

VTAM Installation and Coding

Mini-Course 1

VTAM Major Component and Installation Overview

Mini-Course 1. ACF/VTAM Major Component and Installation Overview

Introduction

Advanced Communication Function for VTAM (ACF/VTAM) is a product in the line of Virtual Telecommunications Access Methods (VTAM) designed along SNA guidelines. This mini-course describes the major components of an ACF/VTAM system with emphasis on the definitions and terminology of a data communications system, and presents an overview of basic ACF/VTAM installation coding requirements.

Physical Components

The major physical components of an ACF/VTAM system are the host processor and any combination of local terminals, communications controllers, and communications lines. These and other physical components necessary for data communications in an ACF/VTAM environment are described in the subsections that follow.

Host Processor

The host processor in which ACF/VTAM resides is any IBM System/370, 30xx, or 4300 Processor which supports one of four IBM operating systems:

- MVS/System Product Version 1 (with JES2 or JES3) for System/370
- MVS/System Product Version 2 (with JES2 or JES3) for Extended Architecture (XA) processors
- Virtual Storage Extended/System Package Version 2 Release 1 Modification Levels 3 or 4
- Virtual Machine/System Product Release 4 (with or without the High Performance Option, HPO).

Earlier ACF/VTAM support varied by product version and release levels. For example, the Operating System/Virtual Storage 1 (OS/VS1) is supported through ACF/VTAM Version 2 Release 1. ACF/VTAM Version 3 for VM/SP is the first version which gives native SNA support to VM systems.

Important: Abbreviations for these products will be used throughout the Text and Personal Reference Guide as follows:

- VTAM for ACF/VTAM
- MVS for both MVS/SP products unless specified
- VM for VM/SP.

When no reference is made to a specific operating system, the text applies to all of the operating systems.

A host processor in an SNA network is defined as a processing unit that contains a system services control point (SSCP). The SSCP in VTAM manages the configuration, provides network operator services and network end user session services. The term VTAM host is used to designate a single VTAM that manages all of the resources defined to it. The collective group of a VTAM host and its resources is called a VTAM domain.

A VTAM host is capable of communicating with another or several VTAM hosts. When a configuration consists of multiple hosts there are multiple domains. A common term for this environment is multidomain, or multisystem network, or simply network.

Version 2 Release 2 of VTAM for MVS, along with ACF/NCP Version 3, introduced the capability for two or more independent networks to communicate with each other. This capability relies on user-defined gateways to provide access from one network to another. This capability is called SNA Network Interconnection (SNI).

Network Addressing

The addressing scheme in a network is analogous to postal service ZIP codes and street addresses. The high level address in a network is a unique subarea number assigned to each VTAM and each communications controller in the network. Like ZIP codes, messages carrying the unique number of their destination identify the location of the subarea in the network. With VTAM, the origin subarea number is also identified in all messages.

A second part of the address, analogous to a street address, is called the element address. Each resource in the network that is to send and receive messages is assigned an element address within the subarea where it is defined. These resources may be referred to as network addressable units (NAUs).

Although the subarea numbers are assigned by the user, element addresses are assigned during the activation process of either VTAM or a communications controller network control program. The Advanced Communications Function for Network Control Program (NCP) provides the control function for IBM communications controllers. NCP is a

generated program based on user definitions of all network resources attached to the communications controller. User-defined physical device addresses are assigned resource identifications which correspond to element addresses. When VTAM contacts an NCP, it will use the resource IDs to build its table of element addresses that belong to that NCP subarea.

Application programs and channel-attached NAUs which are defined to VTAM receive element address assignments at the time they are made known to VTAM. Since these programs and devices are defined by the user in groups known as major nodes, the sequence of activating the major nodes (that is, making the resources known to VTAM) may vary. Element addresses in a VTAM subarea, therefore, are dynamically assigned after VTAM is started.

Prior to Version 3 of VTAM and Version 4 of NCP, subarea and element addresses were contained in a 16-bit structure. The 16 bits were divided from a 2-bit subarea (maximum of 3 subareas), 14-bit element (maximum of 16,384 elements) structure up to 8 bits for the subarea address and 8 bits for the element address (maximum 255 for subareas, 256 for elements).

Extended network addressing (ENA) is a function of Version 3 for VTAM and Version 4 for NCP. The ENA network addressing structure is 8-bits for subarea number and 15-bits for element addresses (maximums of 255 for subarea and 32,768 for elements).

VTAM Environments

VTAM supports communication devices that are attached to a data channel of the host processor. Also, devices attached to 4300 Loop Adapters or 4300 Display/Printer Adapters are supported. These devices represent the VTAM local environment. VTAM supports the local environment as part of its domain with or without other installed communications equipment.

Note: The VTAM considerations for the local environment will be described in more detail in Mini-Course 2. The VTAM considerations for remote SNA terminals in a single domain will be described in Mini-Courses 7 through 12.

Each VTAM in a multidomain network has a section of code called a cross-domain resource manager (CDRM), which controls cross-domain sessions. This allows application programs and terminals that are controlled by VTAM to communicate with application programs in another domain of the multiple system network. It also allows terminals controlled by another domain in the network to communicate with VTAM application programs.

Note: VTAM considerations for cross-domain communications will be described in Mini-Course 16.

Communications Lines

A *communications line* is a physical link, such as a telephone circuit, that connects one or more remote terminals to the host processor, or connects one domain with another. Various terms are used to describe how particular communications lines are connected.

A line that connects a host processor and a single remote station is called a *point-to-point line*. If a single line from the host connects to two or more remote stations, it is termed a *multipoint line*. A permanent connection between the end points of the line is said to be a *non-switched* connection.

Non-permanent connections are also possible between the host processor and multiple remote stations. These *switched* connections work on the same principle as the telephone in that a remote station can be dialed from the host. When transmission of data is completed, another station may be dialed by the host. Dialing may be accomplished by manual dialing, or, with optional equipment, dialing may be automatic. A remote station may also dial the host for a connection if the host site is equipped with automatic answering facilities.

Note: The VTAM considerations for SNA terminals on switched communications lines will be described in Mini-Course 14.

Although transparent to the user, data transmission may be via equipment such as fiber optic cables, microwave facilities, coaxial cables, or satellite networks. Since the actual path that data travels may not be a physical pair of wires, as line implies, the terms data communications path, data link, or communications channel are often used interchangeably with communications line.

Line Configurations

Transmission of data in both directions at the same time is possible if two circuits (that is, a four-wire facility) are provided. This simultaneous two-way independent transmission is called a *full-duplex* communications channel. Full-duplex operation is similar to traffic on a two-lane highway. Thus, a remote station may begin a transmission at the same time that it is receiving from the network. For example, 37xx communications controllers use full-duplex operation when communicating with each other.

The term *half-duplex* describes a communications line which provides alternating, one-way at a time, independent transmission. Half-duplex operation is similar to traffic over a one-way bridge. For example, a remote station must wait for completion of a transmission it is receiving before it can send any information over the line. Half-duplex operation may be over 2-wire or 4-wire facilities. In 2-wire operation, the line is logically reversed at each end before transmitting. With 4-wire operation, one pair of wires is used for receiving, the other pair for transmitting, and no line reversal is necessary.

Normally, the configuration of a particular communications line is given by a combination of terms. For example, valid line configurations are:

- Half-duplex, point-to-point (non-switched)
- Half-duplex, multipoint (non-switched)
- Half-duplex, point-to-point (switched)
- Full-duplex, point-to-point (non-switched).

Note: VTAM in VM or VSE with a communications adapter only supports half-duplex operation. However, 4-wire data communications facilities (in half-duplex data mode) are supported.

Data Terminal Equipment

Two fundamental things must be done to the binary digital data between the main storage of the processor and the communications channel:

- The data and control information must be converted to a serial stream of binary digits from a parallel bits-per-character structure.

This is accomplished by one of a general family of devices called data terminal equipment (DTE). IBM 3705 and 3725 Communications Controllers are examples of data terminal equipment.

- The serialized binary signals must be made compatible with the communications channel.

This is done by a modem (for modulate-demodulate) which is either internal to the DTE or a separate piece of equipment. A modem may also be known as data circuit-terminating equipment (DCE).

The receiving equipment must reverse both processes. The receiving modem recovers the binary stream and the DTE regroups (deserializes) the data into a parallel bit structure.

VTAM supports EBCDIC (extended binary coded decimal interchange code), an 8-bit data coding structure, for all data transmission.

Note: The ASCII code structure is not supported by VTAM for network traffic, although it is supported by communications controllers and communications adapters for use with other teleprocessing access methods. VTAM will support the ASCII 8-bit code defined in ANSI X3.41-1974 for logical unit sessions. End user components are made aware of either the 7-bit or 8-bit ASCII code for proper translation.

Control Devices

Communications control units (CCU) are devices that provide data terminal equipment (DTE) functions to control the transmission of data over lines in a network. There are three major types of CCUs:

- Communications Controllers
- Communications Adapters
- Transmission Control Units.

Communications Controller

A *communications controller* is a device that contains a control program within the unit to control data transmission over many communications lines. For example, an IBM 3725 Communications Controller with NCP can concurrently control data transfer over communications lines assigned to VTAM along with other lines assigned to other communications access methods.

Communications controllers are attached to processor byte multiplexer, block multiplexer, or selector channels. Only one channel address may be used between VTAM and each communications controller attached to the host processor. However, a communications controller may be attached to more than one host processor. For example, up to six host processors may be attached to a 3725 communications controller.

Note: NCP generation is beyond the scope of this course, however, VTAM considerations for NCP will be discussed in Mini-Course 11.

Communications Adapters (CA)

The *communications adapter (CA)* is an optional hardware feature of an IBM 4300 Processor that provides for the attachment of up to eight data communications lines. The CA has preallocated subchannel addresses for its lines and operates in conjunction with VTAM to control the transfer of information. The fixed addresses for the CA communications lines are 030 through 037.

The communications adapter supports the various line speeds up to a maximum of 9,600 bps for any one line. There is one exception: one of the eight lines may be a synchronous high-speed line (BSC or SDLC) up to 56,000 bps; however, the aggregate data rate capability of all CA lines installed must not exceed 64,000 bps. Line speeds over 9,600 bps are used to communicate with other systems such as other 4300 Processors, IBM 8100 Information System equipment, or a communications controller.

Communications adapters are supported by VTAM only in a VSE or VM operating system environment.

Note: VTAM considerations for communications adapters with VSE and VM will be discussed in Mini-Course 12.

Transmission Control Unit (TCU)

A third type of CCU is a *transmission control unit (TCU)*. This hardware unit provides DTE line control functions, but has no internal program to provide additional functions. The additional functions are still necessary and must be within a program in the host computer that the TCU serves. Each line attached to a transmission control unit must be associated with a host subchannel.

VTAM does not support connections to a transmission control unit.

NCP can be generated to emulate a TCU for other access methods. The ACF/NCP emulation program (EP) support can be stand-alone or combined with network control mode (NCP) in a partitioned emulation program (PEP). VTAM will use only the NCP portion of an NCP generated for PEP operation.

Data Link Control

A combination of the data terminal equipment (DTE), the modems, the communications channel, and the necessary controls make up a *data link*, often referred to as simply a *link*. Control of the data link includes activities that are not part of the actual data being transferred.

Data link controls are needed only to operate the data link itself. System control information, such as input/output device controls, are not considered data link controls. The following are data link control activities:

- Synchronizing - getting the receiver in step with the transmitter
- Detecting and recovering transmission errors
- Controlling send/receive transmissions between stations
- Reporting improper data link control procedures to a higher level.

Line Discipline

A *line discipline* defines how data is transmitted over the data link. There are three major line disciplines (or line protocols) used for data communication:

- Synchronous Data Link Control (SDLC)
- Binary Synchronous Communications (BSC)
- Start-Stop (S/S).

SDLC

When transmissions are a continuous, synchronous stream of binary signals, they comprise a line discipline called *synchronous data link control (SDLC)*. SDLC is a line discipline for the management of information transfer over a data communications channel. Transmission exchanges may be full-duplex or half-duplex; the communications channel configuration may be point-to-point or multipoint; a point-to-point configuration may be switched or non-switched. SDLC includes comprehensive detection and recovery procedures at the data link level for transmission errors that may be introduced by the communications channel.

IBM SNA devices are designed to use the SDLC protocol.

BSC

Binary synchronous communications (BSC) is another line discipline which depends on synchronization between sending and receiving hardware. The BSC hardware automatically generates special synchronizing characters at the beginning of each transmission and sometimes during the transmission. A set of BSC control characters allows the sending and receiving hardware to converse, so that a stream of bits may be separated into characters. Strings of characters, delineated by BSC control characters, are sent as blocks of information. After transmission of one or more blocks, the receiving equipment sends a positive or negative acknowledgment to verify reception, or to notify the sending equipment of an error.

The VTAM BSC device support is for those terminals capable of using non-switched EBCDIC BSC 3270 line protocol. For example, the 3270 Information Display System and the 8100 Information System are supported.

Note: The remote non-SNA (BSC) terminal environment will be considered in more detail in Mini-Course 13.

Start-Stop

In addition to SDLC and BSC line disciplines, a third method of data transmission is one where each character is preceded and ended with a special bit or bits. This method works something like a telegraph operator who sends a single character at a time, separating each character with a slight pause. There is no synchronization since each character is sent down the line when the operator taps the key. Transmissions of this nature are called *asynchronous* since the rate of characters being sent can vary. Communications systems using asynchronous line control are known as *Start-Stop* systems.

Note: VTAM does not support asynchronous (Start-Stop) communication devices directly; however, NCP in conjunction with the IBM Network Terminal Option (NTO) program product allows VTAM to support selected start-stop devices. With MVS, an IBM 3710 network controller may be used to provide protocol conversion for start-stop devices. Communications adapters support start-stop lines for access methods other than VTAM.

CA Line Disciplines

Only two of the three line disciplines, SDLC, BSC, and S/S, may be installed on the CA at one time. The lines may be split between two access methods in any grouping. For example, VSE VTAM and the Basic Telecommunications Access Method - Extended Support (BTAM-ES) program product may run concurrently in the same 4300 Processor, each having its own set of lines. However, they are restricted from concurrently sharing any single line.

Data Flow

VTAM manages the flow of data in the network, although it does not actually transfer the data. The flow of data is between application programs and VTAM, and between VTAM and communication controllers or communication adapters to remote terminals. The actual flow of data from VTAM to communication controllers or communications adapters and to the lines (or to the channel for local devices) is accomplished by the I/O facilities of the operating system.

VTAM Resources

The resources of a VTAM domain include application programs, communication lines, control units, and terminals.

Application Programs

VTAM application programs that reside in the host processor are made up of two parts:

- The telecommunications program which uses VTAM macros
- The application processing program.

This separation allows each part to be created independently and means that changes to one part need not affect other parts. For example, the telecommunications program may be the Customer Information Control System (CICS/VS) with its corresponding application (transaction) processing programs.

Other application programs that are not VTAM application programs can be used to control the operation of certain types of terminals. These programs reside in cluster control units.

Cluster Control Units

A *cluster control unit* is a device that controls the input and output operation of a group of terminals (called a cluster). The cluster control unit may be designed to operate with either an SNA or a non-SNA protocol depending on the terminals to be attached.

SNA Terminals

SNA terminals are designed according to SNA guidelines to have physical unit (PU) functions and logical unit (LU) functions. An SNA cluster controller can contain more than one logical unit for its single physical unit. The I/O devices (terminals) attached to the cluster control unit could all be controlled by a single logical unit or by several logical units. Also, a logical unit need not control any device; it might perform some other user-defined function.

SNA devices are physical devices which have differing characteristics. The major physical unit characteristics are identified by a physical unit type. For example, an SNA terminal, such as a 3767, is represented by a physical unit type 1 (PU__T1), while an SNA cluster control unit, such as the 3790 Communications System, is represented by a physical unit type 2 (PU__T2). (NCP in a communications controller is represented as a PU__T4.)

Non-SNA Terminals

Terminals that do not use SNA protocols are called *non-SNA terminals*. A non-SNA terminal is seen by VTAM as a single logical unit or as a cluster control unit with associated logical units. With these devices, there is a one-to-one relationship between the logical unit and the physical device.

The non-SNA devices are supported by VTAM as physical unit type 1 (PU__T1) although VTAM or NCP will supply the PU functions.

■ ■ ■

Please turn to Mini-Course 1, Exercise 1.1, in your Personal Reference Guide and answer the exercise questions.

Basic VTAM Installation Requirements

There are five basic areas for consideration when installing a VTAM system.

- Application program identification to VTAM.
- The hardware configuration definition to VTAM.
- Session requirements and options to be used.
- VTAM initialization start options.
- VTAM initial startup configuration.

The considerations for each of the five areas are based on what the VTAM system is expected to do. More complex networks require a greater extent of considerations. The installation coding effort in each of the areas is a combination of VTAM requirements and user choices of VTAM options.

Refer to Figure 1-1 and Figure 1-2 for the following discussion.

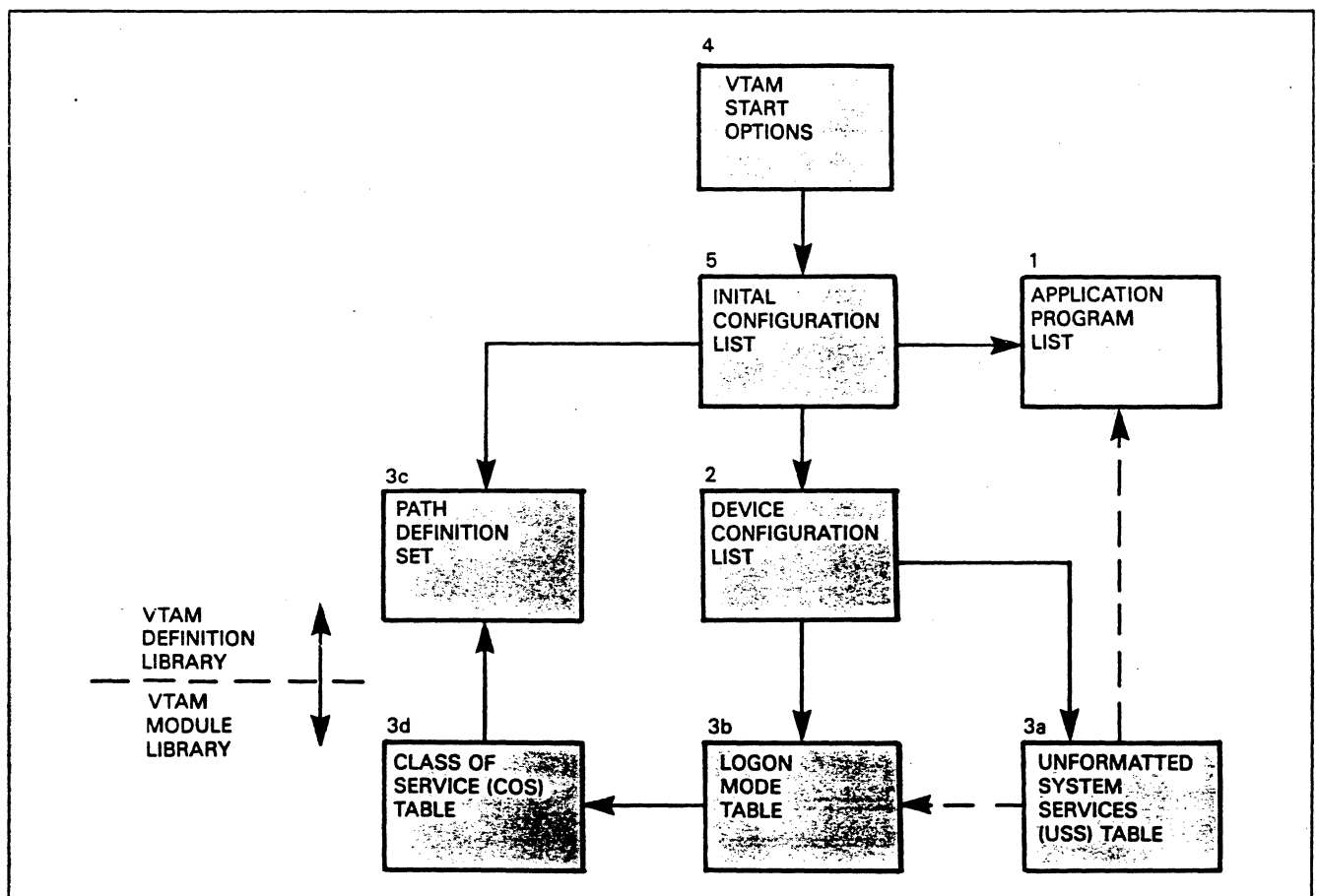
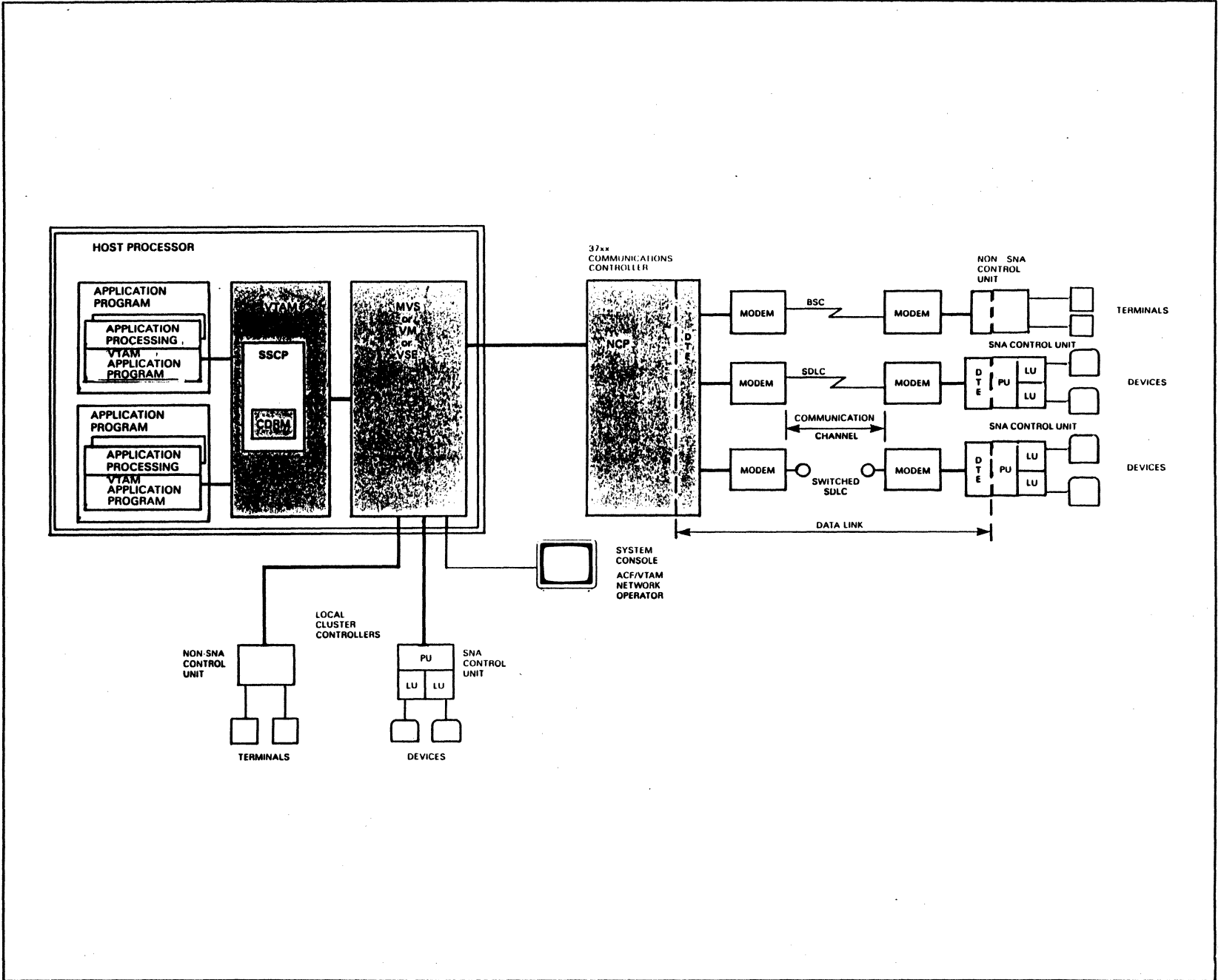


Figure 1-1. VTAM Installation - Five Basic Areas

Figure 1-2. VTAM Reference Diagram



Application Program Identification to VTAM

Application program lists (see item 1 in Figure 1-1) provide VTAM with information for each application program that will communicate with VTAM managed resources. The only required information for VTAM to recognize a particular program is a unique name. Optional keyword parameter considerations are dependent on:

- The types of sessions to be established
- The operating system
- The types of VTAM instructions and exits used in the program.

