request is not accepted.

Terminal User Response: Tell the network operator which application program you want to use. When the network operator tells you that the application program is active, reenter your logon request.

Programmer Response: The application program's ACB is not open.

APPLICATION IS NOT ACCEPTING LOGONS

Explanation: The application program you requested is not accepting logon requests. The condition probably is temporary. Your logon request is not accepted.

System Action: The logon request is not accepted.

Terminal-User Response: Wait a while and try to logon again. The application program has probably reached the user-defined limit on the number of logon requests that it can accept.

Programmer Response: The application program has issued a SETLOGON macro instruction (with OPTCD=STOP), which implies that the condition is temporary. The application program should issue a SETLOGON macro instruction (with OPTCD=START) as soon as it can accept another logon request.

APPLICATION NEVER ACCEPTS LOGONS

Explanation: The application program you requested can only acquire terminals; it does not accept logon requests from the network solicitor. Your logon request is not accepted.

System Action: The logon request is not accepted.

Terminal-User Response: Ensure that this is the application program you really require. If not, enter another logon request, specifying the correct application program. If it is the correct program, ask your system programmer for assistance.

Programmer Response: The application program indicates that it does not accept logon requests (that is, the application program's ACB was opened with MACRF=NLOGON specified). For the application program to accept logon requests, add an OPEN macro instruction with MACRF=LOGON, followed by a SETLOGON macro instruction. Then recompile and link-edit the application program.

APPLICATION UNKNOWN TO VTAM

Explanation: ACF/VTAM does not recognize the application program name that corresponds to the logon message you specified. Your logon request is ignored.

System Action: The logon request is not accepted.

Terminal-User Response: Ensure that you specified the correct logon message. If not, enter the correct logon message. If it is the correct message, ask your system programmer for assistance.

Programmer Response: The application program name (1) has been incorrectly specified in the LOGCHAR macro instruction that defines the application program name in the interpret table, (2) has been incorrectly specified in the APPL definition statement, (3) the configuration being used at this time does not have a APPL definition statement defining the application program, or (4) the application program major node has not been activated.

ERROR WHEN READING LOGON MESSAGE

Explanation: An I/O error (hardware) was detected in the logon message, or the logon message contained no data. Your logon message is ignored.

System Action: The logon request is not accepted.

Terminal-User Response: Reenter the logon message.

Programmer Response: If the error persists, contact maintenance personnel.

INPUT NOT RECOGNIZED

Explanation: You entered a logon message (possibly null) that the network solicitor has compared against the interpret table and found invalid.

The IBM-supplied network solicitor does not convert lowercase logon messages into uppercase, so this message can be produced if both lowercase and uppercase versions of a logon message are not included in the interpret table.

System Action: The logon request is not accepted.

Terminal-User Response: Enter a valid logon message (user-defined), or if you believe the logon message is valid, ask the system programmer for assistance.

Programmer Response: Ensure that the correct logon messages were given to the terminal users. Otherwise, the logon message defined in the interpret table by the LOGCHAR macro instruction is either incorrect or the terminal user did not use the correct logon message.

NO INTERPRET TABLE

Explanation: Your logon message was not in the required format, and no interpret table exists for your terminal. Your request is ignored.

System Action: Your logon request is ignored, but your terminal is resolicited so that the logon can be retried with the correct format.

Terminal-User Response: (1) Repeat the logon request in OS/VS format: LOGON APPLID (applname) where applname is the name of the application program to which logon is desired. Note that the application program can require additional text entered with the logon message to fully authorize you to use this application program. (2) Ask the network operator to log you on with the VARY command. (3) You can also ask the system programmer to consider specifying an interpret table or logon message in an interpret table for your terminal.

Programmer Response: No action is required.

NO ROUTINE TO RECOGNIZE APPLICATION

Explanation: ACF/VTAM could not recognize the application program you requested because the user-supplied logon-interpret routine to do this has not been loaded. Your request is ignored.

System Action: The logon request is not accepted.

Terminal-User Response: Have the network operator check that the interpret table defined for your terminal is available to ACF/VTAM.

Programmer Response: The user-supplied routine referred to by this message is the logon-interpret routine. The interpret table used by this terminal is specified in the LOGTAB operand (in the appropriate TERMINAL macro instruction). The interpret table and any user-supplied logon-interpret routines (referred to by the interpret table) should have been assembled together and link-edited using the name of the interpret table (in the INTAB macro instruction).

USER UNAUTHORIZED FOR THIS APPLICATION

Explanation: The user-defined authorization specifications have not recognized your request and therefore have not allowed you to be connected to the application program. Your request is ignored.

System Action: The logon request is not accepted.

Terminal-User Response: You must arrange with the system programmer for the necessary authorization to use this application program.

Programmer Response: The logon-interpret routine referred to by the terminal's interpret table did not authorize the logon. If the terminal user is authorized, provide the additional information to allow the terminal user's logon message to be recognized by the logon-interpret routine.

THIS TERMINAL IS LOGGED ON TO THE NETWORK SOLICITOR

Explanation: This message is issued by the network solicitor when it receives control of the terminal (whether through a network operator command or as a result of another application program releasing the terminal at the end of a session) and successfully establishes a session with it.

System Action: The terminal remains in session with the network solicitor until another application program requests control of it or until the network solicitor requests that the terminal be logged on to another application program.

Terminal-User Response: You can now log on to an application program.

Programmer Response: None.

Appendix C. Storage Estimates and Buffer Pool Calculations

This appendix provides formulas for buffer pool calculations and work sheets for storage estimates. This edition contains information for OS/VS2 MVS, OS/VS2 SVS, and OS/VS1.

Calculation of Buffer Pool Values

Buffer pool usage depends upon several factors that are installation dependent. These factors are:

- Physical network configuration
- Network operating procedures
- Types of application programs being run
- Network through-put required
- CPU capability
- Use of dynamic buffering

Because of these factors, the buffer pool formulas in this appendix should be used to calculate starting values to be used only until a buffer pool trace can be run to determine the appropriate values for your specific network and application programs.

A buffer pool trace should be used to evaluate ACF/VTAM buffer pool usage when installing ACF/VTAM and when changes are made to the network. This evaluation will help you estimate how many buffers are needed for normal and peak load operations in your particular network. It will also help you determine the best way to use ACF/VTAM's buffer pool expansion facility.

Figure 7-4 in Chapter 7 shows all IBM-supplied buffer pool values. These values will be assumed by default if you do not specify your own values.

Calculation of baseno Values

Figure C-1 gives formulas for calculating *baseno* values for all OS/VS systems (A) without the dynamic expansion facility and (B) with the dynamic expansion facility. All buffers with a *slowpt* value should be added to the buffer *baseno.*

(A) is a starting value to be used only until a buffer pool trace can be run to determine the appropriate values for your specific applications and network.

(B) is a starting value to be used if the buffer pool expansion facility is used. Buffer pool trace should then be run to determine the steady state value for your buffer pools.

Buffer Pool Name	Formula for Calculating the Number of Buffers
APBUF (A)	2(NRECORDSES+NBASICSES)[7]+NQDSIMLOG+NLTSO
APBUF (B)	NRECORDSES+NBASICSES
CRPLBUF (A)	(NBASICSES+NRECORDSES)[1]+NPORT+NTERM+NBSCCLUS+NRTSO+ (2 X NLTSO)+25
CRPLBUF (B)	Value must be determined by buffer trace
IOBUF[5] (A)	(5 X MAXBFRU_{sum})+2(NBUFMSG X NLOCTERM)
IOBUF (B)	Value must be determined by buffer trace
LFBUF[4] (A) and (B)	NACB+4
LPBUF (A)	2(NACB+4)+NPU+NBSCCLUS+NCOMMAND+10
LPBUF (B)	2(NACB+4)+10
NPBUF (A)	NBASICSES[2]+NLTSO
NPBUF (B)	NBASICSES
PPBUF[6] (A)	(BUFFACT_{sum} X BUFLIM_{sum} X NBASICSES)+NQDSIMLOG
PPBUF (B)	Value must be determined by buffer trace
SFBUF (A)	NACB+4 (OS/VS1 and OS/VS2 SVS only)
SFBUF (A)	2(NACB+4)+(NBASICSES+RECORDSES)[3]+8 (OS/VS2 MVS only)
SFBUF (B)	Value must be determined by buffer trace
SPBUF (A)	NBASICSES+NLTSO
SPBUF (B)	Value must be determined by buffer trace
UECBUF (A)[8]	2(NBASICSES+NRECORDSES+NAPPLSES)
UECBUF (B)	Value must be determined by buffer trace
WPBUF (A)	NRECORDSES+NCOMMANDS
WPBUF (B)	NRECORDSES

Notes:

1. (NBASICSES+NRECORDSES) assumes all OPNDSTs and CLSDSTs are concurrent. This expression can be replaced by the maximum number of pending ACF/VTAM macro instructions.
2. Include NTERM if using the network solicitor.
3. The number of elements used in this pool depends upon the CPU's capacity. If using TSO, this value (NBASICSES+NRECORDSES) should be zero.
4. LFBUF does not apply to application programs that specify EAS (on the APPL statement) less than 3239. LFBUF does not apply to TSO.
5. IOBUF represents a starting value until the actual requirements can be determined by a buffer trace.
6. If NBASICSES = 0, specify PPBUF = 1. Segmented messages will be held in PPBUF until the entire message is received (basic mode only).
7. (NRECORDSES+NBASICSES) assumes all CLSDST macro instructions are concurrent. This expression can be replaced by (NRECORDSES+NBASICSES)+K, where K is the number of concurrent CLSDST macro instructions.
8. This value assumes one NCP and one LOSTERM exit routine used for failures. This may be reduced if multiple NCPs are used. If an NSEXIT is used, only one element would be used per session.

Figure C-1 (Part 1 of 2). Formulas for Calculating *baseno* Values (A) without Dynamic Expansion and (B) with Dynamic Expansion

Variables for Formulas

$\text{BUFFACT}_{\text{sum}}$ and $\text{BUFLIM}_{\text{sum}}$ — Two decimal integers whose product is used to determine the maximum number of buffers that can be filled with data that has been obtained by ACF/VTAM but has not yet been transferred to the application program's buffers.

MAXBFRU — Parameter specified in NCP HOST macro instruction or local SNA PU statement.

$\text{MAXBFRU}_{\text{sum}}$ — Sum of all MAXBFRU values if more than one local NCP or local SNA major node is active.

NACB — Number of OPEN ACBs. (Include the number of TSO users.)

NBASICSES — Maximum number of concurrent active basic-mode sessions. (Include TOLTEP and 3270 BSC basic-mode sessions.)

NBSCCLUS — Number of remotely attached 3271, 3275, and 2972 cluster controllers on BSC (PU=NO) lines.

NBUFMSG — Number of buffers required to read the full screen of a 3277 display station. If both Models 1 and 2 are attached, use the Model 2 table. *ACF/VTAM Concepts and Planning* lists the devices supported as 3277 Models 1 or 2.

Read Full 3277 Model 1		Read Full 3277 Model 2	
bufsize	NBUFMSG	bufsize	NBUFMSG
104	4	96	16
256	2	216	8
632	1	632	4
		960	2
		1952	1

NCOMMAND — Number of concurrent operator commands.

NLOCTERM — Number of local non-SNA devices.

NLU — Number of logical units (number of LU statements). (Include local SNA devices.)