Application programs fall into three major categories:

- IBM Program Products
- IBM Program Offerings
- User written programs.

The Hardware Configuration Definition to VTAM

Device configuration lists (see item 2 in Figure 1-1) define the physical components of the network to VTAM. There are six general categories of physical component definitions to consider:

- Channel-attached SNA devices
- Channel-attached non-SNA devices
- Channel-attached and link-attached communications controllers
- Communications adapter (CA) attached devices (VSE and VM)
- Channel-to-channel-adapter (CTCA) attached processors
- Devices attached to switched network connections via a communications controller or a communications adapter.

These lists define device characteristics, device addresses, and VTAM options for the device.

Session Requirements and Options To Be Used

Session options fall into four major areas of consideration:

- How the session is to be initiated (logon options)
- The session parameters (protocols or rules) to be used by the session partners
- Routing options (or path) for messages between the session partners
- Class of service (COS) for the session traffic.

Four tables associated with session initiation are shown in Figure 1-1 (items 3a-3d). The tables correspond to the four areas of consideration for session initiation. However, it should be noted that all tables may not be necessary depending on user choices. For example, VTAM provides many defaults, either internally or in supplied tables, that might be selected by the user in preference to coding one or more of the tables.

Logon Options

Logon options refer to how a particular session between a VTAM application program and a VTAM resource is to be initiated. A session may be initiated by one of the following:

- A VTAM application program
- An end user at a terminal
- A programmable terminal
- Automatically by VTAM
- The network operator.

The unformatted systems Services (USS) table (see item 3a in Figure 1-1) is used in a form of end user logon. Logon commands coded in optional USS tables allow an end user to type only a command word to logon to a VTAM application program. For example, an end user might type only the characters *CICS* to log on to an application program without having to know specific VTAM required information.

A default USS table is supplied with VTAM. It includes standard formats for logons and logoffs, standard USS messages, and a lower-to-upper case translation table. The four types of session initiation other than an end user at a terminal do not require a USS table.

Session Parameters

Session parameters are a set of rules (protocol as established by Systems Network Architecture) to govern how an application program and a VTAM resource are to communicate with each other. Session parameters may be supplied by the following:

- The application program
- User coded table entries in logon mode tables (see item 3b in Figure 1-1)
- The VTAM default logon mode table.

Session parameters for devices must conform to the capabilities of the type of device hardware. Application programs may be written to support any protocol and to match the requirements of different devices. In the case of application program to application program sessions, the parameters depend on coding within the programs themselves.

Routing Options

Routing is the general term used to describe the path of message transfer between two VTAM resources. Paths are chosen and defined by the user depending on the physical paths (routes) available in the network. Paths must be defined when the network includes one or more of the following:

- Channel-attached communications controllers
- Link-attached communications controllers
- Other host processors.

Path selections are coded in path definition sets (see item 3c in Figure 1-1). A path must be defined from the VTAM host to all communications controllers and host processors with which it will communicate.

Class of Service (COS)

Class of service (COS) allows a user to define a sequential list of routes for VTAM to use at session initiation time. Each route in the list is tried, in order of appearance, until an available route is found for the session. In addition, the user may choose one of three message priorities for the session if the route traverses multiple communications controllers. COS entries are coded in a class of service table (COS table, shown as item 3d in Figure 1-1). Each terminal/LU definition in the network may point to a COS entry, or if not, VTAM will select a default class of service.

VTAM Initialization Start Options

Start options (see item 4 in Figure 1-1) are a list of parameters supplied to VTAM when it is first initialized, that is, when VTAM is started under an operating system. The list includes options to:

- Identify VTAM to the network
- Set performance criteria
- Assist the network operator
- Assist in problem isolation
- Gather statistics
- Select an initial network configuration.

VTAM may be started or restarted at any time with the same list of options or with an alternate list. For example, a particular set of start options might be used in the test environment, then another set might be selected for the final production environment.

VTAM Initial Startup Configuration

A configuration list (see item 5 in Figure 1-1) is a list of pointers to other members in the VTAM definition library that are to be automatically made an active part of the VTAM domain. The list is invoked by one of the start options; therefore, the same or a different configuration may be activated each time VTAM is started. When no configuration list is supplied (pointed to by the start option), the VTAM network operator must activate the desired configuration.

■ ■ ■

Please turn to Mini-Course 1, Exercise 1.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 2

Installation Requirements — Application Program Major Node

Mini-Course 2. Installation Requirements - Application Program Major Node

Introduction

Early telecommunication application programs owned their terminals and lines, and no other program could use any terminal on the owned line. As a result, networks were larger than desirable, because each different application program needed its own set of lines and terminals. And, when a program terminated, all lines and terminals that had been owned by that program were dead.

VTAM, however, allows resources to be shared. A VTAM application program does not own a line. The program need only know the 8-byte network name of a network node in order to be able to communicate with that node. The network itself is owned and managed by VTAM, and not by the VTAM application programs. An application programmer need not be concerned about VTAM programs or logical units (LUs) in the network that are not related to the relevant application program.

In effect, a VTAM application program owns only those LUs to which the program is connected at a given point in time. If the program releases an LU or if the program terminates, the LU can then be immediately connected to some other active program and perform useful work.

VTAM Requirements

VTAM is designed for flexibility in the operation and control of a communications system. A user who has a fixed physical configuration may require that only portions of it be active at a particular time. For example, a different set of lines and terminals may be required for application programs run during the day than those run at night.

To provide for this flexibility, the configuration is described to VTAM according to logical groups of components. The description is based on both VTAM requirements and user requirements. Each time VTAM is started, different sets of logical groups may be activated, to select the appropriate configuration. In addition, the VTAM network operator may dynamically build or change the configuration after VTAM startup.

VTAM requirements were introduced in Mini-Course 1. More specifically, the user coding requirements include:

- An application program major node
- One of several types of device major nodes
- A source for session parameters
- A start options list to define VTAM system parameters
- A configuration list to define user-selected major nodes for automatic activation at VTAM startup time.

VTAM will start without a configuration list; however, it will display an error condition before continuing. If the user chooses to have no major nodes initially activated, a dummy configuration list should be filed.

The relationship of the start options, configuration list, and major nodes is shown in Figure 2-1.

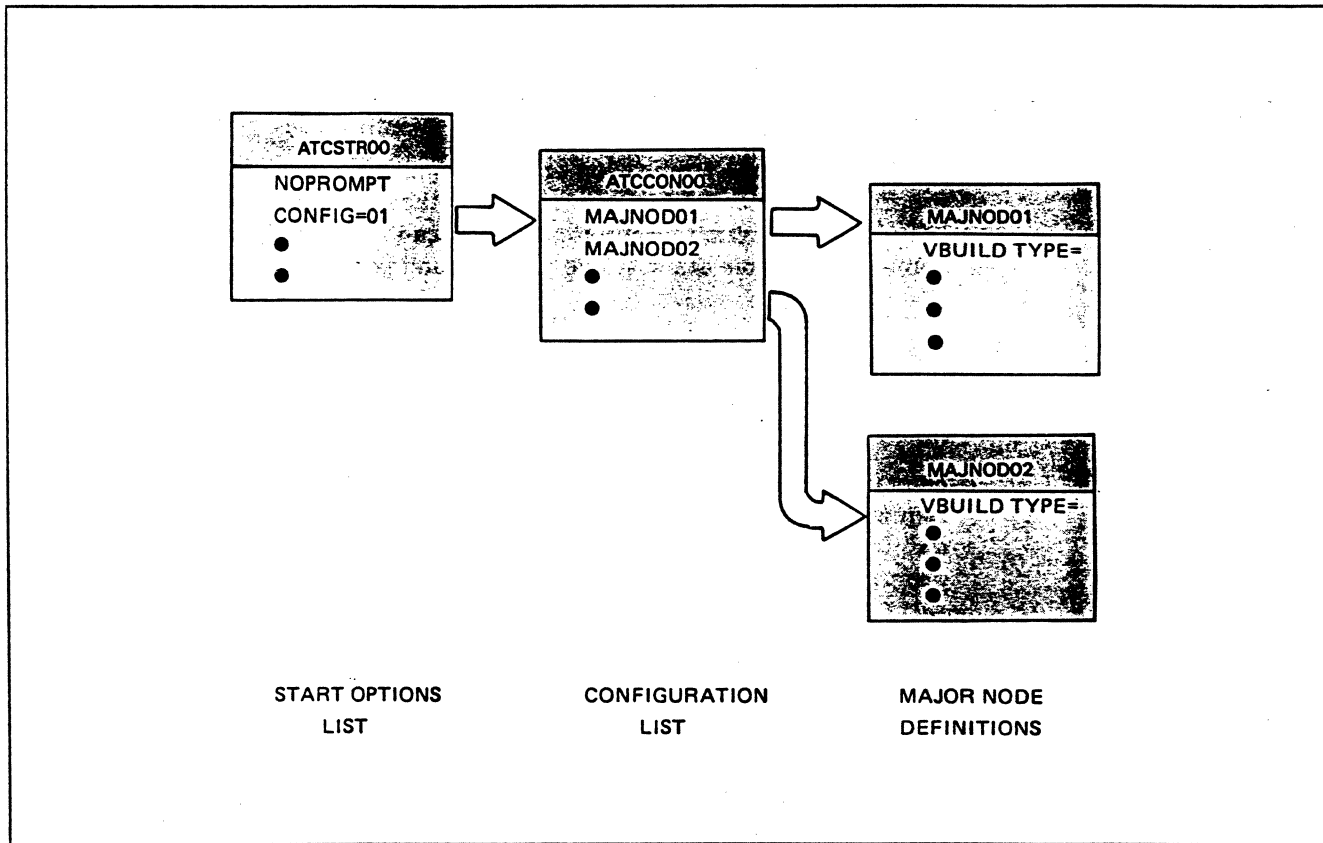


Figure 2-1. VTAM Startup Relationships

Major and Minor Nodes

A *major node* is a set of controllable resources in the VTAM domain. The set is composed of *minor nodes* which may be controlled either independently or as a major node group.

The hierarchical structure of major and minor nodes allows a user to control a group of nodes as a single unit. By activating a major node, one can activate all, any, or none of the minor nodes subordinate to it. However, when activating a minor node independently, all higher-level nodes must be active prior to its activation. Similarly, when deactivating a major node, all minor nodes subordinate to it are automatically deactivated. Deactivation of minor nodes does not deactivate higher-level nodes.

Major Node Types

There are six possible types of VTAM major nodes in a single-domain environment:

- Application program
- Local non-SNA
- Local SNA
- Communications controller (NCP)
- Channel-attached for a 4300 processor communications adapter (VSE and VM)
- Switched network.

There are four additional major nodes that may be used in a multidomain environment:

- Cross-domain resource manager (CDRM)
- Cross-domain resources (CDRSC)
- Channel-attached for a channel-to-channel adapter (CTCA) (MVS and VM)
- Channel-attached for a VTAM data host connection to an NCP owned by another host.

Definition Statements

The user defines the configuration by coding VTAM macros with definition statements. Each major node is filed in its source form as a separate member and becomes a part of the VTAM definition library. In MVS the definition library is always the partitioned data set SYS1.VTAMLST pointed to by the VTAMLST DD statement. The VTAMLST member name is the major node name.

In VSE, the VTAM definition library is either a system or private library with members that have a B suffix on the major node name (for example, MAJNOD01.B).

VTAM major nodes in VM are CMS files with a filename of the major node name and a filetype of VTAMLST. The CMS minidisk is linked to the VTAM virtual machine for VTAM access of the definitions.

VTAM uses the major node name to collectively reference the set of subordinate minor nodes. Each minor node is referenced by the name on its definition statement.

Note: The Assembler Language macro coding format is used for all statements: name field, operation field, and operand fields.

VBUILD Statement

All major node definitions begin with a VBUILD macro statement except for local non-SNA terminal major nodes (LBUILD statement) and communications controller NCP major nodes. The VBUILD statement identifies the type of major node being defined.

The minor node definitions for each major node follow the VBUILD statement and are defined with VTAM macro statements. The number of minor nodes defined for each major node is dependent on the configuration and how it is to be used. For example, local SNA terminals may all be defined in a single major node, or, the user may choose to define them in two or more separate major nodes.

In an NCP major node, all lines, groups of lines, and terminals connected to a communications controller become VTAM minor nodes; that is, one NCP major node for each communications controller. The source statements used to generate the NCP are placed in the VTAM definition library using a major node name equal to the name on the NEWNAME parameter in the NCP BUILD macro.

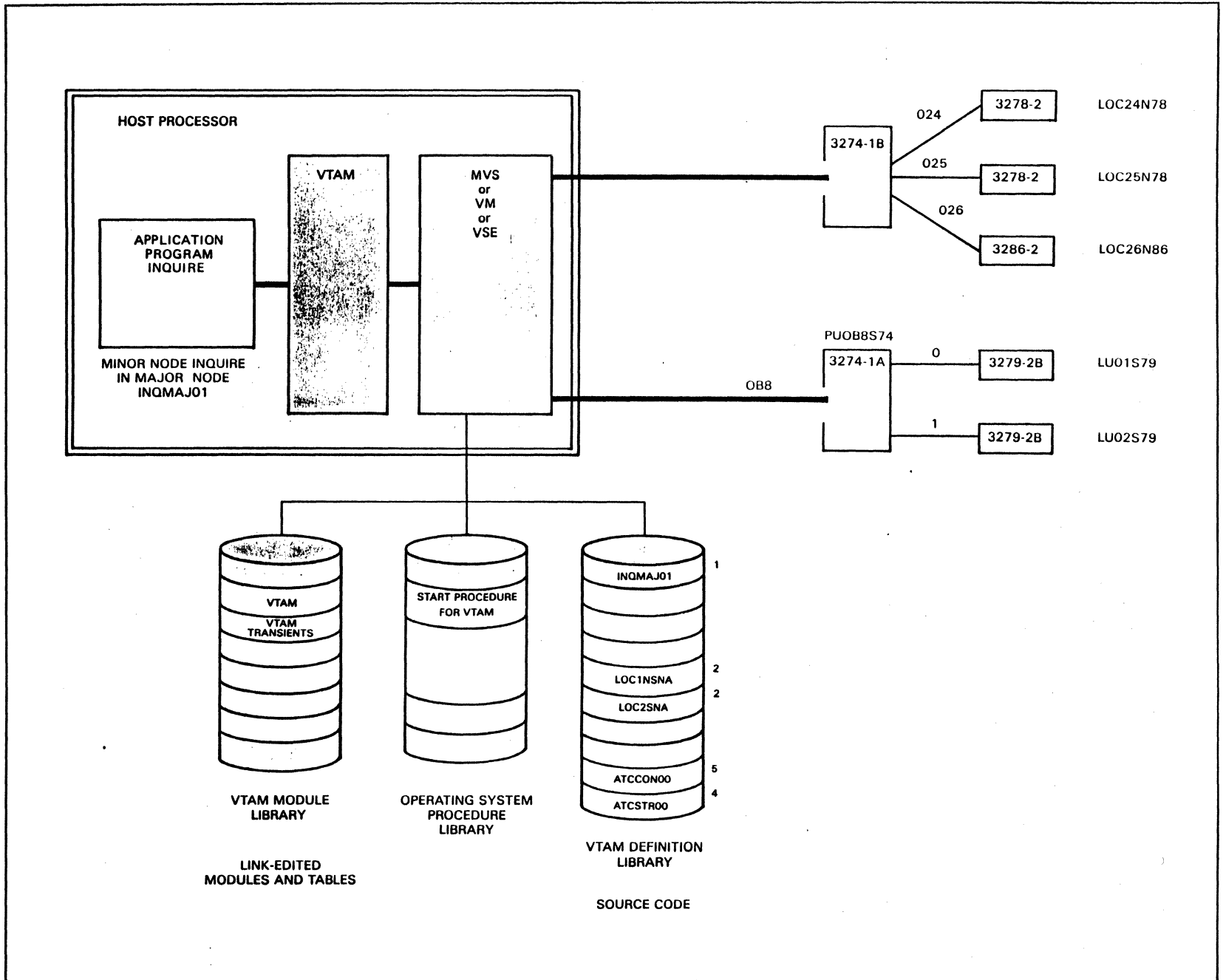
The selection of minor nodes for each type of major node is determined by the user from the network configuration. Some of the considerations are:

- The number of applications
- The terminal needs for each application at VTAM startup
- The terminal needs for applications activated by the domain operator after VTAM startup
- The terminals that are to be free for logon to any application.

Example Configuration

One of the first steps to any installation is to have a complete picture of the physical environment. Figure 2-2 shows an example configuration. It is used to address the basic requirements necessary to have an operable VTAM system.

Figure 2-2. VTAM Example Configuration



The example configuration in Figure 2-2 consists of:

- An IBM Processor
- Two non-SNA IBM 3278 Model 2 Display Stations attached via a 3274 Model 1B Control Unit to the processor byte multiplexor channel at addresses 024 and 025
- A 3286 Model 2 Printer at address 026
- A local 3274 Model 1A Control Unit with two SNA 3279 Model 2B Display Stations
- VTAM running under an operating system
- A VTAM application program.

Figure 2-2 is marked with information to be used during coding of the major nodes and start options. In this case, channel addresses for the non-SNA terminals, SNA 3274/3279 PU and LU addresses, PU type, and major and minor node names have been added. A user may wish to include other information such as physical locations, disk volume identifications, library names, and other notes relative to the environment.

Configuration Assumptions

For the example configuration, the following assumptions have been made in order to keep the number of considerations at a minimum:

- There is a single application program named INQUIRE. The APPLID in the program's access method control block (ACB) points to the name INQUIRE.
- The network is to be automatically activated at VTAM startup.
- The logical units are to be automatically logged on to the application program when activated.
- The application program contains the necessary coding to establish session protocols.
- There is to be VTAM password protection checking for the application program at OPEN ACB time. The password in the application program is RUOKAPPL.
- No VTAM tracing is to be done.
- The 3286 Printer is to be activated by the network operator.
- Two VTAM I/O buffers will hold the largest inbound PIU from any of the local terminals.

After developing a diagram of the configuration, the VTAM node definitions and start options may be coded. For this first example, there are five separate coding requirements:

- An application program major node
- A local non-SNA major node
- A local SNA major node
- A start option list (ATCSTR00)
- A configuration list (ATCCON00).

Note: Routing tables (PATH definitions) are necessary only when PIUs flow between two or more subareas. Therefore, since the first example involves only one VTAM host subarea, PATH definition sets are not required.

Application Program Major Node

For an application program major node, the VBUILD format is:

```
name      VBUILD  TYPE=APPL
```

Recall that a major node gets its name from the cataloged member name. Therefore, the statement name for VBUILD is optional although it may be of reference value to give it the same name that will later be used for the member name.

The application program's cataloged member name need not be the same as the APPLID/minor node name. Each application program in the major node is identified with an APPL statement. The general format for the APPL statement is:

```
name      APPL    optional keyword operands
```

The name field specifies a minor node name for the application program. Recall that the application program must OPEN an ACB before VTAM can establish any sessions for the program. The APPLID operand of the ACB points to the name that VTAM uses at OPEN time to associate the program with its minor node name. Thus, if the major node that contains the APPL statement for the program has been made active, VTAM can process requests for sessions.

Although there are several keyword operands for the APPL statement, only three, EAS = (estimated active sessions), AUTH = (authorization), and PRTCT = (protection/password), will be used for the example configuration; others will be discussed in later mini-courses.

EAS =

The EAS = operand specifies the approximate (estimated) number of concurrent sessions that the application program will have with its logical units (LU-LU sessions). Since the default for EAS is 404, a value should always be specified. The maximum number of concurrent sessions is 32,767. Base the estimate on the maximum number of logical units that could be in session with the program at one time.

With a large number of logical units and more than one application program, the number of sessions per program may vary considerably. A value too small for EAS = will not prevent session establishment for a greater number of sessions; however, VTAM will take additional execution time to service the additional session requests. Once the estimate is determined, the number should be increased by 10 percent to 20 percent and entered for EAS =.

For the example configuration, there are only five logical units (counting the non-SNA terminals) for the application, therefore, the maximum number of concurrent sessions is five (no real estimating here). The number is increased by one for the example:

EAS=6

AUTH =

The authorization parameter on the APPL statement tells VTAM that the program may issue certain VTAM macros during its execution. If the program attempts to issue the macro without the proper AUTH =, VTAM will reject the request.

Examples of authorization uses are:

- AUTH=(ACQ) Allows the program to request a session with a particular secondary terminal/LU. CICS/VS needs this authority when the DFHTCT parameter CONNECT= AUTO is coded for a particular CICS terminal.
- AUTH=PPO or SPO Allows the program to issue VTAM operator commands and receive VTAM messages (including unsolicited messages with PPO). An IBM program product, the Network Communications Control Facility (NCCF) needs this authority and a second authority AUTH=(PASS).
- AUTH=(PASS) Allows the program to end its session with a secondary LU and then pass that LU to another VTAM application program (or subtask) without operator intervention. NCCF needs this authority to pass initial sessions to its subtasks.

- AUTH=(CNM) Allows the program to use the VTAM communication network management (CNM) interface. An IBM program product, the Network Problem Determination Application (NPDA) needs this authority. (NPDA runs in conjunction with NCCF.)
- AUTH=TSO In MVS, authorizes a Time Sharing Option (TSO/VTAM) application program.

PRTCT=

The PRTCT= operand specifies a password for VTAM to compare at ACB OPEN time with a password coded in the application program. It is used to verify the authority of the application program to run with this particular APPL definition. Password protection is optional and no password checking is done when PRTCT= is not coded, regardless of an ACB password. The password may be any of 1 to 8 EBCDIC characters.

For the example configuration, the application program has a password of RUOKAPPL; therefore, it is coded in the APPL operand as:

```
PRTCT=RUOKAPPL
```

Other APPL Operands

Other APPL statement operands meet specific application program and operating system requirements. For example, SSCPFM and USSTAB only apply to program operator applications (AUTH=PPO or SPO) while MODETAB, DLOGMOD, and PARSESS apply only when the application will act as a secondary logical unit (SLU) in an application to application session. Refer to the VTAM Installation and Resource Definition manual for these and other operating system specific APPL operands.

Application Program Activation Sequence

The sequence of events necessary for VTAM to become aware of an application program are:

1. Start VTAM.
2. Activate the application program major node.
3. Start the application program which will issue an OPEN ACB request.
4. VTAM will retrieve the APPLID name from the program and search the application program major node for an equivalent name from an APPL statement.

5. VTAM will use information from the APPL statement and from the program's ACB to complete its control block representing the application program.
6. The program becomes ACTIV to VTAM and is ready for session establishment.

Sample Coding - Application Program Major Node

The following are the completed application program major node statements ready for cataloging in the VTAM definition library, with a member name of INQMAJ01:

```
INQMAJ01  VBUILD  TYPE=APPL
INQUIRE  APPL    EAS=6,PRTCT=RUOKAPPL
```

Any reference to this application program by the network operator will be by the APPL name INQUIRE. For example, the network operator might use the DISPLAY command to examine the status of the application program by keying:

```
d net,id=inquire
```

■ ■ ■

Please turn to Mini-Course 2, Exercise 2.1, in your PRG and answer the exercise questions.

Local Terminal Environment

The local environment for terminals running under VTAM has some distinct differences from the remote environment. Two of the major differences are:

- Local terminals are attached to a processor channel, a 4361 Loop Adapter or a 4361 Display/Printer Adapter.
- A communications controller or a communications adapter are not used with local terminals or local cluster controllers.

Two general categories of local terminals are supported by VTAM: local non-SNA devices and local SNA devices.

The local non-SNA devices supported are:

- The 3272 Control Unit (Models 1 and 2), which supports up to 32 Display Stations and Printers.
- The 3274 Control Unit (B and D Models), which supports up to 32 devices: 3278 Display Stations, 3279 Color Display Stations, 3287 Printers and 3289 Line Printers. Or, the 3274 Control unit which can support up to sixteen 3277 Display Stations, 3284/3286/3287 Printers, or 3288 Line Printers.

The local SNA devices supported are:

- The 3790 Communication System
- The 3730 Distributed Office Communication System
- The 3270 Information Display System including the 3274 Control Unit (A Models), 3279 Color Display Stations, 3278 Display Stations, and 32xx Printers.

The local environment is defined to VTAM according to the devices attached to the system; that is, VTAM requires two separate major node device definitions:

- Local non-SNA devices
- Local SNA devices.

Local non-SNA

Local non-SNA major nodes are defined with VTAM macros as follows:

- Each local non-SNA major node is defined with an LBUILD statement.
- Each minor node representing a terminal is defined with a LOCAL statement.

The operands of the LOCAL definition statement are: CUADDR, TERM, FEATUR2, DLOGMOD, ISTATUS, LOGAPPL, LOGTAB, MODETAB, and USSTAB.

Local 3270 lists are defined for convenience. For example, a definition list might consist of all 3284/3286 printer definitions, or of all 3270 devices attached to the same 3272 local control unit, or any other desired grouping. There is no relationship between hardware configuration and the way local non-SNA 3270 definition lists are coded; that is, each LOCAL statement describes a single terminal (by channel address). Therefore, the terminals may be placed in one or more LBUILD major nodes as required.

LOCAL Statement

Each terminal in the local non-SNA major node is defined with a single LOCAL definition statement. The LOCAL statement operands that define the physical terminals are: CUADDR, TERM, and FEATUR2.

CUADDR

The CUADDR = cua operand indicates the channel unit address for the terminal. Unlike SNA devices attached to a cluster controller (to be described below), each non-SNA terminal has its own subchannel address in its own definition. For example, if 10 non-SNA 3270 Display Stations are to be controlled by VTAM, there will be 10 LOCAL statements, each having a CUADDR operand. The number of 3270 Control Units does not appear in the VTAM definition.

Non-SNA 3270 control units may have up to 16 devices attached if the control unit address is an odd number. The channel address byte has 4 bits for the control unit and 4 bits for the device.

Up to 32 devices may be attached to a control unit with an even numbered address. The channel address byte has 3 bits for the control unit and 5 bits for the device address.

TERM = 3277/3284/3286

The TERM operand of the LOCAL statement represents the device type. The type is not necessarily the same as the actual device type number. For example, all 3277/3278/3279 Display Stations are specified as 3277; all 3286/3287/3288 Printers are specified as 3286; 3284 Printers are specified as 3284.

FEATUR2 =

The FEATUR2 operand specifies the 3270 screen (or buffer) size; MODEL1 for 480 byte screens and MODEL2 for 1920 byte screens. If the terminal has the extended data stream feature installed, it is specified as a FEATUR2 value of EDATS (NOEDATS is the default).

Note: If more than one of the FEATUR2 operand values are coded, they are enclosed in parentheses.

The other parameters (DLOGMOD, ISTATUS, LOGAPPL, LOGTAB, MODETAB, and USSTAB) that may be used on a LOCAL definition statement are common parameters to other VTAM major nodes including NCP definitions. These VTAM-only parameters will be discussed later in the course.

Local SNA Major Nodes

The local SNA major nodes use the following macro statements:

- Each local SNA major node is defined with a VBUILD TYPE=LOCAL statement.
- Each minor node is described with PU and LU statements.

Local SNA devices are attached to cluster control units and are addressed by control unit position (port) rather than channel addresses (unlike local non-SNA terminals described above).

PU Notes

The PU statement defines the control unit with its channel device name specified in the CUADDR operand. The address format is the same as the physical address defined to the operating system; that is, the channel and the control unit address on the channel are defined as three hexadecimal characters. For example: a control unit installed on a multiplexer channel at address 080 would be specified as CUADDR=080 (no X or quotes).

PU statement operands that further define local SNA physical units are PUTYPE, MAXBFUR, and DISCNT.

PUTYPE

PUTYPE=2 is the default value since only type 2 local SNA physical units are supported by VTAM.

MAXBFRU

Before VTAM can read from a local LU, VTAM needs to have a number of input buffers available in the I/O buffer area (IOBUF in MVS and VM or LFBUF in VSE). The MAXBFRU count specifies how many of these buffers VTAM will try to fill when receiving data from this PU. If the specified number of buffers is not available, the read is postponed until that number is available. If the number is too large, VTAM will not use the extra space. If the length of one buffer is large enough to hold all the data which the local SNA device can transmit in a single PIU, then code MAXBFRU = 1 (1 is the default). If the buffers are not that large, then code MAXBFRU as large as necessary to permit acceptance of the longest possible message from the local device.

DISCNT

The DISCNT = YES operand tells VTAM to disconnect the PU and all of its LUs when the last LU-LU session completes. In this case, the PU and LUs are still active to VTAM; however, any new sessions must be established from the host end of the session. For example, an application program might request a session with one or more of the LUs and VTAM would then reestablish the SSCP-PU and SSCP-LU sessions before processing the LU-LU session request.

Coding DISCNT = NO tells VTAM to keep the SSCP sessions although the last LU-LU session has terminated. VTAM would then use one of several other session termination requests to end the SSCP sessions.

The normal procedure for deactivating a PU and its LUs is for the network operator to enter a VARY INACT command to the PU when all device sessions have completed. This breaks both the SSCP-PU and SSCP-LU sessions and marks them inactive to VTAM. An alternative is to leave all devices active until the major node is inactivated or VTAM itself is terminated. Code DISCNT = NO unless there is a specific reason to have SSCP sessions terminated.

LU Notes

The LU statement, as with all SNA LU definitions, defines the local terminal address (LOCADDR), that is, the address of the LU as it is known to the control unit. LU statements follow the PU statement in ascending LOCADDR order. For example, the first LU device address is 2 (port number 0) on an SNA 3274 Control Unit; it would be coded as LOCADDR=2 on the first LU statement followed by additional LU statements as needed.

Additional LU operands for local SNA devices are: DLOGMOD, LOGAPPL, LOGTAB, MODETAB, USSTAB, PACING, SSCPFM, VPACING, and ISTATUS. These operands are used as needed to define the VTAM procedural options for each logical unit.

Major Node Activation

Activation of the local major nodes occur either automatically at VTAM initialization time (via a start option) or by the network operator with the VARY ACT command. The activation sequence is always as follows:

1. The major node
2. The PU minor node
3. The LU minor node.

Application programs request sessions with local SNA and non-SNA terminals/LUs by using the VTAM minor node names. After the session is established, the application program sees the terminal/LU in the same way that it sees any other resource.

Example Configuration

Coding the Local Major Nodes

Figure 2-2 and the assumptions for the Example Configuration given in the first section of this mini-course may be used to create the coding of the two local major nodes.

The Local non-SNA Major Node

The VTAM definition for the non-SNA major node differs very little from other major nodes. With the information at hand, such as the assumptions above, the coding of the major node is primarily a matter of selecting operand values.

The names are chosen so that an operator may easily remember them when using VTAM commands. The definition library member name for the example non-SNA major node will be LOC1NSNA. The LOCAL terminal names (minor node names) will be LOC24N78, LOC25N78, and LOC26N86.

The Local SNA Major Node

The local SNA 3270 Control Unit and Display Stations are to be coded in a VBUILD TYPE=LOCAL major node with the name LOC2SNA. The control unit at address 0B8 is to have a minor node name of PU0B8S74. The two 3279s are to be named LU01S79 and LU02S79.

The Sample Local Major Nodes

Based on the requirements and assumptions given earlier, an example of coding of the local major nodes for the example configuration is shown below. The local non-SNA major node LOC1NSNA is shown in Figure 2-3. The local SNA major node LOC2SNA is shown in Figure 2-4.

LOC1NSNA	LBUILD		CATALOG AS LOC1NSNA	
*				
LOC24N78	LOCAL	CUADDR=024, TERM=3277, ISTATUS=ACTIVE, FEATUR2=(MODEL2), LOGAPPL=INQUIRE	1ST 3278 DEFAULT REQMT #3	x x x x
*				
LOC25N78	LOCAL	CUADDR=025, TERM=3277, FEATUR2=(MODEL2), LOGAPPL=INQUIRE	2ND 3278 REQMT #3	x x x
*				
LOC26N86	LOCAL	CUADDR=026, ISTATUS=INACTIVE, FEATUR2=(MODEL2), LOGAPPL=INQUIRE, TERM=3286	3286 PRINTER REQMT #7 1920 BUFFER	x x x x

Figure 2-3. Example Local Non-SNA Major Node

LOC2SNA	VBUILD	TYPE=LOCAL	CATALOG AS LOC2SNA	
*				
PU0B8S74	PU	CUADDR=0B8, DISCNT=NO, MAXBFUR=2, PUTYPE=2, LOGAPPL=INQUIRE	3274 CONTROL UNIT DEFAULT REQMT #8 DEFAULT FOR ALL LUs	x x x x
*				
LU01S79	LU	LOCADDR=2	3274 PORT 0	
LU02S79	LU	LOCADDR=3	3274 PORT 1	

Figure 2-4. Example Local SNA Major Node



Please turn to Mini-Course 2, Exercise 2.2 in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 3

Basic VTAM Installation Requirements — Start Options

Mini-Course 3. Basic VTAM Installation Requirements - Start Options

Introduction

Mini-Course 2 introduced VTAM installation requirements including definition lists necessary for a workable local terminal configuration. In this mini-course, start options and a sample configuration list will be described.

Start Options

Start Option List

At VTAM startup time, the user must supply information to define the initial domain configuration and select optional VTAM facilities. VTAM expects to find a start options list named ATCSTR00 in its definition library at startup time. All options may be supplied in the list, or they may be merged with overriding options filed in additional ATCSTRyy (yy = two alphanumeric characters) members or books.

Start options may also be entered by the network operator if prompting is defined in ATCSTR00. A LIST option allows the network operator to point to any one of the ATCSTRyy lists. For example,

```
LIST=12 points to ATCSTR12
```

Thus, each time VTAM is started, a variation of options may be selected.

VTAM selects start options from one of four possible places in a hierarchical order. If a start option appears in more than one place, VTAM will select the overriding option in the following order:

1. A start option entered by the network operator (PROMPT).
2. A start option coded in a ATCSTRyy list (via LIST = yy).
3. A start option coded in the required ATCSTR00 list.
4. If none of the above, from an internal default list.

Selected Start Options

SSCPID

The only required start option is the SSCPID number; VTAM cannot be started without it. The SSCPID number is sent to physical devices at activation time. The SSCPID is also used in cross-domain communications as part of session establishment. A physical unit or a cross-domain resource manager knows which SSCP it is in session with via the SSCPID. It must be a unique number for each domain in a multidomain system and is specified as a decimal number from 0 to 65535.

For example, an IBM 8100 can have a user coded list of SSCPID numbers which represents valid VTAM hosts. The SSCPID number, coded as a VTAM start option, is sent to the 8100 for a check against the list. If the 8100 finds a match of SSCPID numbers, then sessions may be initiated; otherwise, the session setup fails.

HOSTSA and Subarea Addressing

Unique subarea numbers are required throughout a multisystem environment. Each host processor and each communications controller (37xx) will have its own subarea number that must be different if there is to be cross-domain communication. The HOSTSA start option defines the host VTAM subarea number.

Some special considerations apply when nodes that use extended network addressing (ENA) coexist in the same network with nodes that do not.

Extended network addressing (introduced with NCP V4 and VTAM V3 for MVS and VSE) increase the size of effective network addresses from 16 to 23 bits. The extended address is split into an 8-bit subarea and a 15-bit element field. A network using this address structure can thus have up to 255 subareas (subareas 1 through 255, with subarea 0 reserved) and 32,768 elements.

Prior to VTAM V3 and NCP V4 network addresses are 16 bits in length, and the split between subarea and element is user-defined. MAXSUBA, the VTAM start option and an NCP operand define the split by specifying the highest possible subarea number. A network that specifies a MAXSUBA of 31, for example, defines a 5-bit subarea and 11-bit element.

Non-ENA nodes continue to use this structure. They can, however, participate in an ENA network under the following conditions:

- A compatibility PTF must be installed on all non-ENA VTAM V2s that communicate with ENA nodes. VTAM version 1 does not support ENA.
- All nodes that communicate with non-ENA nodes must define a MAXSUBA. ENA nodes need this information in order to decode 16-bit network addresses. The value specified for MAXSUBA must be the same at all nodes, ENA and non-ENA.

Non-ENA VTAMs cannot communicate with any node whose subarea is greater than the MAXSUBA.

Non-ENA VTAMs cannot communicate with any logical unit whose element address exceeds the maximum element number defined by the split between subarea and element.

HOSTPU

VTAM's physical unit services component is used to communicate to other hosts (PU__T5s) or NCPs (PU__T4s). The name for the PU component is ISTOPUS.

The HOSTPU start option may be coded to change the name ISTOPUS to a name unique in each host VTAM.

In multisystem and multinet network environments coding a HOSTPU name will help to avoid confusion with certain VTAM displays and Network Logical Data Managers (NLDM) screens. Including the host subarea number in the name is recommended.

ITLIM

The initialization/termination limit (ITLIM) start option specifies the number of concurrent session-initiation, session termination, and USS command requests that VTAM can process. When requests exceed the limit, they are queued for processing after completion of the current requests.

VTAM allocates storage for all requests as they are received, up to the ITLIM limit. Cross-domain requests, however, are not affected by ITLIM and are handled immediately. In networks with a large number of terminals, the overhead at VTAM startup can be significant since VTAM will allocate storage for all pending requests before processing new requests.

By limiting the requests VTAM has to handle at one time, storage allocation can be spread over a longer period and the total amount of storage needed can be reduced.

The ITLIM value is a performance option for systems with a large number of terminals and should be considered when VTAM initialization is slower than expected. In smaller networks the ITLIM value may be allowed to default to zero which indicates all requests will be handled as received.

MAXAPPL (Pre-VTAM Version 3)

The MAXAPPL start option tells VTAM to allocate control blocks for a maximum number of concurrently active user application programs. MAXAPPL=10 is the default for user VTAM application programs. VTAM needs five additional control blocks for its own use.

Each control block is given an element address at VTAM initialization. For example, there are 15 element addresses for application programs when the user default is taken (10 for MAXAPPL, and 5 for VTAM).

VTAM assigns an element address for each control block up to the chosen value of MAXAPPL. Since the MAXSUBA value determines the maximum possible number of element addresses, MAXAPPL should be defined with consideration that there are element addresses left for any local devices.

For example:

If MAXSUBA=255, there are 255 element addresses available.

MAXAPPL could equal 250

8 bits for subarea numbers 1-255 from MAXSUBA
8 remaining bits for:
250 element addresses for MAXAPPL
5 element addresses for VTAM
0 element addresses for local devices

VTAM would set up 250 control blocks for application programs (a waste of storage) and there would be no possibility of using local devices. A more reasonable example:

If MAXSUBA=255 and there will be a maximum of 7 VTAM application programs running concurrently, MAXAPPL could be set as low as MAXAPPL=7.

8 bits for subarea numbers 1-255
8 remaining bits for:
7 element addresses for programs
5 element addresses for VTAM
243 element addresses left for local devices

Note: If the example were in a VSE system with a communications adapter installed, the 243 element addresses would need to include all CA devices as well as any local devices.

VTAMEAS (Pre-VTAM Version 3)

This is the estimated number of active network addressable units (NAUs) and local non-SNA terminals, as opposed to concurrent sessions. This estimate is used by VTAM in a lookup scheme to represent sessions with network addressable units (NAUs). It is similar to the EAS value for application programs; however, the EAS value on the APPL statement applies to LU-LU sessions only. A reasonable method for choosing a value for VTAMEAS is to count the number of PUs and LUs in local and NCP major nodes and the number of local non-SNA terminals, and increase the number by 10 to 20 percent.

If VTAMEAS is set too low, the table can not reflect all NAUs, and the VTAM sessions may experience a degradation in performance.

Note: In VTAM V3, the defaults for EAS are 3000 for MVS and 50 for VSE. These defaults are stored in the VTAM constants module ISTRACON and are modifiable by the user. See the VTAM customization manual for reference to the constants module.

```
* IN MVS - CATALOG AS MEMBER ATCSTROO IN SYS1.VTAMLST
* IN VSE - CATALOG AS ATCSTROO IN THE B LIBRARY
* IN VM - FILE IN CMS AS 'ATCSTROO VTAMLST'
*
NOPROMPT,                                x
SSCPID=101,                               required x
HOSTSA=1,                                 default  x
CONFIG=01,                                x
HOSTPU=VTAM01,                             x
SUPP=NOSUP                                 default
*
*
```

Figure 3-1. Sample Start Options

Configuration List

After reading the start options list, VTAM looks for a configuration list by the name of ATCCON00 in order to activate selected major nodes for the initial configuration. The list contains the names of the major nodes (and PATH definition sets) that the user wants to be automatically activated at VTAM startup. A number of configuration lists may be filed as ATCCONxx (xx = any one or two alphameric characters) in the definition library. Thus, a user may describe several different configurations and then select a particular one at startup, either by operator prompting or by a pointer (LIST=xx) in the start options list.

Note: The user may elect to file a null configuration list (ATCCON00) and therefore have all nodes activated by the operator VARY command. If VTAM cannot find a null configuration list ATCCON00, a warning message is issued to the network operator.

There is no VTAM default configuration list, and only one list may be used with each VTAM initialization. Therefore, each list must include all major node names and PATH definition set names which are to be automatically activated. PATH definition set names must precede any related NCP major node names.

Example Configuration List

The default configuration list ATCCON01 pointed to by the CONFIG=01 operand of ATCSTR00 contains the major node names that are to be activated at VTAM initialization.

The configuration list for the example, which required that the network be automatically activated, is coded as shown in Figure 3-2.

```
* IN MVS - CATALOG AS ATCCON01 IN SYS1.VTAMLST
* IN VSE - CATALOG AS ATCCON01 IN B LIBRARY
* IN VM - FILE IN CMS AS 'ATCCON01 VTAMLST'
*

INQMAJ01,LOC1NSNA,LOC2SNA

*
```

Figure 3-2. Example Configuration List

Note that the coding format for configuration lists is different than for major nodes since the assembler language format is not used (except for the continuation column 72). It is simply a list of major node names (and/or PATH definition set names) separated by commas.

VTAM will activate the major nodes in the sequence coded in the configuration list. One way for VTAM to automatically activate a different list is to change the CONFIG=xx start option to point to a different list.

Major & Minor Node Activation

Before a program can successfully open an ACB, the internal program name must be defined to VTAM in the application program major node, and that major node must be made active. When the application program major node becomes active (either automatically at VTAM startup or later on by the operator VARY command), the major node name and all the application program names defined in the major node definition become available to VTAM.

There is a difference between activating a terminal minor node and an VTAM application program minor node. The terminal minor node is activated when VTAM sends an ACTPU and ACTLU to the terminal, establishing an SSCP-PU and SSCP-LU session. The VTAM application program is activated when the program OPENS its ACB. Prior to the OPEN, VTAM maintains all APPL definitions with a status of connectable (CONCT). The status changes from CONCT to ACTIV after a successful OPEN ACB.