NPORT — Number of dial-up switched lines.

NPU — Total number of locally attached record-mode cluster controllers, plus the number of remotely attached SNA cluster controllers on SDLC lines.

NQDSIMLOG — Number of contending connection requests (SIMLOGON) that have been issued with the Q option. (This occurs when one or more application programs request the same logical unit or terminal when the resource is not available, and the application programs are willing to wait until the resource is available.)

NRECORDSES — Maximum number of concurrent active record-mode sessions. (Include 3270 BSC record-mode sessions, application program-to-application program sessions and the number of TSO users.)

NTERM — Total number of non-SNA terminals locally attached; count each printer and display station; plus the number of remotely attached terminals on BSC and start-stop lines (number of TERMINAL and COMP macro instructions on BSC and start-stop lines).

NTRACE — Number of terminals, lines, or logical units for which an ACF/VTAM buffer content trace (TYPE=BUF) will be started.

NAPPLSES — Number of application program-to-application program sessions.

NRTSO — Number of remote TSO users.

NLTSO — Number of local TSO users.

Figure C-1 (Part 2 of 2). Formulas for Calculating *baseno* Values (A) without Dynamic Expansion and (B) with Dynamic Expansion

Calculation of slowpt Values

For IOBUF, *slowpt* should be .20 times the IOBUF *baseno value.*

For CRPLBUF, *slowpt* should be .08 times the CRPLBUF *baseno* value.

For APBUF, NPBUF, PPBUF, and UECBUF, *slowpt* should be .10 times the *baseno* value for each pool.

For all other buffer pools, *slowpt* should be 0.

Calculation of bufsize Values

Normally, for all pools other than IOBUF and PPBUF, the *bufsize* values should be equal to the IBM-supplied values for the pool. Depending on the devices to be supported, choose an initial IOBUF *bufsize* from the following overview. The *bufsize* value for PPBUF should be equal to the IOBUF value.

Device Support	***bufsize* Calculation**
Local non-SNA	If the network only has local non-SNA terminals, calculate *bufsize* as the average message length for 3277 display stations.
3704 or 3705	*bufsize* must equal the UNITSZ operand on the HOST macro instruction. If multiple NCPs are active concurrently, UNITSZ must have the same value for each NCP.
Local SNA cluster controller only. Large batch transmission only.	*bufsize* = 1978 MAXBFRU for local SNA = 1
Local SNA cluster controller only. Normal transmission.	*bufsize* = 288 MAXBFRU for local SNA = 1 (for interactive transmission) MAXBFRU for local SNA = 2 (for batch transmission)
If priority is assigned to a 3704 or 3705 that is used with a local SNA cluster controller.	*bufsize* = 3704/3705 *bufsize*, or 288, whichever is larger
With 3704/3705 given priority, large batch performance of an SNA cluster controller might be less efficient because of multiple buffer usage, and the local SNA cluster controller might use more storage than necessary because of the priority given to the 3704/3705.	MAXBFRU for local SNA = 1 (for normal transmission) MAXBFRU for local SNA = 2 (for interactive transmission)
If priority is assigned to a local SNA cluster controller that is used with a 3704/3705.	*bufsize* = 288 for normal transmission *bufsize* = 1978 for large batch transmission
With priority given to a local SNA cluster controller, 3704/3705 buffers might not optimize storage.	MAXBFRU for local SNA = 1 (for large batch and normal transmission) MAXBFRU for local SNA = 2 (for normal batch transmission)

ACF/VTAM Storage Requirements in OS/VS2 MVS

Figures C-2, C-3, and C-4 are work sheets for estimating the storage requirements for an ACF/VTAM network under OS/VS2 MVS.

Fixed Storage Constant Requirements

Nucleus	1832	
LSQA (ACF/VTAM ADDR space)	24576	
CSA	16384	
PLPA	16384	
Total Constant Requirements =		59,176

Fixed Storage Variable Requirements

Buffer Pools (in CSA) (See fixed pool work sheet C-4.)

IOBUF ________
SFBUF ________
LFBUF ________

CSA

Number of local NCPs ______ X 464 = ________

Number of local SNA controllers ______ X 464 = ________

Number of application program address spaces ______ X 176 = ________

Number of local non-SNA devices ______ X 416 = ________

Number of open ACBs ______ X 152 = ________

Number of addressable nodes ______ X 8 = ________

PLPA (add in appropriate values)

Local 3704/3705 or Local 3790 (0 if no 3704/3705s or 3790s)	add	8192	=	________
Local non-SNA devices	add	12288	=	________
Local SNA devices	add	4096	=	________
Local NCP (Release 5)	add	4096	=	________

Total Fixed Storage Requirements (Sum of Constant and Variable Requirements Above) ________

Figure C-2 (Part 1 of 4). ACF/VTAM Storage Requirements for OS/VS2 MVS

Pageable Storage Constant Requirements

CSA = 16,384
PLPA = 491,520

Pageable Storage Variable Requirements

Buffer Pools (in CSA) (See paged pool worksheet C-3.)

SPBUF = ________
LPBUF = ________
UECBUF = ________
WPBUF = ________
APBUF = ________
PPBUF = ________
NPBUF = ________
CRPLBUF = ________
Total Pageable Buffer Pools = ________

Find the Sum of the Following Products:

Number of concurrent active sessions ________ X 5 = ________
Total number of *all* major nodes
(except application program major nodes) ________ X 128 = ________
Number of concurrently active remote PUs ________ X 112 = ________
Number of open ACBs ________ X 904 = ________
Number of concurrent OPNPSTs or CLSDSTs ________ X 872 = ________
Constant + 96 = ________
SUMA = ________

In the following calculations, the macro instructions referred to are those used by ACF/VTAM. See the list of macro instructions listed under "NCP Generation Procedure for ACF/VTAM" in Chapter 2. Those macro instructions marked "NCP-only" should be excluded from these calculations, the rest should be included.

Number of ACF/VTAM-used macro instructions in
NCP Generation deck ________ X 40 = ________
Round to a multiple of 4096 ________
Add 4096 +4096
SUMB = ________

Number of CDRM, CDRSC, and associated
VBUILD statements ________ X 264 = SUMC = ________

Number of ACF/VTAM-used macro instructions in
NCP generation deck used to define
SNA devices ________ X 212 = SUMD = ________

Number of ACF/VTAM-used macro instructions in
NCP generation deck used to define
non-SNA devices ________ X 272 = SUME = ________

Figure C-2 (Part 2 of 4). ACF/VTAM Storage Requirements for OS/VS2 MVS

CSA Variable Requirements = SUMA + SUMB + SUMC + SUMD + SUME = ____________

Total of common pageable storage = sum of CSA and PLPA constant requirements plus buffer pools and CSA variable requirements. □

Total common virtual storage = sum of fixed virtual storage plus the sum of the pageable virtual storage minus 24,576 (LSQA for ACF/VTAM address space) and minus LSQA for the application program address spaces. □

ACF/VTAM Address Space Constant Requirements

System region	=	20,480	
ACF/VTAM modules	=	589,824	
SWA, subpools 229/230	=	126,975	
Total constant requirements	=		737,279

ACF/VTAM Address Space Variable Requirements

If the ACF/VTAM Encrypt/Decrypt Feature is installed, enter the total number of concurrently active cryptographic sessions and multiply by 16.

____________ X 16 = ____________

If the ACF/VTAM Encrypt/Decrypt Feature is installed, enter 6298; otherwise enter 0.[1] ____________

If NETSOL = YES, enter 12410. NO enter 0 = ____________

Number of concurrent TOLTEP users ________ X 34816 = ____________

If tuning statistics will be used enter 2560; if not enter 0 = ____________

Number of addressable nodes in local domain______ X 200 = ____________

ACF/VTAM internal trace page number value specified by user ______ X 4096 = ____________

Number of cross-domain LUs______ X 72 = ____________

Number of addressable nodes, same domain, ______ X 2336 = ____________

Number of active sessions______ X 88 = ____________

Total equals above Variable Requirements plus 737, 279 Constant Requirement ____________

ACF/VTAM Address Space Requirement Total = ____________

Figure C-2 (Part 3 of 4). ACF/VTAM Storage Requirements for OS/VS2 MVS

Application Program Address Space

User program modules, control blocks (RPLs, NIB, etc.) and workspace size = ____________

Pageable Data Area

If VPACE is not specified in the APPL VBUILD statement, calculate:
((NRECORDSES X 2)+NQDSIMLOG)(UNITSZ+48)= ____________

If VPACE is specified in the APPL VBUILD statement, calculate:
(((2N[2] – 1) X NRECORDSES)+NQDSIMLOG)(UNITSZ+48)= ____________

System region[3] ____________

SWA subpools 229 or 230[3] ____________

Total Application Program Address Space = ____________

[1] Although the actual size increment is 6298 bytes, it is possible (due to the way OS/VS2 MVS loads modules) that this additional storage will fit in the space at the end of a page or that it will spill on to a new page. This means that the net size increment might be zero or as much as 16K bytes, depending on how the OS/VS MVS loader loads the modules.

[2] N = the number of elements in a receive chain.

[3] Calculated using *OS/VS2 Storage Programming Library Storage Estimates,* GC28-0604.

ACF/VTAM Real Storage Requirements for Record-Mode Sessions

Total fixed storage requirements (see figure C-2, Part 1) = ____________

Pageable working set modules = 45056

Pageable buffer pools (See buffer pool worksheet figure C-3):

WPBUF = ____________
LPBUF = ____________
CRPLBUF = ____________

Number of concurrent active record-mode sessions ______ X 5 = ____________

Constant 8196

Total Real Storage Requirement[4] ____________

[4] This requirement only reflects real storage needed to maintain normal network data flow. During connection, disconnection, activation, termination and error recovery processing; additional real storage will be needed by your system. Additional real storage will also be needed for mainline instructions, control blocks, and tables. Use the buffer pool trace to determine these requirements.

Figure C-2 (Part 4 of 4). ACF/VTAM Storage Requirements for OS/VS2 MVS

For each pool below, get the *baseno* value for that pool from Figure C-1 and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
APBUF	72	X	________	=	________
CRPLBUF	128	X	________	=	________
LPBUF	1024	X	________	=	________
NPBUF	208	X	________	=	________
PPBUF	(UNITSZ+48)	X	________	=	________
SPBUF	112	X	________	=	________
WPBUF	176	X	________	=	________
UECBUF	120	X	________	=	________
The sum of the products is SUM A.			SUM A	=	________

The PAGEPOOLS value is SUM A rounded up to a multiple of 4096. ________

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-3. Calculating Storage Required for Paged Pools in OS/VS2 MVS

For each pool below, get the *baseno* value for that pool from Figure C-1, and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
SFBUF	80	X	________	=	________
LFBUF	128	X	________	=	________
IOBUF	(UNITZ+83)	X	________	=	________
					+2048
The sum of the products is SUM A.			SUM A	=	________

Round SUM A up to a multiple of 4096. ________

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-4. Calculating Storage Required for Fixed Pools in OS/VS2 MVS

ACF/VTAM Storage Requirements in OS/VS1

ACF/VTAM runs in a problem program partition in OS/VS1. This partition contains the ACF/VTAM task and its subtasks, and most of the ACF/VTAM modules and storage pools. Other areas of the OS/VS1 operating system that are used and enlarged by including ACF/VTAM are:

- Nucleus
- Fixed SQA
- Pageable SQA
- Pageable supervisor
- User partition

The amount of storage ACF/VTAM required in each of these areas is estimated in the appropriate worksheets. Figure C-5 shows the overall structure of the calculations. Figure C-1 contains formulas for buffer pool calculations. Figure C-6 contains storage estimate work sheets.

Buffer size Buffer number Slow down point	See Figure C-1
PAGEPOOLS	See Figure C-6, Work Sheet 1
Fixed pools	See Figure C-6, Work Sheet 2
Pageable partition storage	See Figure C-6, Work Sheet 3
RDT size	See Figure C-6, Work Sheets 4A-10A
SYSDEF size	See Figure C-6, Work Sheets 4B-10B
SYSALOAD	See Figure C-6, Work Sheet 11
IOAPPN	See Figure C-6, Work Sheet 12
Fixed SQA	See Figure C-6, Work Sheet 13
Pageable supervisor	See Figure C-6, Work Sheet 14
Storage requirements; user partition	See Figure C-6, Work Sheet 15
Total fixed storage requirement	See Figure C-6, Work Sheet 16A
Real storage requirements	See Figure C-6, Work Sheet 16B
ACF/VTAM partition size	See Figure C-6, Work Sheet 17

Figure C-5. Overview of ACF/VTAM Storage Requirements in OS/VS1

Work Sheet 1. Calculation of Storage Required for PAGEPOOLS

For each pool below, get the *baseno* value for that pool from Figure C-1 and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
APBUF	72	X	________	=	________
CRPLBUF	128	X	________	=	________
LPBUF	1024	X	________	=	________
NPBUF	208	X	________	=	________
PPBUF	(UNITSZ+48)	X	________	=	________
SPBUF	112	X	________	=	________
WPBUF	176	X	________	=	________
UECBUF	120	X	________	=	________
The sum of the products is SUM A.			SUM A	=	________

The PAGEPOOLS value is SUM A rounded up to a multiple of 2048. ________

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-6 (Part 1 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 2. Calculation of Storage Required for Fixed Pools

For each pool below, get the *baseno* value for that pool from Figure C-1, and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
SFBUF	80	X	________	=	________
LFBUF	128	X	________	=	________
IOBUF	(UNITSZ+31)	X	________	=	________
					+2048
The sum of the products is SUM A.			SUM A	=	________

Round SUM A up to a multiple of 2048. ________

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-6 (Part 2 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 3. Calculation of ACF/VTAM Variable Pageable Partition Storage for NWSFMCB

Enter the values indicated below. The meaning of the variable names (such as NBSCCLUS) used below can be found in Figure C-1.

N370X (local and remote) ________
Total number of local SNA controllers ________
NBSCCLUS ________
NPU ________
NLU ________
TERMINAL (with PU = YES) ________

Find their sum and multiply by 216. ________ X 216 = ________

The result is the NWSFMCB value.

Figure C-6 (Part 3 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 4A. Calculation of RDT Size for NCP Major Nodes

This calculation must be made for every NCP major node in the ACF/VTAM network. The input for these calculations is obtained from the NCP definition deck.

Enter the number of the following definition statements and operands used to define the NCP and multiply by the indicated multiplier:

PCCU	________	X	280	=	________
DIALSET	________	X	96	=	________
GROUP	________	X	108	=	________
LINE (with DIAL=NO)	________	X	132	=	________
LINE (with DIAL=YES)	________	X	176	=	________
TERMINAL	________	X	260	=	________
COMP	________	X	260	=	________
VTERM	________	X	260	=	________
CLUSTER or PU (on non-SDLC lines)	________	X	112	=	________
CLUSTER (on SDLC lines)	________	X	180	=	________
PU (on SDLC lines)	________	X	180	=	________
PU (on SDLC dial lines)	________	X	100	=	________
PU (INNODE) for remote 3704 or 3705	________	X	192	=	________
LU	________	X	200	=	________
LUPOOL value	________	X	100	=	________

Then add the products. The result is the RDT size.

RDT size = ________

Figure C-6 (Part 4 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 4B. SYSDEF Work Area for NCP Major Nodes

This calculation must be made for every NCP that will be active in a 3704 or 3705 in your domain of the ACF/VTAM network. The input for these calculations is obtained from the NCP definition deck.

Enter the number of the following definition statements and operands used to define the NCP for this 3704 or 3705 and multiply by the indicated multiplier:

PCCU	________	X	280	=	________
DIALSET	________	X	96	=	________
GROUP	________	X	108	=	________
LINE (with DIAL=NO)	________	X	132	=	________
LINE (with DIAL=YES)	________	X	176	=	________
TERMINAL	________	X	260	=	________
COMP	________	X	260	=	________
VTERM	________	X	260	=	________
CLUSTER (PU) on non-SDLC lines	________	X	112	=	________
PU (on SDLC dial lines)	________	X	100	=	________
PU (on SDLC lines)	________	X	180	=	________
PU (INNODE) for remote 3704 or 3705	________	X	192	=	________
LU	________	X	200	=	________
LUPOOL value	________	X	116	=	________
VIDLIST	________	X	36	=	________
VIDSEQ	________	X	44	=	________
MTALIST	________	X	16	=	________
MTALCST	________	X	20	=	________

The sum of the products is SUM A. SUM A = ________

Subtotal A is SUM A rounded up to a multiple of 2048. Subtotal A = ________

The SYSDEF work area for NCP major nodes =

Subtotal A ________ + 10084 = ________

Figure C-6 (Part 5 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 5A. Calculation of the RDT size for All Switched SNA Major Nodes

This calculation must be made for every switched major node to be activated in the ACF/VTAM network. The input for these calculations is obtained from the switched SNA major node definition deck.

Enter the total number of the following definition statements and operands, and multiply by the indicated values:

VBUILD	________	X	176	=	________
PU	________	X	180	=	________
LU	________	X	200	=	________
unique MAXNO	________	X	33	=	________
unique MAXGRP	________	X	8	=	________
unique PATH	________	X	8	=	________
Add resulting products to get the RDT size. RDT size =					________

Work Sheet 5B. Calculation of the SYSDEF Work Area for Switched SNA Major Nodes

This calculation must be made for every switched SNA major node to be activated in the ACF/VTAM network. The input for these calculations is obtained from the switched SNA major node definition deck.

Enter the total number of the following definition statements and multiply by the indicated values (these calculations should have already been made for Work Sheet 5A):

VBUILD	________	X	176	=	________
PU	________	X	180	=	________
LU	________	X	200		________
Add the resulting products.			SUM	=	________
Round the sum up to a multiple of 2048.					________
Add 10084.				+	10084
The result is the SYSDEF work area for switched SNA major nodes.					________

Figure C-6 (Part 6 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 6A. Calculation of the RDT Size for Application Program Major Nodes

This calculation must be made for every application program major node to be activated in the ACF/VTAM network. The input for these calculations is obtained from the APPL definition deck.

Enter the total number of the following definition statements and multiply by the indicated values:

VBUILD ____________ X 156 = ____________
APPL ____________ X 228 = ____________

Add these two results to obtain the RDT size for the application program major nodes. RDT size = ____________

Work Sheet 6B. Calculation of the SYSDEF Work Area for Application Program Major Nodes

Enter the total RDT size for all application program major nodes (calculated on Work Sheet 6A; rounded up to a multiple of 2048). ____________

Add 11180. + 11180

The result is the SYSDEF work area for application program major nodes. ____________

Figure C-6 (Part 7 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 7A. Calculation of RDT Size for Local Non-SNA Major Nodes

This calculation must be made for every local non-SNA major node to be activated in the ACF/VTAM network. The input for these calculations is obtained from the LOCAL definition deck.

Enter the total number of the following definition statements and multiply by the indicated values:

LBUILD	________	X	156	=	________
LOCAL	________	X	360	=	________
Add these two results to get the RDT size.		RDT size		=	________

Work Sheet 7B. Calculation of the SYSDEF Work Area for Local Non-SNA Major Nodes

Enter the RDT size for the local non-SNA major nodes (calculated on Work Sheet 7A); rounded up to a multiple of 2048. ________

Add 11180 + 11180

The result is the SYSDEF work area for local non-SNA major nodes. ________

Figure C-6 (Part 8 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 8A. Calculation of the RDT Size for Local SNA Major Nodes

This calculation must be made for every local SNA major node to be activated in the ACF/VTAM network. The input for these calculations is obtained from the local SNA definition deck.

Enter the total number of the following definition statements and multiply by the indicated values:

VBUILD	________	X	164	=	________
PU	________	X	276	=	________
LU	________	X	100	=	________
Add the resulting products to get the RDT size.		RDT size		=	________

Work Sheet 8B. Calculation of the SYSDEF Work Area for All Local SNA Major Nodes

Enter the RDT size for the local SNA major nodes (calculated on Work Sheet 8A); rounded up to a multiple of 2048.		________
Add 11180.	+	11180
The result is the SYSDEF work area for all local SNA major nodes.		________

Figure C-6 (Part 9 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 9A. Calculation of RDT Size for All CDRM Major Nodes

This calculation must be made for every major node that can be active in the ACF/VTAM network. The input for this calculation is obtained from the CDRM definition deck.

Enter the total number of the following definition statements and multiply by the indicated values:

VBUILD	________	X	156	=	________
CDRM	________	X	212	=	________

Find the sum of the products. The result is the RDT size for the CDRM major nodes. = ________

Work Sheet 9B. Calculation of SYSDEF Work Area for All CDRM Major Nodes

Enter the RDT size for CDRM major nodes, (calculated on Work Sheet 9A); rounded up to a multiple of 2048.

____

Add 11180. + 11180

The result is SYSDEF work area for all CDRM major nodes. ________

Figure C-6 (Part 10 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 10A. Calculation of RDT Size for All CDRSC Major Nodes

This calculation must be made for every CDRSC major node that can be active in the ACF/VTAM network. The input for this calculation is obtained from the CDRSC definition deck.

Enter the total number of the following definition statements and multiply by the indicated values:

VBUILD ____________ X 156 = ____________

CDRSC ____________ X 212 = ____________

Find the sum of the products.

The result is the RDT size for all CDRSC major nodes. ____________

Work Sheet 10B. Calculation of SYSDEF Work Area for CDRSC Major Nodes

Enter the RDT size for CDRSC major nodes (calculated on Work Sheet 10A); rounded up to a multiple of 2048.

____

Add 11180. + 11180

The result is the SYSDEF work area for CDRSC major node. ____________

Figure C-6 (Part 11 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 11. Calculation of SYSALOAD

This is the area where SYSDEF loads certain tables when building RDTs. Take the input for the variables in the calculations from all NCP generation decks, and all LOCAL and APPL definition decks for the ACF/VTAM system.

Enter the total number of names specified in the name fields of the following macro instructions:

HOST ________
PATH ________
SYSCNTRL ________
PCCU ________
GROUP ________
LINE ________
TERMINAL ________
COMP ________
CLUSTER ________
PU ________
PU (INNODE) ________
DIALSET ________
LU ________
LUPOOL ________
TERMINAL followed by a COMP ________
TERMINAL with CTERM = YES ________

Add these numbers to get SUM A. SUM A = ________

Multiply SUM A by 12 and add 56 to get SUM B.