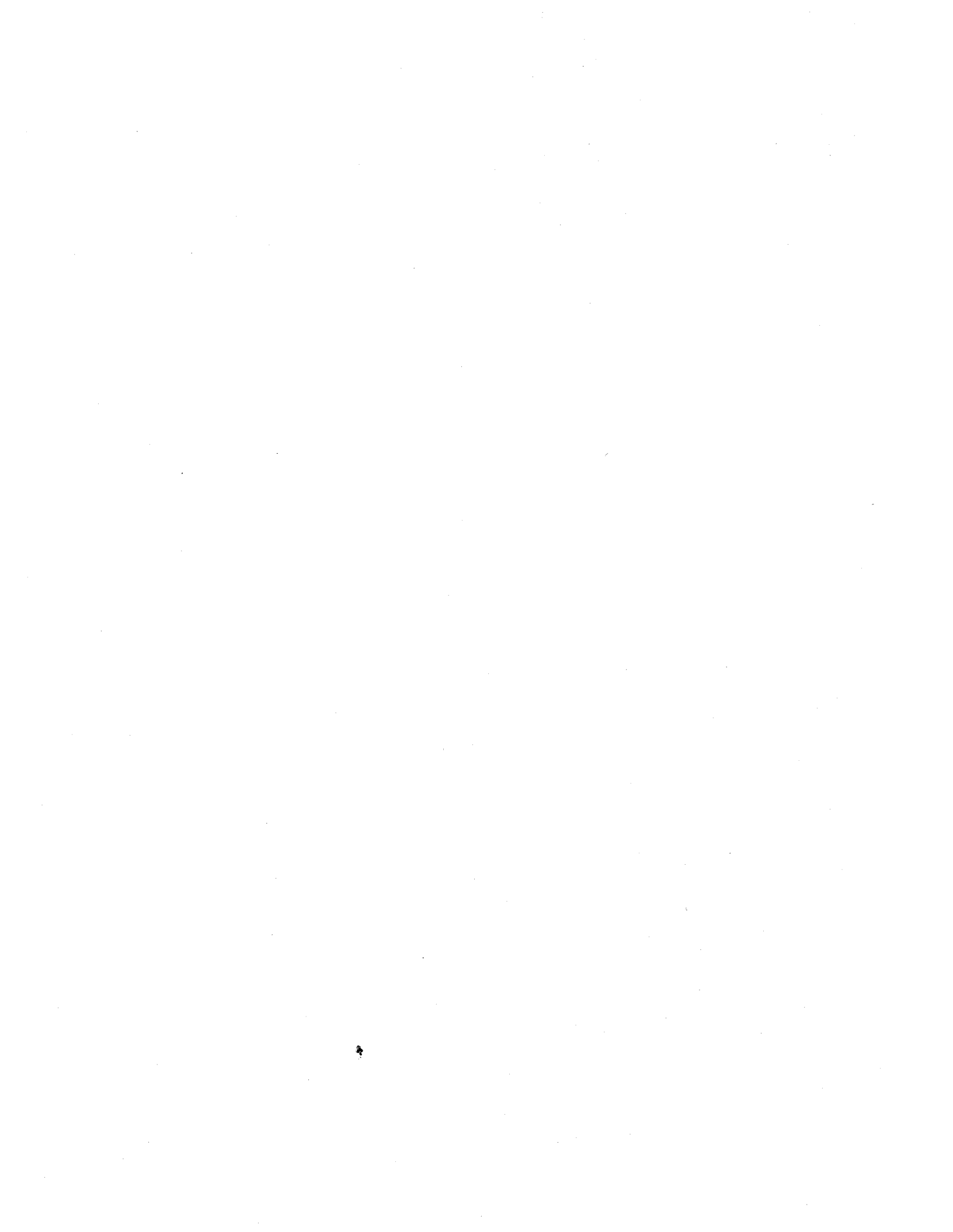
■ ■ ■

Please turn to Mini-Course 3, Exercise 3.1 in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 4

VTAM Considerations in MVS



Mini-Course 4. VTAM considerations in MVS

Introduction

The MVS System Modification Program (SMP) is used to install VTAM from the distribution media. Installation requirements and coding procedures are found in the *Program Directories* and *Memo to Users* which come with VTAM.

The MVS release levels required for VTAM Version 3 are one of the following:

- MVS/System Product Version 1 (with JES2 or JES3) for System/370
- MVS/System Product Version 2 (with JES2 or JES3) for Extended Architecture (XA) processors

MVS Generation

VTAM is included in MVS by specifying it in the DATAMGT generation macro as follows:

```
ACSMETH=VTAM and/or IND=YES
```

Other MVS generation macros to consider are:

DATASET The DATASET macro defines all system data sets that will be used by VTAM, NCP, and terminal subsystems. (The definitions can be made with IEHPROGM instead of the DATASET macro.)

IODEVICE Specifies all channel-attached devices including locally attached cluster controllers and communications controllers. The APFLIB parameter must specify SYS1.VTAMLIB information.

Note: MVS does not need to know about any remote devices that VTAM has in its domain; they are not specified with IODEVICE statements. Addressing for remote communications devices is accomplished entirely within the hardware and software of the VTAM and NCP system.

SECONSOLE If the multiple console support option is chosen, VTAM requires routing codes of 1, 2, and 8 and command codes of 1 and 2. Routing codes of 4, 6, and 10 may also be used by VTAM.

CTRLPROG The CSA value must be at least 800.

MVS VTAM Data Sets

When VTAM is installed in MVS, the data sets required fall into four categories:

- Required system data sets for VTAM modules and routines
- Optional data sets depending on user choices
- VTAM required data sets for user definitions and runtime libraries
- Special VTAM data sets.

Required System Data Sets

The required system data sets and contents for VTAM are:

SYS1.LINKLIB	VTAM initialization module
SYS1.LPALIB	VTAM load modules and user written exit routines
SYS1.NUCLEUS	VTAM resident SVCs and abnormal termination routines
SYS1.LOGREC	VTAM error records
SYS1.PARMLIB	VTAM related information
SYS1.PROCLIB	Will include one or more VTAM start procedures

Optional System Data Sets

Optional data sets depend on user choices for:

- VTAM trace facility
- VTAM non-resident SVCs
- Supervisor call (SVC) dumps.

The optional data sets and contents are:

SYS1.TRACE Generalized Trace Facility (GTF) trace records for VTAM

SYS1.SVCLIB VTAM non-resident SVCs and local device error recovery procedures (ERPs)

SYS1.DUMP SVC dump records

VTAM Runtime Data Sets

Data sets used by VTAM during execution are a combination of system data sets and user defined VTAM data sets.

The start procedure for VTAM will always require two data definition (DD) names for VTAM:

- VTAMLIB
- VTAMLST

The data sets and their uses are as follows:

SYS1.VTAMLIB

VTAMLIB is the VTAM load module and tables library.

VTAMLIB receives VTAM load modules when VTAM is installed. These include VTAM executable modules and supplied tables. The supplied tables are:

- A logon mode table (ISTINCLM)
- A USS table (ISTINCDT)
- A VTAM network operator message and command table (ISTINCNO)
- A VTAM constants table (ISTRACON).

In addition, one or more of the following user-coded VTAM tables may be link-edited into VTAMLIB:

- USS tables
- Logon mode tables
- Logon interpret tables
- A class of service (COS) table (ISTSDCOS)
- A communication network management (CNM) routing table (ISTMGC00).

There may be any number of USS logon, interpret, and logon mode tables in the library. But only one COS table or CNM routing table will be used by VTAM.

SYS1.VTAMLST

VTAMLST is the VTAM definition library.

Members of the definition library include user coded:

- Major nodes
- Start option lists
- Configuration lists
- VTAM PATH definition sets

All members of VTAMLST are cataloged in source statement form. VTAM creates internal tables from the source code for start option and configuration lists at VTAM initialization time. Major nodes and path tables are converted from source to internal tables when they are activated.

Note: VTAM PATH definition sets are not major nodes.

SYS1.VTAMOBJ (Reference for Pre-ACF/VTAM Version 3)

VTAMOBJ is a special data set used in MVS VTAM systems prior to ACF/VTAM Version 3.

When a major node is activated, the member is selected from VTAMLST and VTAM builds an internal table called a resource definition table (RDT). A copy of this table is then placed on VTAMOBJ. Later, if the major node is inactivated and reactivated, VTAM will look at VTAMOBJ for a saved RDT with the same name and use the copy rather than reassemble the RDT.

Note: When VTAM path tables are activated, VTAM builds an internal path table which is not an RDT. Also, path tables are not placed on VTAMOBJ.

IMPORTANT: VTAM looks at VTAMOBJ depending on the VTAMLST data set.

1. If VTAMLST is not a concatenated data set:
 - a. VTAM will compare time stamps of the RDT member on VTAMOBJ with the major node on VTAMLST.
 - b. If the VTAMLST major node time stamp is later than that of the RDT on VTAMOBJ, VTAM will assume that the major node definition has been changed since the copy was placed on VTAMOBJ.

VTAM will rebuild the internal RDT from the VTAMLST member.
 - c. If the time stamps are the same, the VTAMOBJ copy is used.
2. If VTAMLST is a concatenated data set:
 - a. VTAM does no time stamp checking.
 - b. The RDT copy from VTAMOBJ is used regardless of changes to the major node definition on VTAMLST.

It is recommended that VTAMLST not be a concatenated data set for the above reasons. If VTAMLST is made a concatenated data set, the user should make sure that the RDT copy on VTAMOBJ is scratched when changes are made to the major node on VTAMLST.

Special VTAM Data Sets

There are three additional data sets that receive VTAM code when VTAM is installed:

- SYS1.MACLIB** The MACLIB data set contains the VTAM macros required to assemble VTAM application programs.
- SYS1.ASAMPLIB** The ASAMPLIB contains the source code for the VTAM-supplied operator command table (operation-level USS table). The code is supplied for user information. The assembled and link-edited table (ISTINCNO) resides on VTAMLIB after VTAM is installed.
- SYS1.SAMPLIB** The SAMPLIB data set contains the operator command table in source format that allows alterations by the user before VTAM installation. Once VTAM is installed, SYS1.SAMPLIB is no longer required.

Configuration Restart Data Sets

Two other special data sets may be defined if the user selects the VTAM configuration restart facility. A configuration restart data set is used to store the status of minor nodes (active or inactive) when there is a network failure. Configuration restart is selected for each major node; therefore, a separate user named data set can be assigned on a major node basis.

In addition to the individual configuration restart data sets, the user may define a data set, SYS1.NODELST, in which VTAM will keep the status of all major nodes. When VTAM is restarted, the operator may choose to have VTAM restore major nodes listed in SYS1.NODELST to their original status.

MVS Data Sets for NCP

There are four data sets for NCP installation:

- A user named NCP Load Library
- SYS1.VTAMLST
- SYS1.LINKLIB
- An additional data set is needed if the user selects the option to have VTAM provide an NCP storage dump.

NCP Load Library

When NCP is installed in an MVS system, the output of the NCP generation is an NCP load module which is placed on a user named NCP Load Library. VTAM finds the data definition statements by looking at the LOADLIB parameters in the NCP BUILD macro when the NCP is activated.

The output of NCP generation consists of two types of NCP members:

- The NCP load module
- An NCP Resource Resolution Table (RRT)

The member name of the NCP is defined in the NEWNAME operand of the NCP BUILD macro. The RRT member created has the same name as the NCP appended with an "R." For example, NCP11's RRT would be NCP11R.

SYS1.VTAMLST

The NCP generation deck (source code) is placed on VTAMLST and becomes the VTAM major node definition for the NCP. When the NCP major node is activated, VTAM gets the major node definition from VTAMLST and the NCP RRT from the NCP Load Library.

VTAM builds its internal table (RDT) for the NCP major node by adding RRT information to the NCP major node definition. The RRT contains resource IDs for each network addressable unit (NAU) in the NCP (PUs and LUs, for example). After VTAM matches RRT names with minor node names, the NAU resource IDs correspond to VTAM minor node element addresses in the RDT.

Note: The element address and NCP subarea number are used by VTAM when it creates the destination address fields in the PIU transmission header (TH) for particular PUs and LUs.

The name of the NCP in the NCP Load Library must match the member name of the NCP major node on VTAMLST.

SYS1.LINKLIB

The third data set used with NCP is SYS1.LINKLIB. Four NCP programs are placed in SYS1.LINKLIB when NCP is installed:

- The NCP loader utility program
- A dump utility program for NCP
- A dump bootstrap program for NCP
- NCP testing modules for initialization testing.

VTAM will use the NCP loader utility program to load an NCP load module from the NCP Load Library into the 37xx communications controller.

Note: The NCP loader utility program may also be run as a stand-alone job to load an NCP; that is, VTAM need not be running.

VTAM will load the initial test module from SYS1.LINKLIB and an NCP diagnostic routine from a user-defined Initial Test Routine data set into a channel-attached 37xx before loading the NCP. This is a user option selected when the NCP is coded.

NCP Dump Data Set

The user may select the option to have VTAM dump NCP storage automatically or by network operator request. A user-named NCP dump data set must be supplied to receive the NCP dump records.

VTAM will use the NCP dump bootstrap program to obtain the records and store them on the dump data set. The NCP dump utility program may then be used to format and print the NCP dump records.

MVS Start Procedure

The VTAM start procedure for MVS includes an EXEC statement for the VTAM initialization module and data set definitions for the required and optional data sets. VTAM start procedures are cataloged in SYS1.PROCLIB.

An example of the EXEC statement in the start procedure is:

```
//STEP1 EXEC PGM=ISTINM01,REGION=....
```

where ISTINM01 is the member name on SYS1.LINKLIB for VTAM's initialization module.

Naming the VTAM Start Procedure

Any name may be chosen for the VTAM start procedure; however, in some cases it may be advantageous to use the name NET for the procedure.

The operator commands DISPLAY, VARY, and HALT all require the name NET to direct the command to VTAM. The MODIFY command requires the VTAM procedure name rather than NET.

If only one VTAM start procedure will be used, then choosing the VTAM procname of NET would make the command names the same for all VTAM operator commands.

When more than one VTAM procedure will be used, different names must be used for all of the procedures. Thus, NET might be chosen for the normal VTAM production procedure and other names might be chosen for VTAM testing procedures. The VTAM operator would then use the appropriate name in the MODIFY command.

The Example Configuration

The VTAM library members for the example configuration introduced in earlier mini-courses are as follows:

VTAMLST

The members in VTAMLST are:

- The VTAM start options list ATCSTR00
- The initial configuration list ATCCON01
- The major nodes:

INQMAJ01
LOC2SNA
LOC1NSNA

VTAMLIB

The installed VTAM modules and tables.

SYS1.PROCLIB

At this point, there are no NCPs in the configuration and no user options have been chosen which would require data set definitions. Therefore, the VTAM start procedure will be cataloged in SYS1.PROCLIB as shown in Figure 4-1.

```

// * EXEC ISTEINM01 FOR MVS
// *
// VTAMP PROC
// STEP1 EXEC PGM=ISTEINM01,REGION=3000K,DPRTY=(15,15),PERFORM=8,
// TIME=1440
// VTAMLIB DD DSN=SYS1.VTAMLIB,DISP=SHR
// VTAMLST DD DSN=SYS1.VTAMLST,DISP=SHR,LABEL=RETPD=0

```

Figure 4-1. Example Configuration MVS Start Procedure

Starting the System

The VTAM example configuration may now be started with the MVS START command:

Either

```
S VTAMP
```

or

```
S VTAMP,,, (overriding start options)
```

VTAM's initialization module will load the required VTAM modules first. Next, ATCSTR00 is loaded from VTAMLST. The start options specified for the system are examined, and these options are used to establish various operating characteristics for the VTAM domain.

VTAM modules are loaded into the VTAM address space, storage is obtained for various control tables, and the tables are initialized. The buffer pools controlled by VTAM's storage management routines are built, VTAM's queues are initialized, and VTAM files are opened. The configuration list ATCCON01 is read from VTAMLST and the RDTs are built for the three major nodes: INQMAJ01, LOC1NSNA, and LOC2SNA.

VTAM will then issue the console message:

```
"IST020I VTAM INITIALIZATION COMPLETE"
```

After this processing has been completed, VTAM is ready to accept VTAM commands.

■ ■ ■

Please turn to Mini-Course 4, Exercise 4.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 5

VTAM Considerations in VM

Mini-Course 5. VTAM Considerations in VM Installation

Introduction

ACF/VTAM Version 3 for VM/SP provides a native interface to VM, allowing connected terminal devices to act as virtual machine operator consoles. With the VM/Group Control System (GCS), a facility provided in VM/SP Release 4, VTAM can support end-user access, terminal-to-application access, and network operations in an SNA environment. In addition, application-to-application access is provided in a multisystem environment.

VM/Group Control System (GCS)

GCS provides MVS supervisor and data management services with a simulated subset of MVS. Also, selected VM services and conversational monitor system (CMS) functions are provided in GCS. Each virtual machine in the Group Control System machine group can communicate with other machines in the group through common storage.

VTAM must be installed in a virtual machine that is a member of a GCS. Any application program which accesses the VTAM application program interface (API), such as NCCF Version 2, must reside in the same GCS group as VTAM. Only one ACF/VTAM Version 3 may reside in a GCS machine group; however, multiple GCS machine groups may run concurrently in the same VM processor, each having a virtual machine with a VTAM domain installed. Cross-domain communication is possible between the VTAM domains using either a VM virtual channel-to-channel adapter (CTCA) or through a channel-attached communications controller.

VM Environment

Figure 5-1 shows VTAM in a VM environment along with other optional and required virtual machine applications.

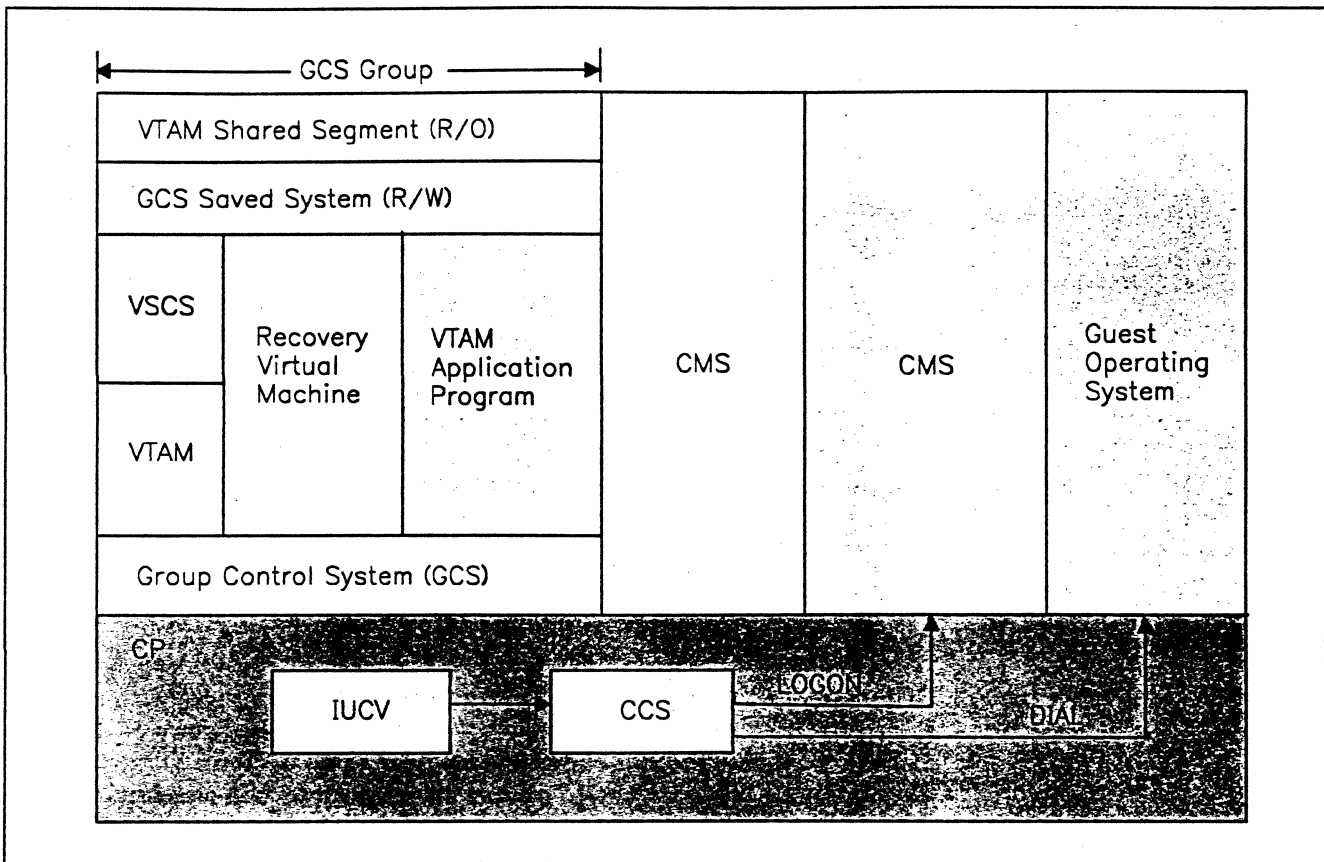


Figure 5-1. VTAM in a VM Environment

VSCS shown in Figure 5-1 is the VM SNA Console Support component of VTAM. It is supplied with VTAM and allows terminals in an SNA network to act as virtual machine operator consoles. For example, a terminal user may first logon to the VTAM application program, and later, logon to CMS as a virtual machine operator, and still later, use the CP DIAL command to connect to the guest operating system as a local terminal.

Communication and data movement between virtual machines or between a virtual machine and a VM control program (CP) service is accomplished through the Inter-User-Communication-Vehicle (IUCV) shown in Figure 5-1. The IUCV is the interface between the console support of VTAM, VSCS, and CCS, a communication component of CP. The console communication services (CCS) component of CP provides the interface between IUCV and CP. One other CP component is necessary for data movement between GCS group members: the CP Signal System Services.

For example, a data movement may begin with an SNA terminal in the network transmitting data destined for the CMS virtual machine. VTAM's VSCS receives the data from the LU through the VTAM API (Application Program Interface) with the use of CP Signal System Services. VSCS then sends the data to CCS through the IUCV component. CCS then routes the data to the CMS virtual machine. In turn, data from CMS back to the LU would retrace the same path.

If, in the example, the LU were in session with the VTAM application program shown in Figure 5-1, VSCS and CCS would not be involved and data would flow through the VTAM API to and from the application program.

CP-owned 3270 non-SNA and remote BSC devices, however, may directly access the VTAM virtual machine and logon to VTAM application programs. These devices would use the CP DIAL command to access VTAM. In order to have this capability, VTAM needs to have a set of nonexistent addresses defined in a LOCAL major node. Also, the CP DEF GRAF command would be used to assign virtual terminals to the non-existing addresses.

The recovery virtual machine shown in Figure 5-1 is responsible for executing the initialization and termination routines for all members in the group. This includes normal termination and abnormal termination with error recovery via group termination exits.

Pre-Installation for VTAM

The program directory shipped with VTAM includes directions for installing VTAM from the product tapes. The memo to users which contains detail installation instructions is printed from the tape by entering the MAINT command and userid.

Pre-installation tasks include standard VM functions as follows:

1. VM directory update
 - Add userids for VTAM, the recovery virtual machine, and other virtual machines such as NCCF with the USER control command.
 - Add minidisks for VTAM and other SNA products such as NCP, SSP, NCCF, NPDA, and NLDM with the MDISK control command.

- Consider using the option DIAG98 for VTAM. It will allow direct I/O operation via VTAM's channel programs. CP I/O address translation will be bypassed giving VTAM a fast path through CP for I/O processing.
- Update the GCS configuration file to give VTAM authorization to run in the supervisory state and use GCS functions. This may be accomplished by answering questions on display panels provided by the GROUP EXEC function.

VTAM runs with a discontinuous shared segment named in the GCS generation with the VTAMSEG entry. If VTAM's userid is to be other than VTAM, the VTAMSEG name must be changed before generating the VTAM GCS. Also, the VMFPARM file which contains minidisk addresses required to install and maintain VTAM and the VTAM discontinuous shared segment must have the same VTAM name for the shared segment.

- Regenerate the directory with DIRECT VMUSERS.

2. System name table (DMKSNT) update

- Add an entry for VTAM with the NAMESYS macro using SYSNAME=VTAM. The segment name in the VMFPARM file must be the same as SYSNAME.
- Use the GENERATE EXEC with ONLY to assemble and verify the DMKSNT before final assembly.

GENERATE DMKSNT ONLY

- Assemble DMKSNT with GENERATE DMKSNT to update the system name table and regenerate the CP nucleus.

GCS and VTAM Storage Layout and Allocation

VM requires user-coded entries in the system name table to define private storage and shared segments for virtual machines. Both GCS and VTAM require private storage and shared segments for the following:

- GCS private storage
- VTAM private storage, constant and dynamic, to contain:
 - VTAM control blocks
 - VTAM buffers
 - VSCS private storage, if VSCS is to run in VTAM's virtual machine.
- GCS common storage to contain:
 - GCS supervisor
 - GCS control blocks
 - GCS trace table.

Within the GCS common storage area there are VTAM application specific requirements, including:

VTAM constant and dynamic common storage

VTAM buffer pools

VTAM internal trace table

Storage for VTAM activation of channel-attached devices.

- VTAM discontinuous shared segment (VTAM's common storage).

VTAM modules reside in the shared segment. It is recommended, but not required, that the shared segment be discontinuous (addresses above the VTAM virtual machine), since storage savings are gained when other virtual machines, such as NCCF, need shared segment storage. While discontinuous shared segments may be shared by two or more virtual machines, shared segments within a virtual machine may not be shared.

Storage calculations and recommendations may be found in the *GCS Guide* reference manual and, for VTAM, in the *Network Program Products Planning* manual. Also, you may reference the *Network Program Products, Samples: VM SNA* manual for storage requirements and VTAM V3 for VM/SP installation.

VTAM Installation

The installation of VTAM V3 in the VM/SP environment consists of a series of steps outlined in the memo to users. Once VTAM is installed, VTAM major nodes, tables, start options, and configuration lists are coded to define the network. All VTAM definitions are coded using CMS and filed in CMS files.

An overview of the installation steps is as follows:

1. Logon to the VM MAINT userid
2. Mount the VTAM installation tape
3. Attach the tape to MAINT (181)
4. IPL CMS large (CMSL)
5. Invoke the installation exec INSTFPP passing the VTAM program number (5664280 for V3)

```
instfpp 5664280
```

At the completion of the INSTFPP EXEC, VTAM will be installed on the minidisks created during the pre-installation process. The contents of the minidisks is described in the memo to users.

At this point, VTAM may be started to verify the installation procedure. A VTAM start procedure, in the form of an EXEC, is included with the product on the BASE disk with the name VMVTAM GCS. The procedure may be modified by the user for specific VTAM requirements. For example, if VSCS is to run in the VTAM virtual machine, the VSCS startup commands may be included in the procedure. The procedure, after modification, should reside on VTAM's runtime A disk, VTM191.

Communications Products Installation Minidisks

Communications products require two sets of minidisks; one set for installation and the other for operation. All of these minidisks should be defined to the MAINT userid for installation and maintenance. The operation minidisks will be either defined or linked to the VTAM userid when the network operator logs onto the VTAM virtual machine.

The communications products include NCP/3705, NCP/3725, SSP, NCCF, NPDA, NLDM and VTAM itself. An overview of the communications products installation minidisks includes:

- BASE minidisk - the product text decks, macros, and installation control files from the product installation tape (VSCS is on the BASE minidisk with VTAM).
- DELTA minidisk - contains program temporary fixes (PTFs) from program update tapes (PUT tapes) for installation or maintenance.
- MERGE minidisk - receives program updates from the installation tape or from the DELTA disk via the VMFMERGE EXEC during maintenance.
- ZAP minidisk - receives user changes made via the VMFZAP EXEC.
- RUN minidisk - contains the product load libraries and link-edit maps at the conclusion of installation or maintenance. For VTAM, the VTAM discontinuous shared segment map is also placed on this disk.

At the end of VTAM installation there will be two additional minidisks to the above.

- VTM191 minidisk - the VTAM virtual machine 191 A disk which will contain VTAM supplied tables source in ISTxxxxx ASSEMBLE files, and supplied start option, configuration, and application major node samples in ATCxxx00 VTAMLST and ISTAPPLS VTAMLST files.
- MACRO minidisk - contains product macros required for user assembling and link-editing of tables and VTAM application programs. A CMS user would link to the MACRO disk before doing assemblies.

VTAM Runtime Minidisks

When the VTAM virtual machine is logged on for operation, VTAM will need access to the following minidisks:

- MAINT 190 disk - the CMS system disk.
- IPCSE disk - contains VTAM text decks required for formatting GCS dumps with the Interactive Problem Control System (IPCS).
- TRAPRED disk - contains VTAM text decks required for formatting trace records created with the CPTRAP command.
- RUN minidisk - the VTAM module disk created during VTAM installation.
- VTM191disk - VTAM's definitions and LOADLIBS for VTAM tables and NCP load modules.

NCP Generation Overview

A Version 3 NCP is generated in a VM environment with the NCP/EP Definition Facility (NDF) supplied with SSP Version 3 for VM. Examples to generate an NCP are provided with SSP in the form of EXECs. Users may select the EXEC which fits the installation needs, then copy it from the SSP BASE minidisk to the RUN disk to make installation dependent modifications to the EXEC.

The NCP is coded in standard source format using CMS, then filed on the userid A disk with a chosen NCP filename and a filetype of VTAMLST. The generation EXEC will access this generation deck for input to NDF.

The NCP/EP generation macros are in a VM macro library (MACLIB) after installation. NCP/EP text modules necessary at link-edit time are in a VM text library (TXTLIB).

The generation EXEC includes VM file definition (FILEDEF) statements for the macro library, text library, NCP source, work files, and output load library.

An example of the statements for the macro library and the NCP source is:

```
*** Access the NCP macro library
FILEDEF SYSLIB DISK MAC3725 MACLIB *
GLOBAL MACLIB MAC3725

*** Access the user coded NCP source
FILEDEF GENDECK DISK ncpname VTAMLST A
```

NDF is invoked with the command ICNRTNDF after the work files and a printer for the NDF return code summary are defined. NDF puts the text output in simulated PDS data sets for input to the linkage-edit step. An example of the EXEC statements for the NDF output file and the link-edit step is:

```
*** NDF Output Object Code

TXTLIB GEN objname TABLE1 TABLE2
.
.

*** OBJ3725 is NDF Output for LKED Input
*** SYSPUNCH for NCP/EP Table Text
*** SYSLMOD for NCP Load Module and RRT

FILEDEF OBJ3725 DISK OBJ3725 TXTLIB *
FILEDEF SYSPUNCH DISK objname TXTLIB A
FILEDEF SYSLMOD DISK ncpname LOADLIB A
LKED NCPINCL ( MAP NCAL ..... .....
```

Produced on completion of the EXEC are a generation report, return code summary, and table assembly listings. The ncpname LOADLIB will contain the NCP load module and a resource resolution table (RRT) ready for VTAM at NCP load and activation time.

Defining the Network

VTAM definitions for major nodes, start option lists, configuration lists, and PATH definition sets are coded using CMS and filed with a filetype of VTAMLST. For example, the application program major node INQMAJ01 would be filed in a CMS file INQMAJ01 VTAMLST on VTAM's A disk. Normally, the user would be logged on as VM MAINT; however, VTAM files could be coded on other CMS user IDs and later copied to VTAM's A disk.

VTAM tables such as USS, logon mode, and class of service (COS) tables are assembled and link-edited into VTAMUSER LOADLIB on VTAM's A disk. The VTAMUSER load module library is created with the VMFLKED command when VTAM is installed.

VTAM Initialization Processing

The initialization routines which start VTAM in VM are invoked when the system operator issues the EXEC command for VTAM startup.

The VTAM host attachment routines are given control first. The start options specified for the system are examined, and these options are used to establish various operating characteristics for the VTAM domain.

VTAM modules are loaded into the virtual machine, storage is obtained for various control tables, and the tables are initialized. The buffer pools controlled by VTAM's storage management routines are built, VTAM's queues are initialized, and VTAM files are opened. The initialization routines then attach sets of VTAM routines as subtasks. After this processing has been completed, VTAM is ready to accept VTAM commands.

Network Configuration

After the system has been initialized, the parts of the domain and the way in which these parts are connected to the system are defined to VTAM. The configuration of the domain and the characteristics of its nodes are represented to VTAM by a series of control blocks called resource definition table (RDT) segments. Each RDT segment represents a major node with which VTAM will communicate. An RDT segment is composed of a header and a number of entries; each entry representing a specific (minor) node in the domain.

The Initial Configuration

The initial configuration is determined either by a series of operator commands to activate major nodes or by the CONFIG start option. For example, if the example configuration consisted of a dummy ATCCONxx list with no major nodes named, the operator would use the VARY command to activate the VTAM major nodes as follows:

```
VTAM V ACT, ID=INQMAJ01
VTAM V ACT, ID=LOC1NSNA
VTAM V ACT, ID=LOC2SNA
```

VTAM configuration services routines activate and deactivate nodes in the domain in response to operator commands and VTAM definition statements. They also perform automatic logon processing (that is, in addition to activating nodes, they allocate terminals to application programs).

The configuration list ATCCON00 is read and processed. The RDT segments are built for major nodes, such as INQMAJ01 and LOC2SNA, from the VTAM definition statements by the system definition component.

The Example Configuration

The channel-attached I/O devices that VTAM will use may be defined in the VTAM user directory or they may be attached at VTAM startup time. Assuming for the example configuration that the local SNA and local non-SNA addresses have been placed in the VTAM directory entry, VTAM may be started with the start EXEC VMVTAM (on VTAM's VTM191 disk).

The coding thus far for the example configuration includes:

On VTAM's A disk VTM191:

- ATCSTR00 VTAMLST
- ATCCON01 VTAMLST
- INQMAJ01 VTAMLST
- LOC1NSNA VTAMLST
- LOC2SNA VTAMLST

The only required change to the coding for the VM system is that VSCS, a VTAM application program, would need an APPL statement added to the application program major node INQMAJ01. For example:

```
VSCS APPL ACBNAME=VM,AUTHEXIT=YES,PRTCT=VM
```

To start VTAM the network operator would need to do the following:

1. Logon to the GCS recovery machine.
2. Logon to the VTAM userid.
3. Access any VTAM required disks that are not in the VTAM directory entry.
4. Issue the VTAM start procedure command VMVTAM.

Since the major node names are coded in the CONFIG list, and no PATH definition sets are required, the VTAM example network will come up active.

The application program INQUIRE can then be started in a virtual machine within VTAM'S GCS group. Session requests can then be made.

■ ■ ■

Please turn to Mini-Course 5, Exercise 5.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 6

VTAM Considerations in VSE

Mini-Course 6. VTAM Considerations in VSE

Introduction

Virtual Storage Extended/System Package Version 2 is a pre-generated interactive operating system. VTAM is included as one of the VSE/SP subsystem services.

The VTAM product levels supported with VSE/SP Version 2 are:

- ACF/VTAM Version 2 Release 1 for VSE/SP Version 2 Release 1, Modification Levels 1 and 2 (VSE/SP V2.1.1 or V2.1.2)
- ACF/VTAM Version 3 for VSE/SP V2.1.3 or V2.1.4

Among other functional changes to VSE is a new library structure which provides a single, logical library structure for source books, relocatable modules, phases, procedures, and user-defined library types. This library structure replaces the core image, relocatable, and source statement libraries of previous VSE systems.

Although VSE/SP may be installed as a ready-to-run system, it can be tailored through the use of dialogs to complete the base installation.

VTAM coding requirements, such as major nodes, are accomplished through an interactive function called the Interactive Interface. The Interactive Interface is a set of function lists, dialogs, selection panels (menus) and data entry panels. Among the functions of the Interactive Interface is the capability to configure the system hardware including local and remote terminals to be managed by VTAM.

Remote devices that VTAM will manage do not need to be defined to VSE; however, channel-attached devices including local terminals and communications controllers must be defined in the hardware configuration table.

VSE IPL Procedure

The initial program load (IPL) procedure brings VSE into processor storage and passes it information including the types and addresses of channel-attached devices. The IPL procedure is automatically created during VSE/SP initial installation. It may be tailored using the "Tailor IPL Procedure" dialog followed by the "Create Startup Books" dialog which catalogs it on the system library IJSYSRS.SYSLIB. IJSYSRS.SYSLIB corresponds to the former system core image library with the additional capability of housing cataloged procedures.

ADD statements in the IPL procedure are used to define the channel-attached devices to VSE. The ADD statements for VTAM managed resources include:

- Devices attached locally on subchannels including:
 - 37xx Communications Controllers
 - Local SNA and non-SNA Cluster Controllers
- Communications lines attached to a communications adapter.

Adding Communications Controllers

Channel-attached communications controllers (37xx) are specified for VSE with ADD statements with a device type of 3705. For example, a 37xx at channel address 020 would be added as follows:

```
ADD 020,3705,01 (3705 with a type 1 or type 4 Channel Adapter)
ADD 020,3705,02 (3705 with a type 2 or type 3 Channel Adapter)
ADD 020,3705,01 (3725 with its type 5 Channel Adapter)
```

Adding CA Communications Lines

For communications adapters, each SDLC communications line is specified on an ADD statement by its address (030 to 037), a device type of 3705, and a mode setting of 10. For example, an SDLC communications line at address 035 would be entered as:

```
ADD 035,3705,10
```

A BSC 3270 line for VTAM is added with its line address, a device type of 2703, and no mode setting as follows:

```
ADD 037,2703
```

Adding Local Devices

Locally attached devices are specified with the IPL ADD statement.

Local SNA Devices

SNA cluster controllers (that is, 3274 A Models, and the 3791), attached locally to a channel, have their own addresses and a device type of 3791L. The devices attached to the cluster controller do not have channel addresses and therefore are not entered on ADD statements. For example, a cluster controller at address 41A would be defined as follows:

```
ADD 41A,3791L
```

The devices attached to the 3774 are defined in a local SNA major node using the "Configure Local SNA 3270s" dialog.

Local Non-SNA Devices

Non-SNA devices attached to processor subchannels and devices which are attached to the Display/Printer Adapter of the 4300 have their own addresses.

The address range for the Display/Printer Adapter is 009 to 01F. Attached terminals, such as the 3278 Display Station - Model 2, are identified in the ADD statement by address and device type 3277. For example, a 3278-2 attached to the Display/Printer Adapter at address 00B would be added as follows:

```
ADD 00B,3277
```

Non-SNA cluster control units (such as, 3274 B & D Models, or the 3272 Models 1 or 2) must have a subchannel address for each device attached. Local 3270s, such as 3277s or 3278s, are added as device type 3277. A mode setting of 01 is specified for printers only. For example, the first eight ports of a 3274 are reserved for 3278s, and ports 009 and up may be used for 3277s. For a 3278 and a 3277 attached to a 3274 cluster control unit at addresses 0A0 and 0A8, and a 3287 Printer at address 0A9, the ADD statements would be as follows:

```
ADD 0A0,3277      (the 3278)
ADD 0A8,3277      (the 3277)
ADD 0A9,3277,01   (the 3287)
```

The VTAM local non-SNA major node is defined with the "Configure Local Non-SNA 3270s" dialog.

Entering IPL Statements

In a VTAM system with local terminals as well as communications controllers or communications adapter lines, the number of ADD statements may be significant. The IPL statements may be cataloged as a procedure for use with VSE Advanced Function - Automated System Initialization (ASI). Or, the IPL statements may be entered by the system operator either through the operator console, or via a card reader.

VTAM Partition

VTAM must run in a partition with a higher priority than any of the VTAM application programs. The virtual and real partitions for VTAM are specified with the ALLOC and ALLOCR commands.

For calculating VTAM storage requirements, refer to the *Network Program Products Planning* manual.

VTAM Start Procedure

The VTAM startup job is supplied with VSE/SP as VTAMSTRT, one of the supplied skeletons. The VTAM start procedure contains VSE job control statements to supply a job name, define disk file labels, and assign devices to channel addresses. The supplied startup skeleton for VTAM startup in a VSE/POWER environment is shown in Figure 6-1.

```
* $$ JOB JNM=VTAMSTRT,DISP=L,CLASS=3
// JOB VTAMSTRT START VTAM
// ASSGN SYS000,UA

// ASSGN SYS001,DISK,VOL=SKSWK1,SHR TRACE FILE ASSIGNMENT
// ASSGN SYS004,DISK,VOL=SKSWK1,SHR TRACE FILE for TPRINT
// ASSGN SYS008,DISK,VOL=SKSWK1,SHR NCP LOAD MODULE ASSIGNMENT
// ASSGN SYS012,DISK,VOL=SKSWK1,SHR NCP DIAGNOSE FILE

// LIBDEF PHASE,SEARCH=(user sublibraries,..,PRD2.COMM,PRD2.CONFIG,
PRD1.BASE)PERM
// LIBDEF OBJ,SEARCH=(user sublibraries,..,PRD2.COMM,PRD2.CONFIG,
PRD1.BASE)PERM
// LIBDEF SOURCE,SEARCH=(user sublibraries,..,PRD2.COMM,PRD2.CONFIG,
PRD1.BASE)PERM
// LIBDEF DUMP,CATALOG=SYSDUMP,private
// EXEC INTINCVT
/&
* $$ EOJ
```

Figure 6-1. VTAM Startup (VTAMSTRT Skeleton)

Changes to the supplied startup skeleton are made using the "Program Development Library" dialog procedure. For example, user sublibrary search chains need to be modified according to installation requirements.

Standard labels are provided for the files shown in Figure 6-1. User files other than shown would require DLBL and EXTENT information in the startup procedure.

A partial list of supplied communications program components and their permanent sublibraries is as follows:

PRD2.COMM - ACF/NCP 3705, NCCF, NPDA X.25 VTAM

PRD2.COMM2 - ACF/NCP 3725, EP 3725

PRD2.BASE - ACF/VTAM, CICS/DOS/VS

VTAM major nodes, start option lists, CONFIG lists, and PATH definition sets may be in PRD2.CONFIG or in other private sublibraries.

More than one VTAM startup book may be cataloged to allow user flexibility. For example, one procedure might include statements for trace files in a test environment while another might exclude tracing for the production environment.

Optional VTAM Files

Configuration Restart Files

Additional files are required if the user selects the VTAM configuration restart facility. The configuration restart file is used to store the status of minor nodes (active or inactive) when there is a network failure. Each major node is related to a different user-defined VSAM file.

Configuration restart is defined to VTAM in each major node definition with the parameters CONFGDS="filename" and CONFGPW="file password". For an NCP major node, the parameters are coded in the NCP PCCU macro.

In addition, the user may select to have VTAM keep the status of all major nodes (and NCP dynamic reconfiguration data sets, DRDS) in a NODELST file. The NODELST file is a user-defined VSAM file which a network operator may select with the CONFIG start option when restarting VTAM. The major nodes would then be restored to their original status before a failure if the operator had started VTAM originally and included the NODELST start option.

Trace Files

When traces are to be taken, the trace records may be stored on tape or disk files by user option. VTAM checks to see if the trace file at SYS001 has been assigned. If it has, tracing is recorded externally on the SYS001 device. VTAM also creates internal trace tables for trace records.

The VTAM TPRINT utility is used to print trace records whether internal or external recording is used. When recording is on an external SYS001 file, SYS004 is assigned to the same file for TPRINT trace record input. TPRINT requires that SYSLST be assigned in the VTAM partition when TPRINT is invoked with a VTAM operator command. SYSLST may be assigned to a tape, printer, or disk.

VTAM Initialization Processing

The *VTAM initialization routines*, which start VTAM in the operating system, are invoked when the system receives the EXEC command for ISTINCVT.

The *VTAM host attachment routines* are given control first. The start options specified for the system are examined, and these options are used to establish various operating characteristics for the VTAM domain.

VTAM modules are loaded into the partition, storage is obtained for various control tables, and the tables are initialized. The buffer pools controlled by VTAM's storage management routines are built, VTAM's queues are initialized, and VTAM files are opened. The initialization routines then attach sets of VTAM routines as subtasks. After this processing has been completed, VTAM is ready to accept VTAM commands.

Network Configuration

After the system has been initialized, the parts of the domain and the way in which these parts are connected to the system are defined to VTAM. The configuration of the domain and the characteristics of its nodes are represented to VTAM by a series of control blocks called *resource definition table* (RDT) segments. Each RDT segment represents a major node with which VTAM will communicate. An RDT segment is composed of a header and a number of entries—each entry representing a specific (minor) node in the domain.

VTAM configuration services routines activate and deactivate nodes in the domain in response to operator commands and VTAM definition statements. They also perform automatic logon processing (that is, in addition to activating nodes, they allocate terminals to application programs).

The source statement library book B.ATCCON00 is read and processed. The RDT segments are built for major nodes, such as INQMAJ01 and LOC2SNA, from the VTAM definition statements by the system definition component.

Example Configuration - Putting It All Together

Sample IPL Procedure

The example configuration IPL procedure shown in Figure 6-2 is cataloged in the VSE procedure library.

```
.  
.
ADD O24,3277          3278 Display Terminal
ADD O25,3277          3278 Display Terminal
ADD O26,3277B         3286 Printer
ADD OB8,3791L         The local 3274 Control Unit
DEF SYSCAT=DOSRES,SYSREC=SYSWK1
DLA VOLID=DOSRES,BLK=....,NBLK=....,DSF=N,NAME=....
SVA PSIZE=nnnK,SDL=nnn,GETVIS=nnK
.  
.
```

Figure 6-2. Sample IPL Procedure

Example Start Procedure

The requirements for the first example configuration assume that no traces are to be taken; therefore, no label sets or ASSGNs are necessary for the trace file or the SYSLST file. Also, the first example assumes that VTAM phases and transients are in the PDR1.BASE sublibrary; the major node definitions, the start options, and the configuration list are in a private source sublibrary PRD2.CONFIG.

The VTAM start procedure for the example configuration, ready to be cataloged in the VSE procedure library, is shown in Figure 6-3.

```
// ASSGN   SYS000,UA
// ASSGN   SYS001,UA
// ASSGN   SYS004,UA
// LIBDEF  PHASE,SEARCH=(PRD2,CONFIG,PRD1.BASE),PERM
// LIBDEF  OBJ,SEARCH=(PRD2,CONFIG,PRD1.BASE),PERM
// LIBDEF  SOURCE,SEARCH=(PRD2,CONFIG,PRD1.BASE),PERM
// LIBDEF  DUMP,CATALOG=SYSDUMP.xx,PERM          xx=partition
// EXEC ISTINCVT,SIZE=nnnn
/*
/&
```

Figure 6-3. Example Configuration Start Procedure

Sample VSE Sublibraries

The current VSE libraries for the example configuration are shown in Figure 6-4 and summarized below.