________ X 12 = ________

+ 56

Enter SUM B (rounded up to a multiple of eight). SUM B = ________

Enter the sum of the sizes of all of the user-specified interpret tables. (Tables referenced more than once are counted only once.) Find the sum by rounding each interpret table size up to a multiple of eight and then adding up the rounded sizes. SUM C = ________

Enter the sum of the sizes of all unique user-specified MODETABs as described for interpret tables. SUM D = ________

Enter the sum of the sîzes of all unique user-specified USSTABs as described for interpret tables. SUM E = ________

Multiply the number of BHSET macros in NCP generation decks (rounded up to a multiple of eight) by nine to get SUM F. ________ X 9 = SUM F = ________

SYSALOAD value = SUM B + SUM C + SUM D + SUM E + SUM F = ________

SYSALOAD value = ________

Figure C-6 (Part 12 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 12. Calculation of IOAPPN

This area is for the ACF/VTAM I/O appendage.

If there is one or more active 3704s or 3705s or local SNA devices in your network enter 2048; otherwise enter zero. __________

If there is one or more active local non-SNA devices in your system enter 8192; otherwise enter zero. __________

If ACF/NCP/VS is used enter 2048; if NCP/VS is used, enter zero. __________

IOAPPN is the sum of these values. IOAPPN = __________

Figure C-6 (Part 13 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 13. ACF/VTAM Fixed SQA Requirements

Enter the values listed below and multiply as indicated.

Number of concurrently active NCPs ________ X 368 = ________

Maximum number of concurrently active local SNA controllers ________ X 368 = ________

Maximum number of concurrently active local non-SNA terminals ________ X 336 = ________

Enter the value of the MAXSUBA parameter from the NCP generation deck ________ X 12 = ________

For each NCP major node, enter the total number of names specified in the name fields of the following macro instructions in the NCP definition deck:

LINE ________
CLUSTER ________
TERMINAL ________
VTERM ________
COMP ________
PU for remote NCP (INNODE) ________
PU ________
LU ________
LUPOOL value ________
+ 2

Find the sum. SUM A = ________

Multiply SUM A by 8 to get the total 3704/3705 SNT size.

SUM A X 8 = ________

If the total size of all 3704/3705 SNTs is equal to or less then 2048, enter the size here. Otherwise, enter 2048 here and the balance (total size - 2048) on Work Sheet 14. ________

For each local SNA major node, enter the total number of names specified in the name fields of the following macro instructions and multiply as indicated:

PU ________ X 2 = ________
LU ________ X 1 = ________
+ 2

Local SNA major node SNTs X 8 = ________

If the total size of all local SNA SNTs is less than or equal to 2048, enter the size here. Otherwise, enter 2048 here and the balance (total size - 2048) on Work Sheet 14. ________

For each local non-SNA major node, enter the number of names specified in the name field of the LOCAL macro instructions:

LOCAL ________
+ 2
________ X 8 = ________

If the total size of all local non-SNTs is less than or equal to 2048, enter the size here. Otherwise, enter 2048 here and the balance (total size - 2048) on Work Sheet 14. ________

For the host SNT, enter the MAXAPPL start option value to be used when ACF/VTAM is started. ________
+ 2
SUM A - ________ X 8 = ________

If the size of the host SNT is less than or equal to 2048, enter the size here. Otherwise, enter 2048 here and the balance (size - 2048) on Work Sheet 14. ________

Constant + 1328

= ________

Find the sum of SNT values above to find the total ACF/VTAM fixed SQA space.

Total ACF/VTAM Fixed SQA Space = ________

Figure C-6 (Part 14 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 14. Calculation of Pageable Supervisor

ACF/VTAM RAM and RSCV Modules

EPTDVT constant. 8192

Enter size of UECBUF baseno (from Figure C-1). ________ X 120 = ________

Round up the product to a multiple of 2048 and enter at far right. ________

Enter the maximum number of concurrent OPNDST, CLSDST, OPEN ACB, CHANGE, INQUIRE, CLOSE ACB, SIMLOGON, SETLOGON, and INTRPRET requests. ________

If NETSOL = YES, enter 1; otherwise, enter zero. ________

Enter the maximum number of concurrently open application programs. ________

Add 3. + 3

Multiply the sum by 32. ________ X 32 = ________

Enter the number of concurrent Init Self and SIMLOGON commands that may exist before an OPNDST macro instruction is issued for the device. ________ X (64+user data) ________

Enter the maximum number of NIB list elements of the concurrent OPNDST requests and multiply by 16. ________ X 16 = ________

Constant 4512

____

Enter the total number of names specified in the name fields of the following NCP generation macro instructions and ACF/VTAM definition statements and find their sum:

PCCU ________
DIALSET ________
GROUP ________
LINE ________
TERMINAL ________
VTERM ________
COMP ________
INNODE ________
LU ________
CLUSTER (non-dial) ________
PU (non-dial) ________
PU (switched SNA) ________
BHSET ________
LOCAL statements ________
APPL statements ________
CDRM statements ________
CDRSC statements ________
Constant 7

SUM = ________

Multiply SUM by 40 and add 4096 to the product.
Enter result at far right. __________ X 40 = __________ + 4096 = __________

Enter the number of all 3704/3705 SNTs in excess of 2048 from Work Sheet 21. __________

Enter the number of all local SNTs in excess of 2048 from Work Sheet 21. __________

Enter the number of all 3790 local SNTs in excess of 2048 from Work Sheet 21. __________

Enter the number of all host SNTs in excess of 2048 from Work Sheet 21. __________

Add 168. + 168

Find the sum of the values entered above to find the ACF/VTAM SQA space. __________

ACF/VTAM Pageable SQA Space = __________

Figure C-6 (Part 15 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 15. ACF/VTAM Storage Requirements for PQA Space in the User Partition

Pageable PQA

Enter the maximum number of open application programs. __________ X 744 = __________

Enter the maximum number of concurrent OPNDST, CLSDST, SIMLOGON, CHANGE, and SETLOGON requests (see notes). __________ X 872 = __________

Pageable Data Area

If VPACE is not specified in the APPL VBUILD statement, calculate:

((NRECORDSES X 2) + NQDSIMLOG) (UNITSZ + 48) = __________

If VPACE is specified in the APPL VBUILD statement, calculate:

(((2N[1] – 1 X NRECORDSES) + NQDSIMLOG) (UNITSZ + 48) = __________

[1] N = number of elements in a receive chain.

Pageable PQA Space in user partition __________

Fixed PQA in User's Partition

Enter the maximum number of open application programs and multiply by 456.

__________ X 456 = Fixed PQA space in the user partition = __________

Some ACF/VTAM code runs under the application program's TCB; thus some of ACF/VTAM's storage is obtained from the user's application program partition as specified above.

Note: If all destinations (terminals, logical units, etc.) are opened at one time, and remain in uninterrupted sessions, the number of OPNDST, CLSDST, and SIMLOGON requests is one. If several application programs may simultaneously establish and terminate sessions, space must be allocated for several concurrent operations. The exact number needed depends on the characteristics of the application programs and the network.

SETLOGON enables an application program to control the number of logon requests that may be queued for it. If no application program uses this feature, no space should be provided for it. CHANGE allows application programs to change processing options for a particular session after the OPNDST is complete. If no application program uses this feature, no space should be provided for it. You must determine the number of SETLOGON and CHANGE requests likely to occur in your particular installation. In many cases, space for one SETLOGON and one CHANGE request is adequate.

Figure C-6 (Part 16 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 16A. Fixed ACF/VTAM Real Storage Requirements

Fixed ACF/VTAM storage requirement in the supervisor NUCLEUS constant.	6224
Enter the total fixed SQA storage from Work Sheet 13.	
Enter the total fixed IOAPPN from Work Sheet 12.	
ACF/VTAM fixed modules constant.	12288
Enter the size of fixed pools from Work Sheet 2.	
Enter the size of fixed PQA in the user's partition from Work Sheet 15.	
Fixed ACF/VTAM PQA.	4096
Constant.	2048
The sum is the total fixed ACF/VTAM real storage requirement.	
Total Fixed ACF/VTAM Real Storage Requirement	

Work Sheet 16B. ACF/VTAM Real Storage Requirements for Record-Mode Sessions

Total fixed storage requirement (from Work Sheet 16A)	=	
Pageable working set modules	=	26624
Enter the number of concurrent record-mode sessions____ X 5	=	
Constant	=	8196
LPBUF see buffer pool formula—Work Sheet 1	=	
CRPLBUF see buffer pool formula—Work Sheet 1	=	
WPBUF see buffer pool formula—Work Sheet 1	=	
Total Real Storage Requirement[1] for Record-Mode Sessions	=	

[1] This requirement only reflects real storage needed to maintain normal network data flow. During connection, disconnection, activation, termination and error recovery processing, additional real storage will be required by your system. Additional real storage will also be needed for application mainline instructions, control blocks, and tables.

Figure C-6 (Part 17 of 18). OS/VS1 Storage Estimate Work Sheets

Work Sheet 17. Calculation of ACF/VTAM Partition Size

ACF/VTAM modules constant. 878568

Fixed PQA constant requirement. 40960

If the Multisystem Networking Facility is installed, enter 56000;
otherwise enter zero. ________

If the ACF/VTAM internal trace is to be used, enter *size* specified
by user, ______ X 2048 ________

If tuning statistics are to be run, enter 4096; otherwise enter zero. ________

If the ACF/VTAM Encrypt/Decrypt Feature is installed, enter 8192;
otherwise enter zero. ________

If the ACF/VTAM Encrypt/Decrypt feature is installed, enter the number
of concurrently active cryptographic sessions and multiply by
16: ________ X 16 = ________

TOLTWKA = maximum number of concurrent TOLTEP
users ________ X 34186. TOLTWKA = ________

Enter the maximum number of concurrent record-mode sessions
(NRECORDSES). ________ X 80 = = ________

Enter the size of pageable pools from Work Sheet 1. ________

Enter the size of fixed pools from Work Sheet 2. ________

Enter pageable storage from Work Sheet 3. ________

Enter the sum of all RDT sizes calculated on Work Sheets 4A - 10A. ________

Enter the *largest* SYSDEF work area calculated on Work Sheets 4B - 10 B. ________

Enter SYSLOAD from Work Sheet 11. ________

Enter IOAPPN from Work Sheet 12. ________

If NETSOL is run as a subtask under the ACF/VTAM partition,
enter 12048; otherwise enter zero. = ________

The sum of above values is SUM A. SUM A = ________

Divide SUM A by 1024. = ________ K

Round up to the next multiple of 64K to get the
ACF/VTAM partition size. ________

ACF/VTAM partition size = ________ K

Figure C-6 (Part 18 of 18). OS/VS1 Storage Estimate Work Sheets

ACF/VTAM Storage Requirements in OS/VS2 SVS

ACF/VTAM runs in its own region in OS/VS2 SVS. This region contains the ACF/VTAM task and its subtasks, and most of the ACF/VTAM modules and storage pools. Other areas of the OS/VS2 SVS operating system that are used and enlarged by including ACF/VTAM are:

- Nucleus
- Fixed SQA
- ACF/VTAM LSQA
- Link Pack Area
- User Region

Storage worksheets are used to calculate the amount of storage required by ACF/VTAM in the following areas:

- Real Storage (Figure C-7)
- ACF/VTAM region (Figure C-8)
- User region(s) (Figure C-9)
- Total virtual storage (Figure C-10)

Real Storage

Real storage includes ACF/VTAM fixed storage for modules in the nucleus, control blocks in fixed SQA, and modules and control blocks in the ACF/VTAM region. It also includes pageable control blocks and modules in the ACF/VTAM region required to operate a non-paged record-mode session. Additional storage will be needed to operate a network, for example, for starting the network, error recovery processing, and establishing and terminating sessions.

Use Figure C-7 to calculate the total real storage required by your network.

Real Storage

Fixed Storage

Nucleus	=		2440
SQA			
Enter (Maximum number of concurrent OPNDSTs or CLSDSTs *or* the maximum number of PUs and remote BSC cluster controllers activated or deactivated concurrently) ________X 32	=	________	
Enter the number of concurrent active sessions ________ X 5	=	________	
Enter the number of concurrent Init Self and SIMLOGON commands that may exist for a device before an OPNDST macro instruction is issued. ________ X (64 + user data)		________	
Enter the total number of name fields specified in the NCP and ACF/VTAM definition macro instructions ________X 40	=	________	
Enter the number of open ACBs ________X 32	=	________	
Add constant	=	32768	
Enter the total number of network addressable units (nodes) ________X 8	=	________	
Enter the number of concurrently active NCPs ________ X 392	=	________	
Enter the number of local non-SNA terminals ________ X 360	=	________	
Enter the MAXSUBA parameter + 1 ________ X 12	=	________	
Add	=	4096	
Calculate total SQA	=		________
Add control blocks	=		6280
Add the LSQA for the ACF/VTAM region			40960
Enter the LSQA for the application program regions			________
Storage Pools			
IOBUF (see buffer pool work sheet Figure C-12)	=		________
SFBUF (see buffer pool work sheet Figure C-12)	=		________
LFBUF (see buffer pool work sheet Figure C-12)	=		________
Modules (includes packaging)			
If there are one or more 370Xs or 3790s in your system, enter 8192, otherwise enter zero	=		________
If there are one or more local 3270s in your system, enter 8192, otherwise enter zero	=		________
If there are one or more NCP5s in your system, enter 4098, otherwise enter zero	=		________
If there are one or more 3790s in your system, enter 4098, otherwise enter zero	=		________
Constant	=		16384
Calculate the Total Fixed Storage Size	=		________

Pageable Storage

LPBUF (see Figure C-11)	=	________	
CRPLBUF (see Figure C-11)	=	________	
WPBUF (see Figure C-11)	=	________	
Modules (includes packaging)	=	28672	
Control blocks	=	+8096	
Total Pageable Storage	=		________
Calculate the Total Real Storage Size (Total Fixed Storage + Total Pageable Storage)			________

Figure C-7. Calculating the Total Real Storage Size in OS/VS2 SVS

ACF/VTAM Region Storage

Modules

If there are one or more 370Xs or local 3790s in your system, enter 5280; otherwise enter zero = __________

If there are one or more local 3270s in your system, enter 8160; otherwise enter zero = __________

If there are one or more NCP5s in your system, enter 780; otherwise enter zero = __________

If there are one or more local 3790s in your system, enter 1260; otherwise enter zero = __________

If you use TOLTEP in your system, enter 34816, otherwise enter zero = __________

If NETSOL = YES, enter 12408; otherwise enter zero = __________

If you have the ACF/VTAM Multisystem Networking Facility in your system, enter 71000, otherwise enter zero = __________

Add constant = 1321000

Control Blocks

Calculate storage pool sizes (see Figure C-11 and Figure C-12)

APBUF = __________
CRPLBUF = __________
IOBUF = __________
LFBUF = __________
LPBUF = __________
NPBUF = __________
PPBUF = __________
SFBUF = __________
SPBUF = __________
WPBUF = __________
UECBUF = __________

Enter the total number of PCCU, DIALSET, GROUP, LINE, CLUSTER, APPL, PU, LU, LUPOOL, LBUILD and VBUILD macro instructions (except those associated with CDRM or CDRSC macro instructions) __________ X 212 = __________

Enter the total number of LOCAL macro instructions __________ X 412 = __________

Enter the total number of COMP, TERMINAL, and VTERM macro instructions __________ X 272 = __________

Enter the total number of CDRM, CDRSC, and associated VBUILD macro instructions __________ X 272 = __________

Figure C-8 (Part 1 of 2). Calculating the ACF/VTAM Region Size in OS/VS2 SVS

There are seven possible major nodes (NCP, switched SNA, application, local non-SNA, local SNA, CDRM, and CDRSC). Count the number of macro instructions under each major node. Choose the major node with the maximum number of macro instructions associated with it. If the node chosen was an NCP, application, switched SNA, or local SNA major node; multiply the number of macro instructions by 360. If the node chosen was a local non-SNA major node, multiply the number of macro instructions by 360. If the node chosen was a CDRM or CDRSC major node, multiply the number of macro instructions by 212. Take this product and add 11180. Total = __________

Enter the number of network addressable units (nodes) __________ X 221 = __________

Enter the total length (in bytes) of all user-specified interpret tables (MODTAB and USSTAB), authorization routines, and accounting routines = __________

Add constant 4232

If the network node activation is staged, enter the maximum number of nodes concurrently activated. If the network node activation is not staged, enter the total number of network nodes __________ X 3320 = __________

Enter the maximum number of concurrent record-mode sessions __________ X 96 = __________

If you use the ACF/VTAM internal trace, enter the number of user-specified pages __________ X 4096 = __________

If you use the ACF/VTAM tuning statistics, enter 4096; otherwise enter zero = __________

Calculate the total number of bytes needed in the ACF/VTAM region = __________

Round up total bytes to the next multiple of 64K to determine the ACF/VTAM region size. The ACF/VTAM region size is __________

Figure C-8 (Part 2 of 2). Calculating the ACF/VTAM Region Size in OS/VS2 SVS

User Region Storage

Estimate the amount of storage needed for the record-mode pageable input data buffer area.
The following factors should be considered:

All input data without an outstanding RECEIVE. = ____________
Segmented data that may be buffered until the full message is received. = ____________
BSC message blocks that may be buffered until the full message is received. = ____________
Full-screen input data that may be buffered for each TSO user. = ____________

Enter the maximum number of concurrent OPNDSTs or CLSDST macro instructions __________ X 872 = ____________
Enter the maximum number of open ACBs __________ X 744 = ____________
Enter the total size (in bytes) of all user application programs and tables. = ____________
Calculate the total size of the user region. = ____________

Figure C-9. Calculating the ACF/VTAM User Region Size in OS/VS2 SVS

Total Virtual Storage

Nucleus = 2440
Enter total SQA (see Figure C-7) = ____________
Enter ACF/VTAM region size (see Figure C-8) = ____________
Enter ACF/VTAM region's LSQA = 40960
Enter the user's region size (see Figure C-9) = ____________
Enter the application program's LSQA (see Figure C-7) = ____________
LPA = +8720
Total Virtual Storage Size = ____________

Figure C-10. Calculating the Total Size of Virtual Storage for ACF/VTAM in OS/VS2 SVS

Paged Buffer Pools

For each pool below, get the *baseno* value for that pool from Figure C-1 and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
APBUF	72	X		=	
CRPLBUF	128	X		=	
LPBUF	1024	X		=	
NPBUF	208	X		=	
PPBUF	(UNITSZ+48)	X		=	
SPBUF	112	X		=	
WPBUF	176	X		=	
UECBUF	120	X		=	
The sum of the products is SUM A.			SUM A	=	

The PAGEPOOLS value is SUM A rounded up to a multiple of 2048.

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-11. Calculating the Storage Needed for Paged Buffer Pools in OS/VS2 SVS

Fixed Buffer Pools

For each pool below, get the *baseno* value for that pool from Figure C-1, and insert it in the appropriate equation below:

	multiplier		*baseno*[1]		*product*
SFBUF	80	X		=	
LFBUF	128	X		=	
IOBUF	(UNITSZ+31)	X		=	
					+2048
The sum of the products is SUM A.			SUM A	=	

Round SUM A up to a multiple of 2048.

[1] If a *slowpt* value has been specified, add the *slowpt* value to the *baseno* value before determining the product.

Figure C-12. Calculating the Storage Required for Fixed Buffer Pools in OS/VS2 SVS

Appendix D. TSO/VTAM (OS/VS2 MVS Only)

TSO/VTAM provides the capability of using TSO through ACF/VTAM. TSO, formerly an option of the control program (hence the name time sharing option), is a standard feature in OS/VS2 MVS that provides conversational time sharing. ACF/VTAM is the access method that directs the transmission of data between ACF/VTAM application programs and terminals.

TSO/VTAM supports the following terminals:

IBM 3270 Information Display System

IBM 3767 Communication Terminal

IBM 3770 Data Communication System

The basic elements of TSO/VTAM are the terminal control address space (TCAS) and the ACF/VTAM terminal I/O coordinator (VTIOC). TCAS accepts logons from TSO/VTAM users and creates an address space for each user. VTIOC is the interface between TSO and ACF/VTAM; it coordinates data flow. TCAS and VTIOC, together with existing TSO components such as the LOGON scheduler, the command processors, the terminal monitor program, and the TSO service routines, make up a TSO/VTAM time-sharing system.

TSO/VTAM uses the ACF/VTAM application program interface (API). TSO/VTAM may co-reside with TSO/TCAM in the same host processor.

Planning Information

Programming Prerequisites

TSO/VTAM requires the following programs:

OS/VS2 Release 3.7

NCP/VS Version 5 (within a domain) or ACF/NCP/VS (within a domain or for cross-domain communication)

ACF/VTAM (within a domain)

ACF/VTAM Multisystem Networking Facility (within a domain or for cross-domain communication)

Programming Considerations

This section provides information of interest to system programmers.

ACF/VTAM Considerations

Refer to Chapter 2, "Defining the Network" for information about the definition and tailoring of an ACF/VTAM system. Part of the ACF/VTAM definition process involves the definition of ACF/VTAM application programs, a logon mode table, and an interpret table.

Defining TCAS Program Properties

TCAS must have an entry in the program properties table (PPT), load module IEFSD060 (CSECT IEFSDPPT). This makes certain designations as to execution attributes of TCAS:

The program cannot be canceled.

A unique protection key is assigned to the program.

The program is privileged and is not swapped unless the address space is in a long wait.

The program is a system task and is not timed.

There is no host processor affinity.

The contents of the entry are an 8-byte EBCDIC field, IKTCAS00, followed by a 4-byte hexadecimal field, X'D860FFFF'.

After TSO/VTAM is installed, the PPT should be checked. This check is necessary because the complete PPT can be replaced during TSO/VTAM installation. The new PPT contains all current system entries. However, private user entries must be inserted into the new table.

Defining TCAS and Each User to ACF/VTAM

An APPL definition statement defines an application program to ACF/VTAM. Because TCAS and each TSO user are ACF/VTAM application programs, APPL definition statements must be specified for them and put into SYS1.VTAMLST. Code the following APPL statement for TCAS:

```
TSO       APPL   PRTCT=password,BUFFACT=n,AUTH=(NOACQ,NOBLOCK,  C
                 PASS,NVPACE,TSO,NOPO)
```

Code as many APPL statements, in the following format, as there will be users logged on to TSO/VTAM at one time:

```
TSOnnnn   APPL   PRTCT=password,BUFFACT=n,AUTH=(NOACQ,NOBLOCK,  C
                 PASS,NVPACE,TSO,NOPO)
```

The same *password* must be specified for TCAS and each TSO/VTAM user. A different application program name, in the form TSO*nnnn*, must be specified for each terminal. *nnnn* is a decimal number; the numbering must start with 0001 and must be sequential. Note that NOACQ, NOBLOCK, and NOPO need not be coded; they are default values.