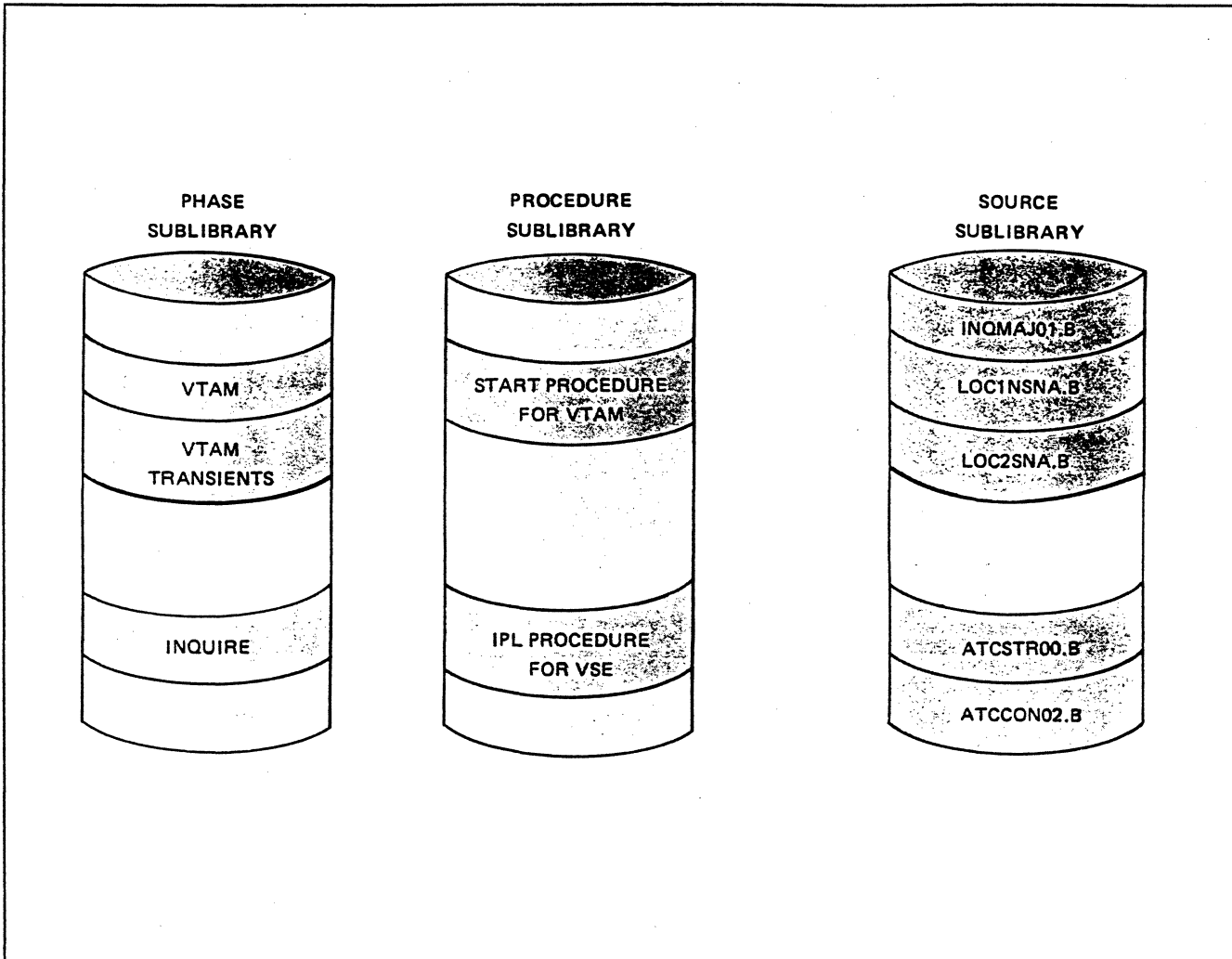


Figure 6-4. Sample VSE Libraries

Phase Sublibrary

The VSE PHASE sublibraries for the example configuration will contain the following:

- The VTAM phases
- The VTAM transients
- The INQUIRE application program.

Source Sublibrary

The source sublibrary includes the following books from the example configuration:

- INQMAJ01.B - the application program major node definition
- LOC1NSNA.B - the local non-SNA definition
- LOC2SNA.B - the local SNA definition
- ATCSTR00.B - the start options list
- ATCCON01.B - the configuration list.

Procedure Sublibrary

The VSE procedure sublibrary for the example configuration will contain the following:

- The VTAM start procedure.
- The VSE IPL procedure.

Starting the System

Once the VSE IPL procedure has been invoked and the partitions have been allocated, VTAM can be started.

The general procedure for starting VTAM is the same as it is for the example configuration. The one difference would be when the Start Option list specifies PROMPT for the operator to enter the options. In the example, all necessary options have been coded in ATCSTR00.B including NOPROMPT.

The steps for starting the VTAM example configuration are as follows:

1. Start the VTAM partition with the VSE START command.
2. Invoke the VTAM sample start procedure.
3. Call in the first VTAM phase (ISTINCVT) with the VSE EXEC command. VTAM will begin its initialization process and display a console message for each major node that it activates. The last message will be:

"5A20I, VTAM INITIALIZATION COMPLETE"

The application program INQUIRE can now be started in another VSE partition of lower priority.

Had there been a requirement for the operator to supply any start options (PROMPT in ATCSTR00.B), there would be a message, "5A51A, ENTER VTAM START PARAMETERS". The operator would then type in any overriding options. The start options are merged from ATCSTR00.B, the operator options, and another ATCSTRyy.B list (if it is selected by the operator LIST=yy option). VTAM asks for corrections if any options are in error.

The Initial Configuration

The initial configuration is determined either by a series of operator commands to activate major nodes or by the CONFIG start option. For example, if the example configuration consisted of a dummy ATCCONxx list with no major nodes named, the operator would use the VARY command to activate the VTAM major nodes as follows:

```
V NET,ACT,ID=INQMAJ01
V NET,ACT,ID=LOC1NSNA
V NET,ACT,ID=LOC2SNA
```

■ ■ ■

Please turn to Mini-Course 6, Exercise 6.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 7

VTAM-Only Operands in Device Major Nodes

Mini-Course 7. VTAM-Only Operands in Device Major Nodes

Introduction

There is a set of VTAM-only operands that appear on device definitions in all types of device major nodes. Although the operands appear with the device definitions, they function to define session options during installation. The specific operands, grouped with their primary function, are:

MODETAB,DLOGMOD	LU-LU Session Initiation
USSTAB,SSCPFM,DISCNT	SSCP-LU Session for End User Logon/Logoff
ISTATUS	Initial Component Status, ACTIVE/INACTIVE
LOGAPPL	Automatic LU-LU Session Request
VPACING	Data Flow Control, VTAM outbound

In an NCP major node, these VTAM-only operands are changed by modifying only the source code on the VTAM definition library. Since the NCP load module is generated without knowledge of these operands, the changes will be effective the next time that the NCP is activated.

In local major nodes, or in a VSE or VM communications adapter (CA) major node, these operands are modified by changing and recataloging the major node definition in the VTAM definition library. The changes will become effective the next time the major node is activated.

MODETAB, DLOGMOD

When there is an LU logon request, VTAM will send the name of a suggested set of session parameters (protocol) to the primary LU regardless of the request source. These session parameters are always stored in logon mode tables which reside in SYS1.VTAMLIB (MVS), VTAMUSER LOADLIB (VM) or in the VTAM module library (VSE). There is a default logon mode table supplied with VTAM that contains valid entries for selected device types. User coded entries for other device types or default overrides are stored in one or more assembled and link-edited tables.

If the supplied logon mode table does not contain an appropriate entry, 3270 devices for example, then the MODETAB operand is coded. MODETAB points to a specific user-coded table. Since it is coded at the terminal/LU level, every device could have a separate logon mode table. In practice, all like devices usually use the same logmode entry and the same MODETAB coding. ***VTAM will search the default table automatically if MODETAB is not coded.***

DLOGMOD stands for default logmode entry name and is a pointer to a specific entry in either a user-coded table (MODETAB=) or in the IBM-supplied default table.

DLOGMOD is only used by VTAM when a logmode entry name is not supplied at some higher level. Entry name codings take precedence over DLOGMOD in a hierarchy as follows:

1. When a LOGMODE pointer is entered by the end user in a LOGON command.
2. When a LOGMODE pointer is entered by the network operator on behalf of the end user in a VARY LOGON command.
3. When a LOGMODE pointer is coded as a default in a USS table.

VTAM checks for each of these LOGMODE pointers in turn. If none are supplied, the device definition is checked for the DLOGMOD pointer and used to search the logon mode table. VTAM first searches the optional table (MODETAB=), and if the entry is not found, the default table is searched. ***If the named entry can not be found in either table, VTAM rejects the logon request.***

There is a special procedure when no LOGMODE pointer (or DLOGMOD) is given to VTAM. In this special case, VTAM will pass the name of the first entry coded in the MODETAB= table or, if no MODETAB= is coded, the name of the first entry in the default table.

Note: The user can take advantage of the VTAM search order by coding a separate logon mode table for each device type and not supply a LOGMODE pointer or DLOGMOD operand at all. VTAM will then pass to the primary LU the correct entry name since it is the first (and only) entry in the table.

It is important to recognize that the primary LU has the option to use a set of its own session parameters. This happens when the primary LU (application program) either has hard-coded session parameters or gives VTAM its own LOGMODE entry name. In either case, the suggested entry name is ignored even though VTAM is required to pass a valid name in the logon request.

USSTAB, SSCPFM, & DISCNT

The device definition operands USSTAB and SSCPFM are used by VTAM when the device is not in an LU-LU session. VTAM is in session with the device after the device becomes ACTIVE. This session is called the SSCP-LU session.

The SSCPFM operand appears on SDLC device definitions to inform VTAM of the type of messages to send to or expect from the logical unit. There are two general types of SDLC terminal/LUs:

- Programmable terminals are capable of sending to VTAM a formatted logon request as a command (the INITIATE-SELF command). The command includes the necessary information for VTAM to pass the logon request directly to the primary LU (application program).

Included are:

- a. The name of the requesting terminal/LU.
- b. The name of the application program for which the request is intended.
- c. The name of an entry in a logon mode table associated with the terminal/LU.

Formatted logon requests are processed in VTAM in a routine called formatted system services. VTAM knows from the SSCPFM operand to expect formatted logon requests from programmable devices such as the IBM 8100 or 3790.

For these devices, either SSCPFM = FSS must be coded or left out (FSS is the SSCPFM default).

- Nonprogrammable terminals are not capable of sending the INITIATE-SELF command and therefore use a different type of LU-LU session request.

One form of logon request is called a character-coded logon request. This normally means that an end user enters the request from a keyboard, such as at a 3270 display station. The character coded requests are processed by the SSCP unformatted system service (USS) routine according to the SSCPFM operand.

There are various methods of USS character coded logon message processing, depending on the type of SDLC device and user options. A summary of SSCPFM USS values follows:

SSCPFM = USSSCS for SNA terminals which use the SNA character string (SCS) character coded messages. For example, all display stations connected to IBM 3274 Control Units in SDLC mode use SCS messages for the SSCP-LU session.

SSCPFM = USS3270 for SDLC 3271 Models 11 & 12 LU logon requests.

SSCPFM = USS3270 or USS3275 for SDLC 3275 Models 11 & 12 display stations. USS3275 suppresses a USS *good morning* message when a printer is attached to the 3275.

SSCPFM = USSNTO or USS3780 for non-SNA terminals supported by the IBM Network Terminal Option (NTO) program product.

Session initiation using character-coded logon requests is always processed by the SSCP unformatted system service by referencing a USS table. VTAM has a supplied default USS table which contains a standard format for character-coded logon requests (logon commands). If an easy to remember mnemonic is desired, USS tables may be coded by the user to replace the standard logon format. The default USS table also contains the LOGOFF command, a standard format for LU-LU session termination.

The USSTAB operand on the device definition is used to tell VTAM that a user-coded USS table exists for the device. When a message is received from the terminal during the SSCP-LU session, SSCP's USS processing uses that USS table to translate the logon (or logoff) request. If the incoming logon command can not be found in the USS table, the default USS table is searched as well before an error message is sent back to the originator.

The DISCNT operand on the device definition tells VTAM what to do when all LU-LU sessions for a given PU are terminated. If DISCNT is coded YES, then VTAM will automatically terminate the SSCP-PU and SSCP-LU session as soon as the last LU-LU session terminates.

When DISCNT=NO is coded, an option such as to code the desired disconnection of the PU and LUs in the USS table is possible. This is done with operands on the USS logoff command.

ISTATUS

VTAM will automatically attempt to activate a network component that has `ISTATUS=ACTIVE` in its definition as soon as its superior node becomes `ACTIVE`. (`ISTATUS=ACTIVE` is the default.) This allows the user to plan how the initial status of the network is to look. `ISTATUS=ACTIVE` is equivalent to the network operator command `VARY ACT`. The alternative is to code `ISTATUS=INACTIVE` which is equivalent to the `VARY INACT` command.

For example, if there were a number of PUs and LUs defined on a particular communications line, the `LINE` definition could be coded as `ISTATUS=INACTIVE`, letting all of the PU and LU definitions default to `ISTATUS=ACTIVE`. When the network operator uses the single `VARY ACT` command for the line, VTAM would attempt to contact all of the PUs and LUs automatically.

The `ISTATUS` operand allows limitless combinations for initial network status. If in the example above, the line and all of the LUs were coded `ACTIVE` and the PUs `INACTIVE`, the network operator would have a choice when to activate the PUs (and their LUs automatically) on an already active line.

LOGAPPL

There may be devices in the network that are intended for a single application program. In this case, the user may request that VTAM automatically pass a logon request to a particular application program when that device becomes active to VTAM. This is done by coding the `LOGAPPL` operand, in the device definition, with the VTAM name of the application program. The VTAM name is the name on the `APPL` statement for the program in the application program major node.

For example: If a device defined as LU111 is to be primarily used for transactions in the CICS/VS application program, and the CICS APPL statement were coded as:

```
DBDCCICS  APPL  AUTH=(....
```

VTAM would send a logon request to CICS on behalf of LU111 at activation time if the LU definition were coded:

```
LU111    LU    ...,LOGAPPL=DBDCCICS,...
```

A LOGAPPL logon does not involve USS tables; however, a valid logon mode table entry name is required. This could be a DLOGMOD entry name, or the default first entry in a mode table as discussed above under MODETAB and DLOGMOD.

The LOGAPPL operand also establishes a special condition for the LOGAPPL device. VTAM associates the LOGAPPL terminal/LU with the application program as if that application program owned the terminal/LU. VTAM marks the terminal/LU as having a controlling application program.

Controlling application program means that after an end-user logs off from the original application program to log on to a second program, when a logoff occurs for the second program, VTAM will automatically send a new logon request to the owning controlling application program.

For example: Steps in the above example for LU111 (LOGAPPL=DBDCCICS) could be:

- LU111 becomes ACTIVE (via ISTATUS=ACTIVE or the VARY ACT command)

VTAM automatically passes a logon request to CICS which becomes the controlling application program for LU111.

- The terminal operator at LU111 logs off from CICS to log on to a second program.
- The terminal operator logs off the second program.
- VTAM automatically passes a logon request to CICS, the controlling application program.

Another way to establish a controlling application program is to use the network operator command VARY LOGON on behalf of the LU. This approach would normally be used only in a test environment or to change the controlling application program.

VPACING

It is possible that VTAM application programs can flood the network with messages to the point where network components cannot keep "pace" with the data flow. The VTAM-only operand VPACING can be coded in device definitions to slow-down the rate of messages entering the network. VPACING applies to outbound traffic to the network when coded on device definitions. Conversely, when VPACING is coded on an application program's APPL statement, it applies to messages flowing inbound from the network to the application program.

The VPACING operand on the device definition tells VTAM how many PIUs to send from the primary LU (application program) to the secondary LU before waiting for a pacing response. A second VPACING value can be specified to tell VTAM on which of the PIUs to turn on the pacing indicator.

If PIUs were destined for LUs connected to an NCP, VPACING would define pacing only between VTAM and the NCP (the boundary function). Pacing between NCP and the LU would depend on another device operand: PACING. The combination of VPACING to the NCP and PACING from the NCP to the LU is called *2-stage pacing*.

With certain devices pacing is not used. For example, due to the logical pacing in 3270 data stream operations, explicit pacing is not generally required. For display output, an application program usually will not send subsequent output until receiving acknowledgement from the operator that the current display has been received and can be overwritten.

■ ■ ■

Please turn to Mini-Course 7, Exercise 7.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 8

VTAM Logon Considerations

Mini-Course 8. VTAM Logon Considerations

Introduction

This mini-course explains how an LU may log on to a VTAM application program and how session protocols are established for SNA LUs. Before logon, the required preliminary SNA commands, including commands to activate the physical unit and logical unit must be transmitted and the appropriate positive responses returned to VTAM.

An application program is available when it has an open access method control block (ACB). Generally, a program also needs an active logon exit.

There are four ways that VTAM can establish a connection between a VTAM application program and an available terminal.

- Field-formatted or character-coded
- Host acquire
- Automatic logon
- Operator VARY command.

The logon considerations differ between programmable LUs and non-programmable LUs. Figure 8-1 on page 8-2 illustrates the logon request processing discussed in the next sections.

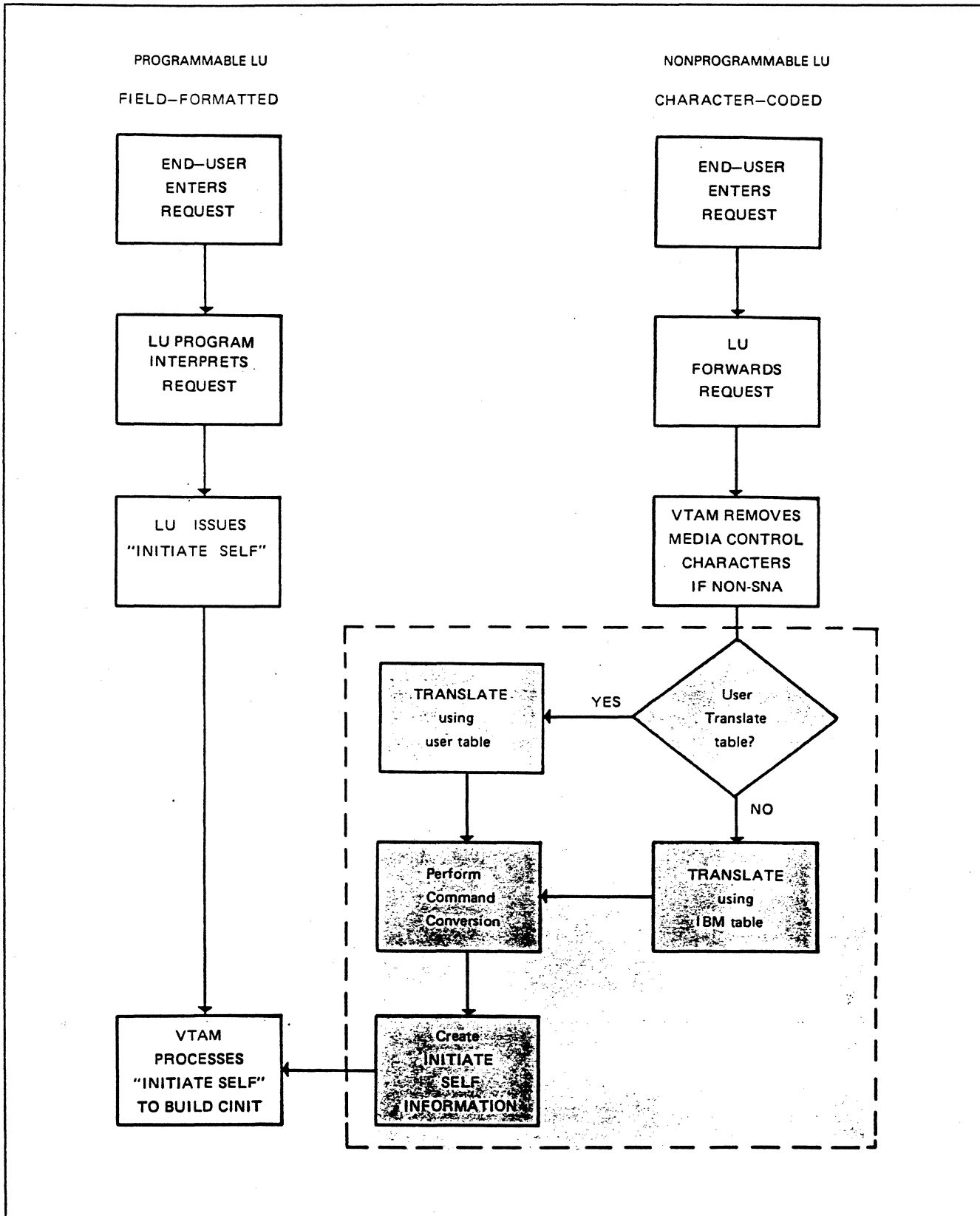


Figure 8-1. LU Logon Processing

Programmable LUs

A programmable LU is capable of transmitting certain SNA commands, including Initiate-Self and Terminate-Self. A logon request may be initiated by naming the VTAM application program through transmission of an INITIATE-SELF command.

To begin the logon, the terminal operator enters data, perhaps in response to a prompt by the LU function program. The LU program interprets the data as being a logon request for a particular application program and issues the INITIATE-SELF command to VTAM.

The SSCP processes the INITIATE-SELF command and builds a control initiate (CINIT) command to pass to the application program. The CINIT contains the session parameters, and the name of the application program requested and the requesting LU. This type of logon from a programmable LU is called field-formatted, because the INITIATE-SELF command has an SNA defined format.

Non-programmable LUs

Logons from non-programmable LUs are more complex. A typical non-programmable LU is essentially a typewriter with a keyboard, and does not possess the ability (it is non-programmable) to generate an INITIATE-SELF command. Its capability is limited to transmitting data keyed by the operator and cannot accept user-coded programming.

Since VTAM's SSCP can only process logon requests in the form of an INITIATE-SELF command, an intermediate piece of code must be introduced in VTAM. This code, called *unformatted systems services (USS)*, is designed to receive character-coded messages from an LU, convert the character string to a form acceptable to the SSCP, and pass the converted request to the SSCP which makes the desired connection. Thus, the USS functions to convert the data string sent by the LU into the equivalent of an INITIATE-SELF command which SSCP can recognize and then pass to an application program.

An IBM-supplied USS table must be present for VTAM to start. This table, called the session-level USS table, is looked-up by VTAM at initialization time. The user may modify, replace or code additional USS tables, then assemble and link-edit them into the VTAM module library.

VTAM also loads another USS table at initialization time called the operation-level USS table. This table is supplied with VTAM and contains:

- All of the network operator console messages issued by VTAM
- Selected VTAM operator commands.

This table may also be modified by the user.

Logon Mode Tables

The session protocols which SNA LUs may use during a session with an application program are specified by logon mode tables, frequently called *logmode tables*. IBM supplies a standard logmode table with every VTAM system. That table must be present in order for VTAM to start successfully. In addition to the IBM logmode table, the user may have any number of user-defined logmode tables. In practice, the user will probably define at least one logmode table entry for every different device type, since session parameters vary from one SNA device to another. (For example, what is acceptable to printers will not work for display devices.)

The LU macro should be coded with a `MODETAB=` parameter to specify the logmode table which lists the appropriate parameters. Logmode tables are coded, assembled, and link-edited to the VTAM module library by the user.

The logmode table is specified by `MODETAB=`, but the table entry can be pointed to in several ways as listed below:

- The end-user in a USS logon
- A USS table `DEFAULT` logmode value
- The application program
- The LU definition `DLOGMOD=` operand
- The network operator with a `VARY LOGON` command
- A programmable LU sending an `INITIATE-SELF` command
- The first entry in a logmode table if none of these are specified.

Session Parameters

With a programmable LU, the name of a logmode entry may be specified by a field in the INITIATE-SELF command which the LU sends to request the logon. With a non-programmable LU, the operator keys in a data string, which is converted by a USS table into the equivalent of an INITIATE-SELF command.

The operator can override the USS table's default LOGMODE parameter when logging on by supplying the name of an entry in the logon request. Typically, however, the logmode name is supplied by a USSPARM default in the USS table.

A logmode entry contains a set of rules which specifies how the requested session is to be conducted. For example, it determines if pacing and brackets will be used, and whether the communication is half-duplex or full-duplex.

After the CINIT command is built, VTAM passes control to the application program which processes the logon request and issues an OPNDST macro, which returns control to VTAM. VTAM then transmits an SNA BIND command, containing the session parameters, to the LU to establish the session.