The BUFFACT parameter of this macro is a decimal number between 1 and 255 used as a multiplicand in determining the number of buffers that the application program has at its disposal. The multiplier is the BUFLIM parameter on the LU macro. The size of these buffers is determined during ACF/VTAM initialization. Therefore, the user must be careful when defining the number of buffers so that the maximum expected amount of data can be handled.

In a multiple-domain network, you must have special definition statements with unique names for TCAS and the terminals in your domain and in interacting domains. Code the following APPL statement for TCAS in your domain:

```
thisTSO     APPL   ACBNAME=TSO,PRTCT=password,BUFFACT=n,        C
                   AUTH=(NOACQ,NOBLOCK,PASS,                    C
                   NOVPACE,TSO,NOPO)
```

Code as many APPL statements in the following format as the maximum number of terminals that will be logged on to TSO/VTAM at one time:

```
thisnamen     APPL    ACBNAME=TSO00n,PRTCT=password,BUFFACT=n,      C
                      AUTH=(NOACQ,NOBLOCK,PASS,NVPACE,              C
                      TSO,NOPO)
```

this is a unique prefix that must be included on all APPL statements for your domain. The same prefix must be specified in *thisnamen* for each terminal. The *n* suffix is a decimal number that must start with 1 and be sequential. The remaining operands are described above.

Code the following statement for each TCAS in another domain with which your domain will communicate:

```
thatTSO    CDRSC    CDRM=that
```

Code the following CDRSC statement for each resource in another domain that can communicate with your domain:

```
thisnamen  CDRSC    CDRM=that
```

that specifies the owning CDRM for each domain and must be included on each TCAS and CDRSC definition statement. The *n* suffix is a decimal number that must start with 1 and be sequential.

Defining a Logon Mode Table to Specify 3270 Screen Sizes

If your installation has IBM 3270 system network architecture (SNA) terminals (that is, 3270s attached by SDLC links) of more than one screen size, you must define a logon mode table and use the PSERVIC operand of the MODEENT macro instruction to specify one of the screen sizes. The PSERVIC operand specifies presentation services information.

Code the following MODEENT macro instruction for a 3270 terminal having a screen size of 480 characters:

```
MODEENT      FMPROF=X‘02’,TSPROF=X‘02’,PRIPROT=X‘71’,        C
             SECPROT=X‘40’,COMPROT=X‘2000’,                  C
             PSERVIC=X‘000000000000000000000100’
```

Code the following MODEENT macro instruction for a 3270 terminal having a screen size of 1920 characters:

```
MODEENT      FMPROF=X‘02’,TSPROF=X‘02’,PRIPROT=X‘71’,        C
             SECPROT=X‘40’,COMPROT=X‘2000’,                  C
             PSERVIC=X‘000000000000000000000200’
```

Note that the FMPROF, TSPROF, PRIPROT, SECPROT, and COMPROT values are the same as those used in IBM-supplied logon mode table ISTINCLM. The PSERVIC value is 12 bytes of hexadecimal digits.

Using the PSERVIC operand to define the less common screen size, and the SCRSIZE parameter of SYS1.PARMLIB member TSOKEY00 to define the more common screen size, requires less coding by the system programmer.

If your installation has LU2 terminals attached by SDLC links, you must define a logon mode table and use the RUSIZES and PSERVIC operands of the MODEENT macro instruction. RUSIZES specifies the primary and secondary RU sizes, and PSERVIC specifies LU type and screen size. In the following examples, RUSIZES indicates a 256-byte maximum secondary logical unit RU send size, and a 1024-byte maximum primary logical unit RU send size. Use the PSERVIC operand that reflects the type of LU2 terminal being defined.

```
MODEENT       FMPROF=X‘03’,TSPROF=X‘03’,PRIPROT=X‘B1’,              C
              SECPROT=X‘90’,COMPROT=X‘3080’,RUSIZES=X‘8587’,        C
```

For a 480-character screen size with no large-screen feature, code:

```
PSERVIC=X‘020000000000000000000100’
```

For a 1920-character screen size with no large-screen feature, code:

```
PSERVIC=X‘020000000000000000000200’
```

To switch between 480-character and 960-character screens, code:

```
PSERVIC=X‘0200000000000C280C507F00’
```

To switch between 1920-character and 2560-character screens, code:

```
PSERVIC=X‘020000000000185020507F00’
```

To switch between 1920-character and 3440-character screens, code:

```
PSERVIC=X‘02000000000018502B507F00’
```

To prevent switching of screen sizes on a 480-character screen terminal with alternate 960-character capability, code:

```
PSERVIC=X‘0200000000000C2800007E00’
```

Defining an Interpret Table for Compatible Logons

Because TSO/VTAM uses ACF/VTAM’s logon facilities, the TSO LOGON command is not supported. This fact can be made transparent to terminal users if the system programmer defines an interpret table to allow logon requests to have the same format as that used by the TSO LOGON command. The following interpret table definition may be used:

```
INTBL      INTAB
           LOGCHAR APPLID=(APPLICID,TSO),SEQNCE=‘LOGON’
           LOGCHAR APPLID=(APPLICID,TSO),SEQNCE=‘logon’
           ENDINTAB
           END
```

This interpret table allows a user to enter a logon command in uppercase or lowercase letters. The definition requires the characters *logon* to be coded in lowercase letters. This can be done by multipunching with a card punch.

TSO Considerations

Information about the TSO-related steps that must be performed before a TSO/VTAM terminal user logs on can be found in *OS/VS2 System Programming Library: TSO,* GC28-0629, *OS/VS2 System Programming Library: Initialization and Tuning Guide,* GC28-0681, and *ACF/VTAM Logic: VTIOC and TCAS,* LY27-8028. (Make sure you have updated the first two manuals with TSO/VTAM SU supplements before referring to them.) The following list outlines the steps and refers the reader to the appropriate publication:

Write LOGON cataloged procedures and include them in SYS1.PROCLIB. Refer to *OS/VS2 System Programming Library: TSO.*

Construct the TSOKEY00 member of SYS1.PARMLIB (or an alternate member) to set VTIOC parameters. Refer to *OS/VS2 System Programming Library: Initialization and Tuning Guide.*

Include SYS1.CMDLIB in a LNKLSTxx member of SYS1.PARMLIB or in a LOGON cataloged procedure. Refer to *OS/VS2 System Programming Library: Initialization and Tuning Guide.*

Create or convert the user attribute data set (UADS) and the broadcast data set. Refer to *OS/VS2 System Programming Library: TSO.*

Build translation tables if character translation is desired. Refer to *OS/VS2 System Programming Library: TSO.*

Write the procedure that starts TSO/VTAM time sharing. Refer to *OS/VS2 System Programming Library: TSO.*

Write any command exit routines you plan to use. Refer to *OS/VS2 System Programming Library: TSO.*

Write any editing, attention handling, and nonsupported terminal exit routines you plan to use. Refer to *ACF/VTAM Logic: VTIOC and TCAS,* LY27-8028.

Performance Considerations

The following list contains suggestions for improving TSO/VTAM performance:

Terminal users should stack input data whenever possible. This means using the multiple-line input technique on 3270 terminals and buffered SDLC transmission on 3767 and 3770 terminals. Stacking input results in a decreased number of data transmissions to TSO. Users of 3270 terminals can stack up to a full screen of data; 3767 and 3770 terminal users can stack from 256 bytes to 2,000 bytes of data, depending on the terminal's buffer size.

Users of 3767 and 3770 terminals can reduce idle time by "typing ahead." This means that whenever the terminal is not receiving data or is transmitting data, the user can enter input.

Using automatic line numbering on 3270 terminals results in a shorter TSO/VTAM processing path when handling input; this reduces system overhead and expedites unlocking the keyboard.

Use BREAK mode with 3276 and 3278 terminals to expedite unlocking the keyboard. BREAK mode reduces idle time by allowing typing ahead.

3270 Large Screen Considerations

The following information describes screen management techniques for IBM 3270 terminals that have alternate screen sizes specified in the LOGMODE entries used for them at logon. In both cases, the erase-write (EW) command is used to reset the terminal to the default size and the erase-write-alternate (EWA) command is used to switch the terminal to the alternate (larger) screen size.

TSO/VTAM Screen Management

TSO/VTAM can manage the screen for TSO command processors that use line-oriented I/O. In this case, TSO/VTAM uses the alternate (larger) screen size whenever possible. At logon, the controller initializes the terminal to the default screen size. TSO/VTAM writes to the terminal using the default value specified in the LOGMODE entry until the first screen erasure is needed, because of screen overflow or use of the CLEAR key.

At that time, an erase-write-alternate (EWA) command is issued to switch the screen to the alternate size. TSO/VTAM continues to write to the alternate size when it is managing the screen within a session. If the CLEAR key is pressed at any time, the controller switches the terminal to the default size. When TSO/VTAM manages the screen, it issues an EWA to reset the terminal to the alternate screen size.

Full-Screen Command Processor Screen Management

An IBM-supplied or user-supplied command processor that uses full-screen I/O in TSO full-screen mode can be used to manage the screen (see *OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor,* GC28-0648). TSO/VTAM uses the screen size used by the application program. If the CLEAR key is pressed, TSO/VTAM erases the screen with either an EW or EWA command depending on which was last used by the command processor.

How TSO/VTAM Differs from TSO through TCAM

The following list describes differences between TSO/VTAM and TSO through TCAM. It is intended for system programmers who are familiar with TSO through TCAM. The differences are:

TSO/VTAM does not use the TCAM message control program (MCP).

VTIOC replaces TIOC (terminal I/O coordinator) as the means for controlling data movement between TSO and the access method.

TSOKEY00 replaces IKJPRM00 as the SYS1.PARMLIB member that contains time-sharing parameters.

The system START command starts TSO/VTAM time sharing. The MODIFY command controls the number of users that may be logged on and terminates user address space. The STOP command stops TSO/VTAM time sharing.

TSO/VTAM user address spaces are totally independent of each other. Each address space is an individual TSO/VTAM application program, with ownership of its terminal. A noncontiguous terminal status block (TSB) is allocated dynamically when each user logs on. Users do not share buffers; VTIOC allocates buffers as needed.

TSO/VTAM uses ACF/VTAM's logon facilities. The existing TSO LOGON command, which provides a logon to the TSO terminal monitor program (TMP), is supported only for re-logons. The existing TSO LOGOFF command is supported only if HOLD is not specified.

TSO/VTAM support for the PROFILE and TERMINAL commands differs from that provided through TCAM. Specification of character-delete and line-delete control characters (PROFILE command) is not supported, due to the editing capabilities of TSO/VTAM-supported terminals. Definition of attention interruption characteristics (TERMINAL command), including the simulated attention interruption, is not supported, because attention interruptions are always accepted on TSO/VTAM-supported terminals. Character translation (TERMINAL command) is provided for TSO/VTAM-supported terminals.

There are five terminal control macros: STTRAN sets character translation. STFSMODE turns 3270 full-screen mode on or off. STLINENO specifies the number of a 3270 screen line on which the next non-full-screen message should appear and optionally turns full-screen mode on or off. STTMPMD indicates whether the terminal control routine is active for the terminal and whether the ATTN and CLEAR keys are to be passed to the application program as data. GTTERM obtains the primary or alternate screen sizes and places them in a user-provided area. Macros STATTN and STCC are not supported by TSO/VTAM.

No data is lost when a user re-logs on (reconnects) after his line was disconnected.

TSO/VTAM takes advantage of multiprocessing because it is multitask structured; TCAM executes in one host processor at a time.

Publication Support

Only the publications that are new or have changed for TSO/VTAM are identified here. The names of all the publications that apply to systems that have TSO and ACF/VTAM are not included.

Publication	Base Order Number	SU Supplement Order Number
Reference		
OS/VS2 TSO Terminal User's Guide	GC28-0645	GD23-0044
OS/VS2 TSO Command Language Reference	GC28-0646	SD23-0064*
OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor	GC28-0648	SD23-0065*
OS/VS2 System Programming Library: TSO	GC28-0629	GD23-0047
OS/VS2 System Programming Library: Debugging Handbook (Volume 1)	GC28-0708	GD23-0048
OS/VS2 System Programming Library: Debugging Handbook (Volume 2)	GC28-0709	GD23-0049
OS/VS2 System Programming Library: Initialization and Tuning Guide	GC28-0681	GD23-0050
OS/VS2 System Programming Library: Supervisor	GC28-0628	None
OS/VS2 System Programming Library: System Generation	GC26-3792	None
Operator's Library: OS/VS2 MVS System Commands	GC38-0229	GD23-0051
OS/VS Message Library: VS2 System Codes	GC38-1008	GD23-0052
OS/MVT and OS/VS2 TSO Terminals	GC28-6762	GD21-0004*
Logic		
ACF/VTAM Logic: VTIOC and TCAS	LY27-8028	None
OS/VS2 TSO Terminal Monitor Program and Services Routines Logic	SY28-0650	SD23-0053
OS/VS2 TSO Command Processor Logic (Volume IV)	SY28-0652	SD23-0054
OS/VS2 TSO Terminal Messages Directory	SY28-0654	SD23-0055
OS/VS2 System Logic Library (Volume 1)	SY28-0713	None
OS/VS2 System Logic Library (Volume 2)	SY28-0714	None
OS/VS2 System Logic Library (Volume 4)	SY28-0716	None
OS/VS2 System Logic Library (Volume 6)	SY28-0718	None
OS/VS2 System Logic Library (Volume 7)	SY28-0719	None

Each SU supplement has a cover letter that indicates which pages of the manual are affected. The pages of the SU supplements marked with an * above have the ACF/VTAM program number (5735-RC2) on them. These SU supplements apply to TSO/VTAM in ACF/VTAM. The other SU supplements apply to both TSO/VTAM Level 2 and TSO/VTAM in ACF/VTAM; the pages of these supplements have the TSO/VTAM Level 2 ID (5752-858) on them. Inserting the pages of an SU supplement into a manual may be different from inserting technical newsletters. Be sure to read the instructions on the cover letter before inserting or removing any pages.

Appendix E. ACF/VTAM Buffer Pool Control Blocks

This appendix contains information that can be of assistance when allocating storage for ACF/VTAM buffer pools. Figures E-1 through E-3 list the buffer pools and their associated ACF/VTAM control blocks for each OS/VS operating system.

OS/VS2 MVS

Buffer Pool[1]	Associated Control Blocks	Subpool	Fetch-Protected Storage	Fixed Storage
APBUF	ACE, ICE, CSP, FSB, FDB, LSCB, LCPB, PLCPB	231	Yes	Optional[2]
SFBUF	LUCB, SRB, MPLT, MQL	231	Yes	Yes
SPBUF	DCLCP, NMLPB, CKPAB, TSCB	231	Yes	Optional[2]
LFBUF	LULB	231	Yes	Yes
LPBUF	CRA, NCSPL, DCCRR	231	Yes	Optional[2]
UECBUF	UECB, RPL	241	No	Optional[2]
WPBUF	FMCB, APCRR, CSCB	231	Yes	Optional[2]
NPBUF	FMCBB, TCCW	231	Yes	Optional[2]
CRPLBUF	CRPL, RPH, DNCB, HCNCB	231	Yes	Optional[2]
PPBUF	Input Data (basic-mode only)	231[3]	Yes	Optional[2]
IOBUF	Input/Output data	231	Yes	Yes

[1] *All buffer pools are located in CSA.*
[2] *User may specify fixed storage by using the [,F] start option.*
[3] *Input data area for basic-mode sessions only. PPBUF for record-mode sessions is located in the user's region.*

Figure E-1. ACF/VTAM Buffer Pool Control Blocks for OS/VS2 MVS

OS/VS1

Buffer Pool	Associated Control Blocks	Subpool	Fetch-Protected Storage	Fixed Storage
APBUF	ACE, ICE, CSP, FSB, FDB, LCPB, LSCB, TCCW	0	Yes	Optional[1]
SFBUF	LUCB	0	Yes	Yes
SPBUF	DCLCP, NMLPB, CKPAB, TSCB	0	Yes	Optional[1]
LFBUF	PST, LULB	0	Yes	Yes
LPBUF	DCCRR., CRA, NCSPL, APCRR	0	Yes	Optional[1]
UECBUF	UECB, RPL	241	No	Optional[1]
WPBUF	CSCB, FMCB	0	Yes	Optional[1]
NPBUF	FMCBB	0	Yes	Optional[1]
CRPLBUF	RPH, CRPL, DNCB, HCNCB	0	Yes	Optional[1]
PPBUF	Input data (basic-mode only)	0[2]	Yes	Optional[1]
IOBUF	Input/Output data	0	Yes	Yes

[1] *User may specify fixed storage by using the [,F] start option.*

[2] *Input data area for basic-mode sessions only. PPBUF for record-mode sessions is located in the user's region.*

Figure E-2. ACF/VTAM Buffer Pool Control Blocks for OS/VS1

OS/VS2 SVS

Buffer Pool	Associated Control Blocks	Subpool	Fetch-Protected Storage	Fixed Storage
APBUF	ACE, ICE, CSP, FDB, FSB, LCPB, LSCB, PLCPB	250	Yes	Optional[1]
SFBUF	LUCB	250	Yes	Yes
SPBUF	DCLCP, NMLPB, CKPAB, TSCB	250	Yes	Optional[1]
LFBUF	PST, LULB	250	Yes	Yes
UECBUF	UECB, RPL	252	No	Optional[1]
WPBUF	CSCB, FMCB	250	Yes	Optional[1]
NPBUF	FMCBB	250	Yes	Optional[1]
CRPLBUF	RPH, CRPL, DNCB, HCNCB	250	Yes	Optional[1]
PPBUF	Input data (basic mode only)	250[2]	Yes	Optional[1]
IOBUF	Input/Output data	250	Yes	Yes

[1] *User may specify fixed storage by using the [,F] start option.*

[2] *Input data area for basic-mode sessions only. PPBUF for record mode sessions is located in the user's region.*

Figure E-3. ACF/VTAM Buffer Pool Control Blocks for OS/VS2 SVS

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.

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing,* copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. 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 *SC1* 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.

ACB name. (1) The name of an ACB macro instruction. (2) A name that can be specified in the ACBNAME parameter of an APPL statement. This name allows an ACF/VTAM application program that is used in more than one domain to specify the same application program identification (pointed to by the APPLID parameter of the program's ACB statement) in each copy. ACF/VTAM knows the program by both its ACB name and its network name (the name of the APPL statement). Program users within the domain can request logon using the ACB name or the network name; program users in other domains must use the network name (which must be unique in the network). Contrast with *network name.*

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 layer. In SNA, the functional layer of each individual session in which the end user's application program is executed. See also *function management, transmission subsystem.*

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 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 information unit (BIU). In SNA, the unit of data and control information that is passed between connection point managers. It consists of a request/response header (RH) followed by a request/response unit (RU).

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.*

basic transmission unit (BTU). (1) In the network control program, the unit of exchange between the host processor and the communications controller. It consists of control information and may also include data. The control information consists of a basic transmission header (BTH) and a basic device unit (BDU). All data transferred between the host and the communications controller is preceded by a BTU. (2) In SNA, the unit of data and control information passed between path control components. The BTU can consist of one or more path information units (PIUs), depending on whether blocking is done by the path control that builds the BTU.

block. In the basic mode of ACF/VTAM, a unit of data that is transmitted between an ACF/VTAM application program and a terminal.

boundary function. In SNA: (1) A general term used for any one of several capabilities of a host node or a communications controller node: (a) transforming the network address form to a local address form, and vice versa, for attached terminals or cluster controller nodes; (b) performing physical unit services and sequence numbering for attached, low-function terminals within its subarea; and (c) providing pacing of the data flows for secondary LUs within a subarea. (2) The programming component and functional structure that performs the above capabilities.

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.*

checkpoint. A point in time at which the status of a data communication system is recorded. The system can then be reconstructed to its status at or near the time of failure.

CID. Communication identifier.

ciphertext. In ACF/VTAM, synonym for *enciphered data.*

clear data. Data that is not enciphered. Synonymous with *plaintext.*

clear session. A session in which only clear data is transmitted or received. Contrast with *cryptographic session.*

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.

common network. In SNA, the network consisting of path control and data link control elements that routes and moves path information units between any two transmission control elements.

communication control character. *A control character intended to control or facilitate transmission of data over communication networks.

communication control unit. A communication device that controls the transmission of data over lines in a telecommunication network. Communication control units include transmission control units and communications controllers.

communication identifier (CID). In ACF/VTAM, a key for locating the control blocks that represent an active 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.*

connection point manager (CP manager). In SNA, one of the three components of transmission control; it provides a common mechanism by which session control, network control, and network addressable units communicate with their corresponding elements through the common network. The unit of information handled by the connection point manager is a request/response unit (RU). The unit of control information built by the sending connection point manager and interpreted by the receiving connection point manager is a request/response header (RH). See also *session control, network control.*

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 keys. In ACF/VTAM, see *cross-domain keys.*

cross-domain. Pertaining to control or resources involving more than one domain.

cross-domain keys. A pair of *cryptographic keys* used at a host processor to *encipher* the *cryptographic session key* that is sent to another host processor and to *decipher* the *cryptographic session key* that is sent from another host processor during cross-domain cryptographic session establishment.

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.

CRV. Cryptographic verification. See *cryptographic verification request.*

cryptographic. Pertaining to the transformation of data to conceal its meaning.

cryptograhic algorithm. A set of rules that specify the mathematical steps required to *encipher* and *decipher* data.

cryptographic key. A 64-bit value (containing 56 independent bits and 8 parity bits) that functions with the *DES algorithm* in determining the output of the *DES algorithm.* See also *cross-domain keys, cryptographic session key, host master key,* and *secondary logical unit key.*

cryptographic session. A session in which data may be enciphered before it is transmitted and deciphered after it is received. See also *required cryptographic session* and *selective cryptographic session.* Contrast with *clear session.*

cryptographic session key. In ACF/VTAM, a *data encrypting key* used to *encipher* and *decipher* data transmitted in a *cryptographic session.*

cryptographic verification (CRV) request. A request unit used to return an *initial chaining value* to the secondary logical unit to verify that the secondary logical unit is using the same *cryptographic session key.*

D

data communication. The transmission, reception, and validation of data.

data encrypting key. A *cryptographic key* used to *encipher* and *decipher* data transmitted in a *cryptographic session.* See also *cryptogrpahic session key.* Contrast with *key encrypting key.*

Data Encryption Standard (DES) algorithm. A *cryptographic algorithm* designed to *encipher* and *decipher* 8-byte blocks of data using a 64-bit *cryptographic key,* as specified in the *Federal Information Processing Standard Publication 46,* January 15, 1977.