The parameters may or may not be the same as those originally requested by the LU. The application program has four options:

- To accept the parameters requested by the LU in the logon request.
- To obtain the bind parameters requested by the LU and make any desired changes to the requested parameters.
- To ignore the parameters requested by the LU and create different parameters in an area in the program known as a bind area.
- To ignore the parameters requested by the LU and select a different set of parameters from the logmode table associated with the LU trying to log on.

Whichever option the VTAM program selects, the program-determined session parameters are transmitted to the LU as a part of the BIND command.

The LU has the option of accepting or rejecting the session parameters. If the LU rejects them, no session is established. In a non-programmable LU, the hardware (microcode) determines whether the BIND parameters are acceptable. In some programmable LUs, the LU function program may examine the BIND parameters and decide whether to accept them.

The search logic for logon mode table entry is illustrated in Figure 8-2.

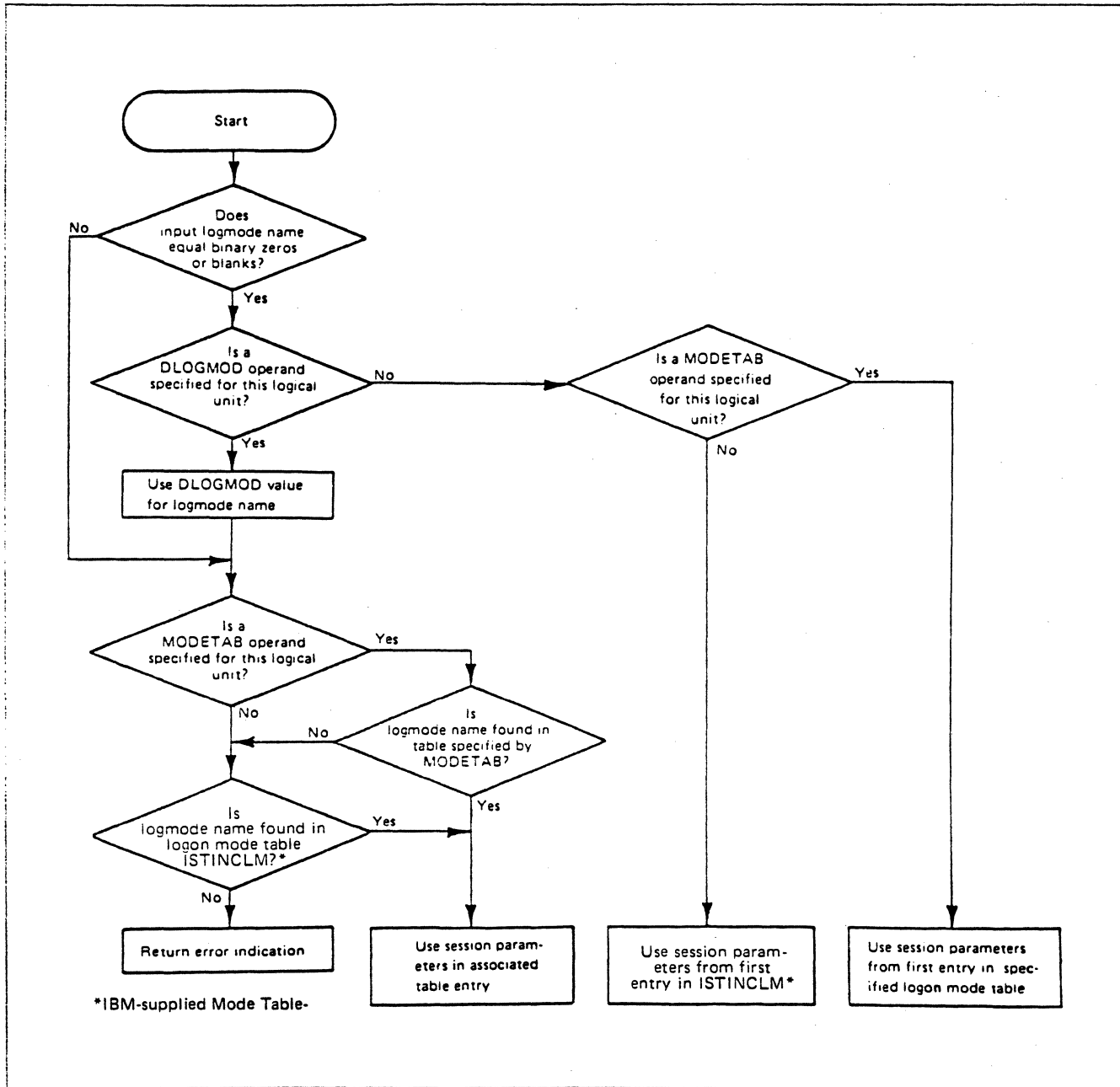


Figure 8-2. Search Logic for Logon Mode Table Entry

If the LU request doesn't specify a particular logmode entry name, a default is implied. When MODETAB= is not coded for the LU, then the standard IBM logmode table is the one associated with the LU. While VTAM must have a valid entry name, regardless of whether the application program uses it, VTAM follows a default hierarchy to determine the logmode entry name, as listed below:

1. A name supplied in the LOGON command
2. A default coded in the USS table logon entry
3. A DLOGMOD coded in the LU definition
4. The first entry in the associated logmode table

Thus, if logmode tables are coded for each device type with valid single entries, it is never necessary for the logon message to identify the name of a logmode entry. If each application program specifies the session parameters for every session, the required IBM logmode table is the only one that need be present, and the user does not have to code any logmode tables, MODETAB= parameters, DLOGMOD parameters, or USS table LOGMODE defaults.

■ ■ ■

Please turn to Mini-Course 8, Exercise 3.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 9

USS Table Macro Coding and USS Messages

Mini-Course 9. USS Table Macro Coding and USS Messages

Introduction

A non-programmable terminal cannot directly log on to VTAM since it is incapable of generating an INITIATE-SELF command. USS tables are coded in the host to interpret the character-coded message the terminal can generate, and convert it to the required formatted message that SSCP can process. In this mini-course we will examine how USS tables are coded for unformatted logons. In addition, error messages, IBM-supplied session-level USS tables, and user translation tables will be discussed.

Function of Session-Level USS Tables

The top part of Figure 9-1 on page 9-2 shows the sequence of events involved in a USS logon request. The following explanation of how the conversion is made from a character-coded message to equivalent Initiate-Self information using a USS table should make the coding logic easier to understand.

At the bottom of Figure 9-1, a simplified version of an INITIATE-SELF command is shown. (An actual INITIATE-SELF command is more complex than the format shown in Figure 9-1. Certain fields have been omitted for this discussion.)

Recall that the purpose of the session-level USS table is to generate Initiate-Self information in the format shown at the bottom of Figure 9-1.

Please note the terminology at the bottom of Figure 9-1. The command is the word LOGON. For logons there are only three parameters—the keyword parameters of APPLID, LOGMODE, and DATA. The values are the actual data coded for each of the three parameters.

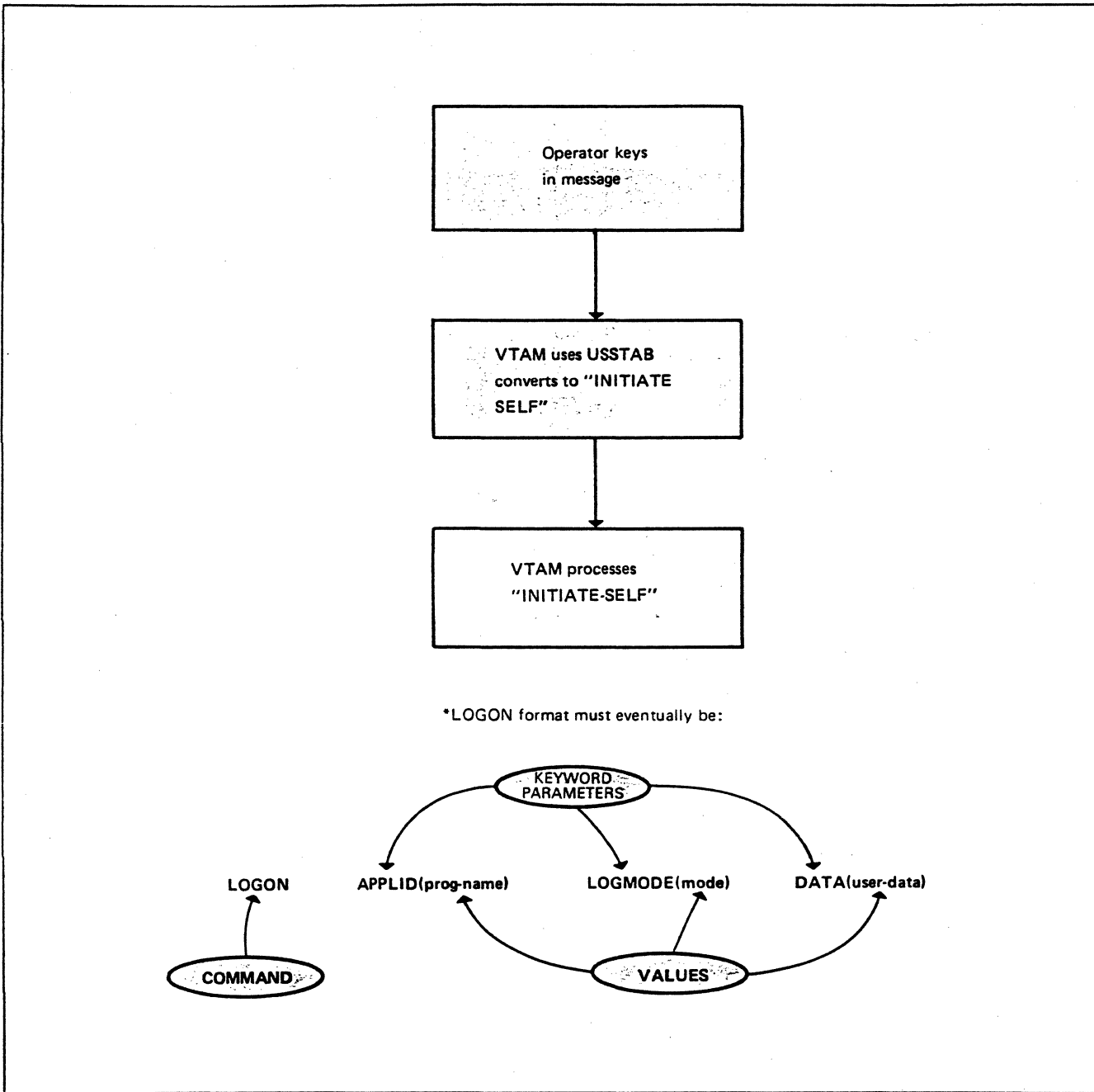


Figure 9-1. Character-Coded Logon - Standard Format

APPLID

APPLID is the keyword identifying the application program onto which the end user wants to log. It will be matched with a name on an APPL statement in the application program major node.

LOGMODE

LOGMODE is the keyword identifying the logmode entry for the session. In a USS table, a logmode name is the name of an entry in a logmode table associated with the requesting LU. The purpose of logmode tables is to provide session parameters which may be used in requested sessions.

DATA

DATA is the keyword identifying the optional user data field.

Note the values for each of the above parameters shown at the bottom of Figure 9-1. The values are the user-supplied program minor node name, logmode entry name, and optional user data.

Uses of Session-Level USS Tables

One use of a USS table is to allow substitute names for LOGON or LOGOFF commands. The user can substitute any word to replace either command word, LOGON or LOGOFF. In the USS table, the user can supply defaults for various parameters and values. (See Figure 9-1 to review the meaning of *parameter* and *value*.)

Default values from the USS table may be overridden by the LU operator, if desired.

The normal use of a USS table is to allow the operator to key in a few characters and have the USS table generate the program name, the logmode name, and any required optional user data. For example, the operator may key:

INQ

The USS table then generates everything necessary to log on to the program.

USS commands are normally coded with defaults for APPLID and LOGMODE to save end-user overrides when logging on to an application.

USS Table Macros

Figure 9-2 shows the macros which make up a USS table.

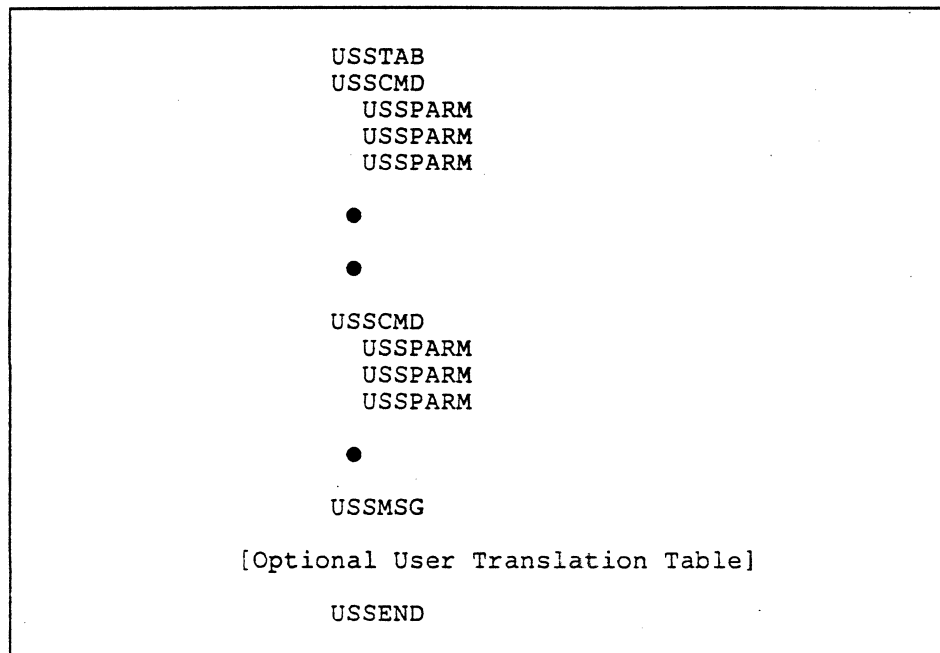


Figure 9-2. USS Table Format

The first macro is always USSTAB; the last macro is always USSEND. In between, there must be at least one USSCMD macro. Usually, each USSCMD will be followed by three USSPARM macros:

- To define the APPLID (APPL name)
- To define the LOGMODE (session parameter list name in a mode table)
- To define the optional user DATA field.

The optional USSMSG macro is used to specify user-defined text for a specific condition.

■■■

Please turn to Mini-Course 9, Exercise 9.1, in your PRG and answer the exercise questions.

USS Table Macro Coding

The USS table macros shown in Figure 9-2 consist of keyword operands which are coded to suit a particular VTAM environment. The macros, the operands, and coding considerations are described as follows:

USSTAB

Figure 9-3 shows the format of the USSTAB macro.

<p>[name] USSTAB [TABLE = table addr]</p> <ul style="list-style-type: none">● Optional name (member name in module library is real name).● TABLE = Address pointer to optional character translation table. <p>If none is specified, the default table STDTRANS is used.</p> <p>It converts lower case to UPPER CASE, etc.</p>

Figure 9-3. USSTAB Macro

USSTAB may have an optional name, which is ignored if present. The only valid parameter is the TABLE = parameter identifying an optional user translation table.

The IBM-provided translation table translates all lowercase letters to uppercase and translates the X'05' (horizontal tab character) to a blank. No other characters are altered.

If the default IBM translation table is acceptable, there is no need to code a user translation table in a user USS table. However, if it is necessary to translate characters keyed in by the LU operator, a user translation table must be included in the same assembly with the USS table and coded just before the USSEND macro.

Figure 9-4 shows the format of a USSCMD macro.

[name]	USSCMD	CMD=user command name
		REP=LOGON or LOGOFF
		FORMAT=PL1 or BAL

Figure 9-4. USSCMD Macro for Session-Level USS Table

The USSCMD macro is used to define each command with which an LU will initiate a logon or a logoff. For example, if the LU operator keys either RUN, EXEC, or INQ to request a logon, three USSCMD macros must be included in the USS table, one for each command. Each of these USSCMD macros will define one command; each will have some USSPARM macros associated with it.

Figure 9-4 shows the three parameters which may be coded in a USSCMD macro: CMD, REP, and FORMAT.

CMD=char-string

This parameter defines the character string which the operator keys to generate the LOGON command. For example, if the LOGON command is to be INQ, it is coded:

CMD=INQ

REP=LOGON|LOGOFF

The REP value specifies whether the command (*char-string*) is to be interpreted by VTAM to mean LOGON or LOGOFF. For this example, it will be coded REP=LOGON.

FORMAT=PL1|BAL

This parameter specifies the format of any overriding that the operator may do when keying a particular logon request. In most operations, the end user will not override any table parameters; in these cases, it doesn't matter whether BAL or PL1 is coded since PL1 is the default.

Figure 9-5 shows a coding example for the USSCMD macro.

```
USSCMD  CMD=INQ,REP=LOGON,FORMAT=BAL
```

Figure 9-5. USSCMD Example for Session-Level USS Table

The parameters `CMD=INQ` and `REP=LOGON` mean that if the end user keys `INQ`, the command is converted by the operation of the USS table to `LOGON`. In other words, `INQ` is defined as a synonym for `LOGON`. `FORMAT=BAL` says that if end users will ever override any of the default values supplied by the table, they must use the `BAL` format. (After some basic considerations, the difference between `BAL` and `PL1` formats will be shown.)

The user may now key the word `INQ` (or `inq`) and `LOGON` is understood. But, the parameters must still be supplied. (See Figure 9-1 on page 9-2 to review parameters.) The parameters may be specified with the `USSPARM` macro as shown in Figure 9-6.

```
USSPARM  PARM=user name or Pn
          REP=APPLID or LOGMODE or DATA
          DEFAULT=substitution value
          VALUE=optional substitution method
```

EXAMPLE:

```
USSCMD  CMD=INQ,REP=LOGON,FORMAT=PL1
USSPARM  PARM=P1,REP=APPLID,DEFAULT=INQUIRE
USSPARM  PARM=P2,REP=LOGMODE,DEFAULT=MODE3270
USSPARM  PARM=P3,REP=DATA,DEFAULT=PASSWD14
```

Figure 9-6. USSPARM Parameters with PL1 Example

The top of Figure 9-6 shows the parameters that may be coded in a `USSPARM` macro. The rest of the figure shows a sample `USSCMD` and its associated `USSPARM` macros.

At the top of Figure 9-6, notice that each `USSPARM` macro has four parameters: `PARM=`, `REP=`, and `DEFAULT=` or `VALUE=`. (`DEFAULT=` and `VALUE=` cannot be coded together.) Three of these parameters are normally specified for every `USSPARM` macro.

PARM =

PARM means *parameter*. If the operators will not be overriding default parameters specified in the USS table, PARM = is not referenced. In the example shown in the lower part of Figure 9-6 the parameters are coded PARM = P1, PARM = P2, and PARM = P3. These positional parameters are explained in the next section.

REP = replacement

For a logon request, replacement must always be one of three parameters: APPLID, LOGMODE, or DATA.

DEFAULT = default

For a logon request, default is the default name of the application program (the APPL minor node name), the name of the logmode entry, or the optional user data. For each of these possibilities, the default value is the one used for this USSCMD command, unless the user overrides the default by specifying a different one.

Note: If the defaults are coded for a particular application program, an end user only needs to enter the CMD name to request a session with the program.

VALUE = value

The VALUE parameter and DEFAULT are mutually exclusive. When the keyword (APPLID, LOGMODE or DATA) is entered by the operator in a logon request without a value, the name coded for VALUE is substituted for the keyword. An end user would have to know the appropriate keywords for VALUE to be substituted properly. Normally, USS tables are coded using DEFAULT rather than VALUE.

Example USS Command

Look again at the example in the lower part of Figure 9-6. The first USSPARM macro which follows the USSCMD macro can be read as follows:

The application program which the command INQ requests is the program named INQUIRE, which is coded default for APPLID.

If the operator wants to use the INQ command to log on to an application program other than the one named INQUIRE, the first parameter must be overridden (P1 means *positional parameter #1*).

We will now examine how the end user can override the first USSPARM shown in Figure 9-6. The P1 in the first USSPARM can be read as:

The first override value keyed by the operator is understood by VTAM to be the APPLID value—that is, the APPL name of the VTAM application program. For example, keying INQ PROG2 would request a session with a different program; the program with an APPLID of PROG2.

The P1 in the second USSPARM in Figure 9-6 can be read as follows:

The logmode name (the name identifying session parameters to be used in this session) defaults to MODE3270 if not overridden by the operator.

The second parameter (P2) must be overridden if the operator wants to override MODE3270. When the operator overrides the default values in the INQ command, the second value keyed by the operator is understood by VTAM to represent the name of the logmode entry.

The third USSPARM can be read as follows:

The INQ command always generates a user data field of PASSWD14 unless specifically overridden by the end user.

The operator must override the third parameter (P3) in order to override this default DATA value.

If the end user keys only the LOGON command INQ, all defaults are generated by the action of USS processing. Those defaults, shown in Figure 9-6, are:

```
APPLID default:    INQUIRE
LOGMODE default:  MODE3270
DATA default:     PASSWD14
```

In other words, the INQ command generates a logon request for an application program which has an APPLID of INQUIRE, requested logmode name of MODE3270, and a user data field of PASSWD14. Item #1 in Figure 9-7 shows this sample output from the INQ command.

```
#1 user keys:      INQ (or inq)
  converts to:
  LOGON APPLID(INQ) LOGMODE(MODE3270) DATA(PASSWD14)

#2 user keys:      INQ (CICS NEWMODE)
  converts to:
  LOGON APPLID(CICS) LOGMODE(NEWMODE) DATA(PASSWD14)
```

Figure 9-7. USS Session-Level Command Examples

One way to implement USS tables is to have a separate USS table for every unique device type in the system. This allows all possible valid combinations of program name, logmode name, and user data field to appear as defaults, and, the same CMD name can be used in each. That way, the operator never has to know how to perform overrides when logging on. Instructions to the operator can read something like this:

```
Key  INQ  to log on to inquiry program
        UPDT to log on to update program
        PERS to log on to personnel program
```

Note: If different device types are represented in a single table, then different CMDs would be needed to make the LOGMODE parameter default correctly. For example, a single USS table for CICS would need CMDs such as CICS3279, CICS3767, CICS3290, and so on.

An end user would have to remember multiple USS commands in this case.

Overriding Parameters

Now, let's look at another example showing how the operator can override default parameters specified in the USS table. Assume that the INQ command defined in Figure 9-6 on page 9-7 is being used and that a logon to program CICS (instead of the default of INQUIRE) is required. A logmode of NEWMODE (instead of the default of MODE3270) is also required. The default user DATA field of PASSWD14 is acceptable for program CICS. Look at item #2 in Figure 9-7 which shows how to key this logon request.

The first parameter specified in parentheses is CICS. This override replaces the default program name of INQUIRE because the first USSPARM for the INQ command in Figure 9-6 specifies that the APPLID (or application program name) is parameter 1. Therefore, since CICS is keyed as parameter 1 in the override it replaces INQUIRE in the generated output. Similarly, the second parameter keyed by the operator—NEWMODE—replaces the second parameter in the USS table definition (LOGMODE value of MODE3270). The result appears as shown on the bottom line of Figure 9-7. The APPLID is CICS, LOGMODE is NEWMODE, and DATA remains the default value of PASSWD14.