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. (1) (SC1) An assembly of those parts of two data terminal equipments that define the protocol together with their interconnecting data circuit. This assembly enables a data source to transfer data to a data sink. (2) The communication channel, modem, and communication controls of all stations connected to the communication channel, used in the transmission of information between two or more stations. (3) The physical connection and the connection protocols between the host and communication controller nodes via the host data channel. (4) Contrast with *communication line.*

Note: *A communication line is the physical medium; for example, a telephone wire, a microwave beam. A data link includes the physical medium of transmission, the protocol, and associated communication devices and programs—it is both logical and physical.*

data link control (DLC). (1) The noninformation exchanges that set up, control, check, and terminate the information exchange(s) between two stations on a data link. (2) In SNA, one of the constituent parts of the transmission subsystem, and one of two constituent parts of the common network. It initiates, controls, checks, and terminates the data transfer over a data link between two nodes. Two distinct DLCs are defined in SNA: the DLC for the System/370 data channel, and SDLC for serial-by-bit data links. See also *path control* and *transmission control.*

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.

decipher. The process of converting *enciphered data* into *clear data.* Synonymous with *decrypt.* Contrast with *encipher.*

decrypt. Synonym for *decipher.*

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.

DES. Data Encryption Standard.

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.

DLC. Data link control.

domain. In a data communication system, the portion of the total network that is controlled by the SSCP in one telecommunication access method.

dormant. In ACF/VTAM, when processing a VARY NET,INACT,I command, the stage at which all I/O for the *node* has been completed. At this stage, the node is logically disconnected, but it may be physically disconnected as well.

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.*

encipher. The process of converting *clear data* into *enciphered data.* Synonymous with *encrypt.* Contrast with *decipher.*

enciphered data. Data that is intended to be illegible to all except those who legitimately possess the means to reproduce the *clear data.* Synonymous with *ciphertext.*

encrypt. Synonym for *encipher.*

end user. The ultimate source or destination of information flowing through a system. An end user may be an application program, an operator (such as a terminal user or a network operator/administrator), or a data medium (such as cards or tapes).

error lock. In the basic mode of ACF/VTAM, a condition in which communication with a non-SNA terminal is suspended until a reset operation occurs.

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

FID. Format identification. FID0, FID1, FID2, FID3. See *format identification.*

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.*

format identification (FID) field. In SNA, a field in a transmission header (TH) that defines the subsequent format of the header and the type of TH fields involved with a transmission. FID0 (for pre-SNA product support) and FID1 are the header formats used between host and communications controller nodes and between two communications controller nodes. FID2 and FID3 are the header formats used between communications controller nodes with boundary function and cluster controller and terminal nodes.

formatted system services (FSS). A portion of ACF/VTAM that provides certain system services as a result of receiving a field-formatted command, such as an Initiate or Terminate command. Contrast with *unformatted system services (USS).* See also *field-formatted.*

function management (FM). (1) In SNA, the layer of functional capability between the application layer and the transmission subsystem. It includes data flow control and function management data (FMD) services. See also *application layer, transmission subsystem.* (2) In ACF/VTAM, the insertion of control information within messages so that the messages sent to a particular type of terminal are in the required format and so that messages received from that type of terminal are handled properly.

function management data (FMD) services. In SNA, the component of function management responsible for request/response units marked as "FM data." This includes presentation services and logical unit services (within the logical unit), physical unit services (within the physical unit), and network services (within the system services control point). Contrast with *data flow control.*

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 master key. A primary *key encrypting key* used to *encipher operational keys* that are used at the host processor. In ACF/VTAM, a *key encrypting key* used to *encipher* and *decipher cryptographic session keys.*

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

ICV. *Initial chaining value.*

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.*

initial chaining value (ICV). An eight-byte random number used to verify that both ends of a *cryptograhpic session* have the same *cryptographic session key.* It is generated by the secondary end of the session upon receipt of a Bind request for a cryptographic session and is returned to the primary end of the session in the Bind response, enciphered under the cryptographic session key received in the Bind request. The initial chaining value is also used as input to the CIPHER macro to *encipher* or *decipher* data in a cryptographic session.

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.

K

key encrypting key. A *cryptographic key* used to *encipher* and *decipher* other cryptographic keys. Contrast with *data encrypting key.*

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 address. In SNA, an address transformed to or from a network address by the boundary function (for example, in a communications controller node) for use by a cluster controller node or terminal node. See also *network address, boundary function.*

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.*

mandatory cryptographic session. See *required cryptographic session.*

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

NAU. Network addressable unit.

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.*

negative response to polling limit. For a start-stop or BSC terminal, the maximum number of consecutive negative responses to polling that the communications controller accepts before suspending polling operations.

network. (1) (SC1) The assembly of equipment through which 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 address. In SNA, the address, consisting of subarea and element subfields, that uniquely identifies a link or the location of a network addressable unit. The conversion from a local address to a network address, or vice versa, is accomplished as part of the boundary function in the node attached to a cluster controller node or a terminal node. See *local address.* See also *network name.*

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted in the transmission subsystem. Each NAU has a network address that represents it to the transmission subsystem. The transmission subsystem and the NAUs collectively constitute the communication system. See also *network name, network address.*

network control (NC). In SNA, a transmission control component that permits logically adjacent connection point managers to communicate through the common network, using sessions established for other purposes and thereby avoiding special session establishment. See also *connection point manager, session control.*

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 name. In SNA, the symbolic identifier by which a network addressable unit or a data link is referred to by end users. See also *network address.* In ACF/VTAM, for multidomain users, the name of the APPL statement is the network name and must be unique across domains. Contrast with *ACB name.*

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. A remote BSC 3270 terminal defined with PU=YES is considered an SNA terminal.

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

operational key. A *data encrypting key* used to *encipher* and *decipher* data. In ACF/VTAM, synonymous with *cryptographic session key.*

orderly closedown. The orderly deactivation of ACF/VTAM and the communication network. An orderly closedown does not take effect until all application programs have been disconnected from ACF/VTAM. Until then, all data transfer operations continue. Contrast with *cancel closedown* and *quick closedown.*

P

pacing. 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.

path control (PC). In SNA, one of the components of the transmission subsystem, and one of two components of the common network. It is responsible for managing the sharing of data link resources of the common network and for routing basic information units (BIUs) through it. It is aware of the location of NAUs in the network and of the paths between them. It maps the BIUs, handled by transmission control, into path information units (PIUs), and then into basic transmission units (BTUs) that are passed between path control and data link control. The unit of control information built by the sending path control component and interpreted by the receiving path control component is a transmission header (TH). See also *data link control, transmission control.*

path information unit (PIU). In SNA, the unit of transmission consisting of a transmission header (TH) and either a basic information unit (BIU) or a BIU segment.

path table. A table, contained in a network node, whose entries contain path information. For example, the ACF/VTAM table constructed from PATH statements, that lists the communications controllers adjacent to the host computer (called *adjacent subareas*) that are used for cross-domain communication and indicates for each controller the subareas in other domains (called *destination subareas*) for which messages are to be routed through that controller.

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.

PIU. Path information unit.

plaintext. Synonym for *clear data.*

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

RDT. Resource definition table.

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.*

required cryptographic session. A *cryptographic session* in which all outbound data is enciphered and all inbound data is deciphered. Contrast with *selective cryptographic session* and *clear session.*

resource definition table (RDT). In ACF/VTAM, a table that describes for a major node the characteristics of each node available to ACF/VTAM and associates each node with an address.

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 e 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.

secondary logical unit (SLU) key. A primary *key encrypting key* used to protect a *cryptographic session key* during its transmission to the secondary end of the session.

selective cryptographic session. A *cryptographic session* in which an authorized application program is permitted to specify the enciphering of certain request units.

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.*

SLU. Secondary logical unit.

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.*

specific node. See *minor node.*

SSCP. System services control point.

SSCP ID. An identifying number associated with an SSCP (that must be unique in a multidomain system) that enables a device (especially a dial-in device) to identify an SSCP at a particular location and enables another SSCP to identify this SSCP when establishing a session with it.

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

TC. Transmission control.

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.

terminal system. In a data communication network, a terminal control unit or a cluster control unit and its attached devices. Synonymous with *teleprocessing subsystem.*

TH. Transmission header.

transit node. An intermediate node that is capable of routing path information units to another domain.

transmission. In data communication, one or more blocks or messages. For BSC and start-stop devices, a transmission is terminated by an EOT character. See also *block* and *message.*

transmission control (TC). In SNA, one of three components of the transmission subsystem. It has three subcomponents: the connection point manager, session control, and network control. It establishes, controls, and terminates sessions, and also controls the flow of information into and out of the common network for a session between network addressable units. It provides access to the transmission subsystem; this direct access is used by function management components. A transmission control element exists for each active session. See also *data link control, path control.*

transmission header (TH). In SNA, a control field attached to a basic information unit (BIU) or to a BIU segment, and used by path control. It is created by the sending path control component and interpreted by the receiving path control component. See also *path information unit.*

transmission subsystem. In SNA, the innermost layer of the communication system. It provides the control in each session to route and move data units between NAUs, and to manage the NAUs and their interconnecting paths. Its three constituent parts are data link control, path control, and transmission control. See also *application layer function management.*

U

unformatted system services (USS). A portion of ACF/VTAM that translates a character-coded command, such as a logon or logoff command, into a field-formatted command for processing by formatted system services (FSS). Contrast with *formatted system services (FSS)* and *character-coded.*

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.*

user logon data. Synonymous with *logon data (1).*

Index

Advanced Communications Function
for VTAM (ACF/VTAM)
System Programmer's Guide

Order No. SC38-0258-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

[ ]	As an introduction	[ ]	As a text (student)
[ ]	As a reference manual	[ ]	As a text (instructor)

[ ] 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:

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.)

Cut or Fold Along Line

SC38-0258-1

Reader's Comment Form

Cut or Fold Along Line

Fold   Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 40   ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 63T
Neighborhood Road
Kingston, New York 12401

Fold   Fold

ACF/VTAM System Programmer's Guide (File No. S370-30) Printed in U.S.A. SC38-0258-1

IBM®

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

Advanced Communications Function
for VTAM (ACF/VTAM)
System Programmer's Guide

Order No. SC38-0258-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.
IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

[ ] As an introduction [ ] As a text (student)

[ ] As a reference manual [ ] As a text (instructor)

[ ] 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:

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.)

Cut or Fold Along Line

SC38-0258-1

Reader's Comment Form

Cut or Fold Along Line

Fold

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 63T
Neighborhood Road
Kingston, New York 12401

Fold

Fold

IBM®

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

SC38-0258-1

ACF/VTAM System Programmer's Guide (File No. S370-30) Printed in U.S.A. SC38-0258-1

IBM®

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