Item #2 in Figure 9-7 illustrates that when multiple parameters are overridden, the entire set of parameters must be enclosed in parentheses. This requirement is true of the PL1 format but not of the BAL format. (Remember that FORMAT=PL1 was specified in the USSCMD macro in Figure 9-6.)

Either a comma or a space may be used by the operator to delimit parameters within the parentheses in PL1 format.

By their very nature, positional parameters (P1, P2, and P3) must be overridden by the operator in the same order in which they were coded in the USSPARM macro.

Regardless of the order in which the USSPARM macros in the USS table are specified, or in which order the operator keys in the override parameters, the output of a USS table is always referenced by VTAM in this sequence:

```
LOGON APPLID(value) LOGMODE(value) DATA(value)
```

We have been discussing USSCMD macros with FORMAT=PL1. Now let's look at a USS table with the BAL format.

EXAMPLE:

```
USSCMD      CMD=UPDT,REP=LOGON,FORMAT=BAL
USSPARM     PARM=P1,REP=APPLID,DEFAULT=CICS
USSPARM     PARM=P2,REP=LOGMODE,DEFAULT=MODE3270
USSPARM     PARM=P3,REP=DATA,DEFAULT=PASSWD14

#3  user keys:      UPDT  (or updt)
    converts to:
          LOGON  APPLID(UPDT)  LOGMODE(MODE3270)  DATA(PASSWD14)

#4  user keys:      INQ  CICS,NEWMODE
    converts to:
          LOGON  APPLID(CICS)  LOGMODE(NEWMODE)  DATA(PASSWD14)
```

Figure 9-8. USS Table – BAL Format Example

Figure 9-8 shows a USSCMD and USSPARMs with positional parameters and FORMAT=BAL. Remember, specification of the FORMAT parameter (BAL or PL1) has no effect on the output of the table if the operator keys only the command. The purpose of the parameters BAL or PL1 is to specify how the operator keys in overrides to the default values. If the operator is not going to key any override values, it does not make any difference whether FORMAT=BAL or FORMAT=PL1 is coded.

Now, look at item #3 in Figure 9-8. When the operator keys UPDT, all the USS table defaults are generated.

In item #4, P1 (CICS) and P2 (NEWMODE) are specified, and the table default for P3 (user data of PASSWD14) is acceptable. Notice that when the BAL format for a command is used, multiple parameters are not enclosed in parentheses. When overriding PL1 multiple parameters, the parentheses must be keyed.

USSPARM and Keyword Parameters

The other kind of parameters aside from positional parameters—keyword parameters—are shown in Figure 9-9.

```

EXAMPLE:
USSCMD      CMD=INQ3,REP=LOGON,FORMAT=PL1
USSPARM     PARM=PROG,REP=APPLID,DEFAULT=CICS3
USSPARM     PARM=MODE,REP=LOGMODE,DEFAULT=MODE33
USSPARM     PARM=USER,REP=DATA

```

```

#5  user keys:          INQ3   (or inq3)

```

```

converts to:

```

```

LOGON  APPLID(CICS3)  LOGMODE(MODE33)  DATA()

```

```

#6  user keys:          INQ3  USER(EMP780)  MODE(MODE3270)

```

```

converts to:

```

```

LOGON  APPLID(CICS3)  LOGMODE(MODE3270)  DATA(EMP780)

```

Figure 9-9. USSPARM - PL1 and Keyword Parameters

When keyword parameters are coded in a USS table, parameters are specified by user chosen names. The upper part of Figure 9-9 shows an example using keywords.

Look at example #5 in Figure 9-9. Notice that all the defaults are taken, including null data since no DATA default was coded.

Recall that if the operator is not going to override any of the parameters, then it does not matter whether the parameters are positional (P1, P2 and P3) or keyword.

Example #6 shows how the operator can override the keyword parameters to specify a different program name, different logmode, and different user data. Because of `FORMAT=PL1` in the `USSCMD` macro, the override format shown in #6 must be used. That is, for each parameter to be overridden the keyword name must be keyed, followed immediately by a left parenthesis, followed by the override value, and closed by a right parenthesis.

Using keyword parameters, the macro writer can allow the operator to key in replacement names in any order for the three parameter names. That is what has been done in Example #6; that is, the operator keys `MODE` instead of the longer `LOGMODE`, and so forth.

The second `USSPARM` in Figure 9-9 shows that if the parameter `MODE` is encountered, it is understood to mean `LOGMODE`. Similarly, `PROG` is converted to `APPLID`, and `USER` becomes `DATA`.

The operator does not have to key in the override keyword parameters in any particular order, as is necessary when using positional parameters. Thus, example #6 could have been keyed as:

```
INQ3 MODE(MODE3270) USER(EMP780)
```

The conversion to the standard format would have been the same.

Because PL1 or BAL can be coded in the USSCMD macro, and either keyword parameters or the positional parameters (P1, P2, P3) can be coded, there are actually four possible combinations. Three of these combinations have been discussed:

BAL and positional

PL1 and positional

PL1 and keyword

The fourth possibility, BAL and keyword parameters, works the same way as PL1 and keywords.

USS Table Notes

The following are additional notes on USS Table coding

- The types of parameters may be mixed in one set of USSCMD/USSPARM macros. That is, both positional parameters and keyword parameters may be included. This combination of parameters makes the operating instructions for the remote operators somewhat difficult to follow, but the mix is legal. However, unnecessary mixing of parameters is not recommended.
- The positional parameters (P1, P2, and P3) are also automatically keyword parameters when the operator overrides them. Therefore, the following two logon commands are equivalent:

```
INQ PROG77  
or  
INQ P1=PROG77
```

- Normally, three USSPARM macros are coded for every USSCMD macro. If there is no need for one of the parameters, it may be left out. For example, if there is no user data requirement, a USSPARM for REP=DATA is unnecessary.

■■■

Please turn to Mini-Course 9, Exercise 9.2, in your PRG and answer the exercise questions.

USS Messages

There are two situations where error messages are necessary for logon procedures:

- When an operator makes an error keying a parameter for logon (or logoff).
- When a table is incorrectly coded.

The default error messages are precoded in the IBM standard USS table furnished with each VTAM system. When an error occurs, for example, the name of the requested program is unknown to VTAM, and the appropriate error message is sent back to the LU.

The USSMSG macro allows the substitution of user-defined text for each of the standard IBM-supplied error messages. In addition, the macro allows the user to code a good morning message and a logon accepted message.

Figure 9-10 shows a sample of the VTAM messages coded in the supplied session-level USS table.

```
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'
```

Figure 9-10. Sample USS Standard Messages

SSCP will select messages according to message numbers in the table for a corresponding error. Therefore, the wording may be changed, but not the meaning. The default error messages in the supplied USS table may be modified in one of two ways:

- In the USS table for a particular LU, specify the USSMSG with MSG=n, where n is the number of the IBM message to be replaced. If a USSMSG macro in a user USS table has been coded, that error message is used by VTAM instead of the equivalent error message from the IBM USS table.
- Change the IBM USSTAB by changing the text of a particular USSMSG error message and reassembling the IBM USSTAB table.

The general format of the USSMSG macro is as follows:

```
USSMSG MSG=n,TEXT='new message text'
```

The message text must be enclosed in single quotes.

As an example, error MSG=4 always means PARAMETER INVALID. It may be altered to read: 'REQUESTED PROGRAM CURRENTLY NOT AVAILABLE' which is one of the common reasons for MSG=4 to be issued. This is more descriptive if the application program is installed but not

active to VTAM, since the APPLID parameter is invalid only until the program is made active. It would be coded as:

```
USSMSG MSG=4,TEXT='REQUESTED PROGRAM CURRENTLY NOT AVAILABLE'
```

In Figure 9-10, notice the use of the percent (%) character in three of the USSMSG macros. When an error message is sent to the LU, the % is replaced with the incorrect command or parameter, as applicable. Thus, if the operator keys:

```
XYZ
```

and XYZ is not defined by a USSCMD macro in either the user USS table or in the IBM USS table, the command is invalid, and VTAM generates error message #2, substituting the characters XYZ for the percent sign. Therefore, the standard IBM error message sent is:

```
XYZ COMMAND UNRECOGNIZED
```

If replacements for the IBM error messages are coded, the % sign may be used for command substitution.

Another useful replacement in messages is the requesting LU's name. If @@LUNAME is coded in the message text, VTAM will replace the keyword with the appropriate LU name any time the message is issued.

When the operator keys in a USS command, VTAM looks for the USSCMD definition in the user-defined table (if any) for that LU. (Remember that a user USS table for a particular LU is specified by coding the USSTAB parameter in the LU macro.) If there is no user USS table specified, or if the user USS table does not define the command, VTAM looks in the standard IBM USS table for the definition.

A value with a pair of single quotes around it is passed to VTAM without any of the translation that normally occurs. For example, if the user data parameter is keyed as:

```
DATA='hello AppL' (BAL format)  
or  
DATA ('hello AppL') (PL1 format)
```

translation does not occur, and the characters 'hello AppL' are passed to VTAM (and subsequently to the user application program) without translating them in any way.

IBM-Supplied Session-Level USS Table

The IBM-supplied default table should not be modified unless it is necessary. If the IBM-supplied USS table is changed, however, the USSMSG macros coded there should not be deleted. If a macro is deleted, there is a possibility that the error message 14, MESSAGE NOT DEFINED, could be generated by USS.

MSG = 0

USSMSG macros may be coded to specify MSG=0 and MSG=10. These two error messages are not specified in the IBM-supplied USS table, but they may either be added to the IBM USS table or specified in a user-written USS table.

MSG=0 corresponds to a message indicating that a USS command has been processed by the SSCP with no detected errors. This message verifies that a valid USS LOGON or LOGOFF command has been generated from the data string keyed by the operator.

It is a good idea to code this message for the operator. If the system programmer coded USSMSG 0, the operator would know that the original message had been accepted by SSCP as a valid request. This knowledge would make it easier to determine the reason for a logon failure.

MSG = 10

MSG=10 corresponds to a message that is automatically sent to an LU when the LU is first activated or after it is disconnected from an application program. In effect, MSG=10 indicates to the LU operator that the LU is available for logon to a VTAM application program. This can be a good morning message, a logo, or even a menu of valid application programs.

To create multiple line display images for MSG=10, the form of the USSMSG macro is:

```
USSMSG MSG=10,BUFFER=address
```

The address is a pointer to a series of assembler language define constants (DCs) beginning with a two-byte binary length field of the total number of bytes in the display image. Following the length is the desired logo or menu image (usually defined with a string of character constants).

When using BUFFER= it is important to determine the characteristics of the device. For example, a 3767 has a 256-byte buffer and cannot accept segmented PIUs; therefore, a 3767 menu screen cannot exceed 256 characters. Remote SNA 3270 display stations are not restricted to PIU size (segmentation is supported) and full-screen menus (MSG=10 with BUFFER=) may be coded.

Note: LU name replacement (@@LUNAME) is not valid when using the BUFFER= format of USSMSG.

Coding MSG=0 and MSG=10 in USS tables is highly recommended.

User Translation Table

A user-coded translation table is pointed to by the TABLE= parameter in the USSTAB macro. A user translation table is coded only when there is a specific reason to override the supplied character translation.

The user translation table, which must define all 256 bytes, is in all respects a standard translation table, such as might be used with the assembler language TR (translate) instruction.

If a user translation table is coded, that table determines the translation that will take place on the logon/logoff message, and the IBM-provided translation table is ignored.

■ ■ ■

Please turn to Mini-Course 9, Exercise 9.3, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 10

Logmode Table Coding,
VTAM Connections and Disconnections

Mini-Course 10. Logmode Table Coding, VTAM Connections and Disconnections

Introduction

Logon mode tables, often referred to as logmode tables, were introduced in the videotape for Mini-Course 8. Recall that they are lists of session protocols which SNA LUs may use during a session between an LU and a VTAM application program. The user codes, assembles, and link-edits logmode tables into the VTAM module library.

Logmode Table Macros

During logon, VTAM selects an appropriate entry to pass to the program from the list of session parameters (also called BIND parameters or logmode entries) supplied by the logmode table. The application program is in control in this respect since the program determines which set of BIND parameters actually accompanies the BIND command when the BIND is sent to the LU.

When the LU receives the BIND command and the accompanying BIND parameters, the LU determines whether it will accept those parameters and responds to the BIND command. If the parameters are accepted, the session is bound between the LU and the application program.

A logmode table begins with a MODETAB macro and ends with a MODEEND macro. In between are any number of MODEENT (MODE ENTRY) macros. Mode tables are assembled and link-edited into the VTAM module library. (SYS1.VTAMLIB in MVS, a private definition library in VSE, or VTAMUSER LOADLIB in VM).

Figure 10-1 shows the format of these macros.

```
[name] MODETAB  
[name] MODEENT LOGMODE = MODE3270,etc.  
[name] MODEENT LOGMODE = M3279,etc.  
      ●  
      ●  
[name] MODEEND
```

Figure 10-1. Logon Mode Table Macros

The example shows two logon mode entries. Note that the logmode entry name is coded inside the macro with the LOGMODE = parameter. This differs from other VTAM definitions where names are taken from the name field of the macro. Here, the name field is optional.

Code a MODEENT macro for every different set of session parameters you want to define in this table. The MODEENT macro specifies a particular set of session parameters and gives those parameters a name. That name can be specified by the operator in a LOGON command or generated as a default by the USS table.

A sample logmode table is shown in Figure 10-2.

MTAB3270	MODETAB		
	MODEENT	LOGMODE=MODE 3279, (entry name)	X
		FMPROF=X'03', (FM profile)	X
		TSPROF=X'03', (TS profile)	X
		PRIPROT=X'B1', (Primary protocols)	X
		SECPROT=X'90', (Secondary protocols)	X
		COMPROT=X'3080', (Common protocols)	X
		RUSIZES=X'0000', (Request unit sizes)	X
		PSERVIC=value, (Presentation services information, 12 bytes)	X
		COS=COS3270A (Class of service name)	
	MODEENT	LOGMODE=BATCH, (entry name)	X
		etc.	
	MODEEND		

Figure 10-2. Logmode Table Example

Determining Session Parameters

Determining the proper values for the various parameters of the MODEENT macro requires some attention. Some of the parameters are determined by the type of device which will use the logmode table. For example, the IBM 3274 Control Unit LUs require different session parameters for display stations and printers.

To determine the BIND parameters for a particular SNA device:

- Refer to the logon mode table entries in the IBM-supplied logon mode table ISTINCLM. The table may be found in the *VTAM Customization* manual for either MVS and VSE, or VM in the appendix titled “IBM-Supplied Tables.” The IBM-supplied entries include, for example, 36xx devices, and many of the IBM 3270 Information Display System devices.
- Refer to the *VTAM Programming* manual appendix titled “Specifying Session Parameters” for an explanation of the meaning of the bits in the BIND parameters.
- Consult SNA reference material in device component description manuals for session parameter information.

■ ■ ■

Please turn to Mini-Course 10, Exercise 10.1, in your PRG and answer the exercise questions.

VTAM Connections and Disconnections

VTAM Connections

One of VTAM's advantages over other access methods is its ability to establish connections between any VTAM application program and any LU in the network. This advantage exists regardless of what line the LU is on and regardless of what the other LUs on that line are doing.

A terminal or LU can not be connected to a VTAM application program unless both the program and the terminal or LU are available.

An application program is available when it has an open access method control block (ACB). Generally, a program also needs an active logon exit.

A terminal or LU is available when all three of the following conditions are met:

1. The VTAM status of the LU is active. Active means that either ISTATUS=ACTIVE was coded in the terminal's definition, or the operator VARY ACT command has been used to activate the terminal.
2. The terminal is not connected to another application program.
3. There is no logon request pending for that terminal.

VTAM can establish a connection between a VTAM application program and an available terminal in the five ways shown in Figure 10-3.

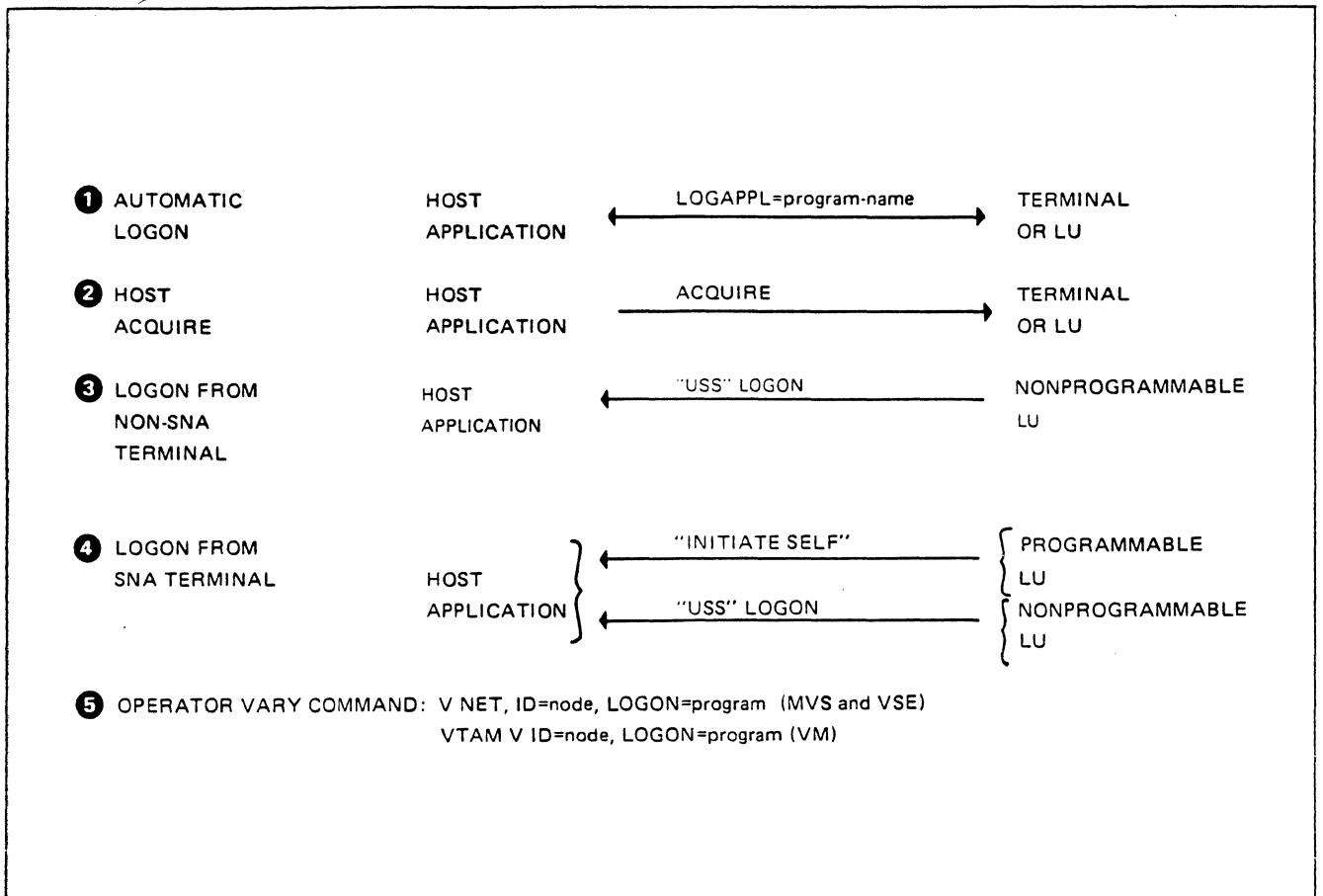


Figure 10-3. VTAM Connections

Automatic Logon

In an automatic logon (see Figure 10-3 item #1) VTAM requests connection as soon as both the application program and the LU are available.

To enable an automatic logon, the system programmer specifies in the terminal or LU definition:

LOGAPPL=appl-name

When a VTAM application program has opened an ACB with the internal name of appl-name, and when the LU is available, VTAM will drive the program's LOGON exit.

No matter what the source of the logon request, a VTAM application program is never required to accept the logon request. If the application program rejects the logon request, no session is established.

Host Acquire

The host application program can attempt to acquire the LU (see Figure 10-3 item #2) by issuing the OPNDST macro with the acquire option set. If the LU is not available, the connection attempt is rejected by VTAM. If the LU is available, then the connection is made, and data can be exchanged.

An alternative for the program is to issue a SIMLOGON macro for a particular LU which requests VTAM to drive the LOGON exit when the LU becomes available. In this case, VTAM will build the CINIT, drive the LOGON exit, and receive an OPNDST with option ACCEPT requesting VTAM to send the BIND image to the LU. CICS/VS uses this method to acquire terminals when CONNECT= AUTO is coded in the TCT table for the device.

Logon - Non-SNA

Non-SNA terminals are internally treated as SNA PU__T1 terminals. That is, VTAM contains code which provides for the modification of transmission data to an SNA structure. This allows non-SNA terminals to use USSTABs for logon (see Figure 10-3 item #3).

An alternate logon method is to code a logon interpret table to define the format of the logon message from a non-SNA terminal. One of the advantages of the logon interpret table is that the logon message is not limited to eight characters. For example, a terminal user might type "I WANT PROGRAM3" as an acceptable logon message for a program defined in the table. Character translation, however, is not available, and no user data or logmode table name can be passed to the program.

Refer to "Interpret Table" in the *VTAM Customization* manual for details on coding logon interpret tables.

Logon - SNA

Certain SNA programmable LUs have the ability to issue the SNA command INITIATE-SELF (see Figure 10-3 item #4). The INITIATE-SELF command is processed by VTAM's System Service Control Point (SSCP). The SSCP interprets the INITIATE-SELF command from the LU as being a request for a connection to a specific VTAM application program. If the desired program is available, SSCP passes the connection request to the application program for acceptance or rejection.

Non-programmable LUs do not have the ability to send an INITIATE-SELF command. These LUs must send VTAM a character-coded message for SSCP to interpret by using a USSTAB (as described in the previous mini-course). A major advantage of a USS table is the ability to use a user-specified format for the logon messages.

Both the logon interpret table and USS table may be coded for the same device. VTAM looks for both LOGTAB= and USSTAB= on the device definition. If both are coded, VTAM will search the logon interpret table before searching the USS table when a character-coded logon is received.

Note: Logon interpret tables may also be coded for SNA devices. The LOGTAB= parameter on the LU definition would point to the table.

Operator VARY Command

The operator VARY command can be used by the network operator to request a connection (see Figure 10-3 item #5).

For example, to request that logical unit LU24 be connected to VTAM application program PAYROLL, the network operator would key:

```
V NET, ID=LU24, LOGON=PAYROLL
```

And, using the same example, if the operator wanted to override the logmode entry name, it would be entered as:

```
V NET, ID=LU24, LOGON=PAYROLL, LOGMODE=NEWMODE
```

The network operator form of logon is useful in a test environment for specific LUs. As with all VTAM operator commands issued in a VM environment, the command must be preceded by the word VTAM. The VTAM identifier NET is optional, therefore the first example above would be entered as:

```
VTAM ID=LU24, LOGON=PAYROLL
```

VTAM Disconnections

An existing connection may be broken in three ways, as shown in Figure 10-4.

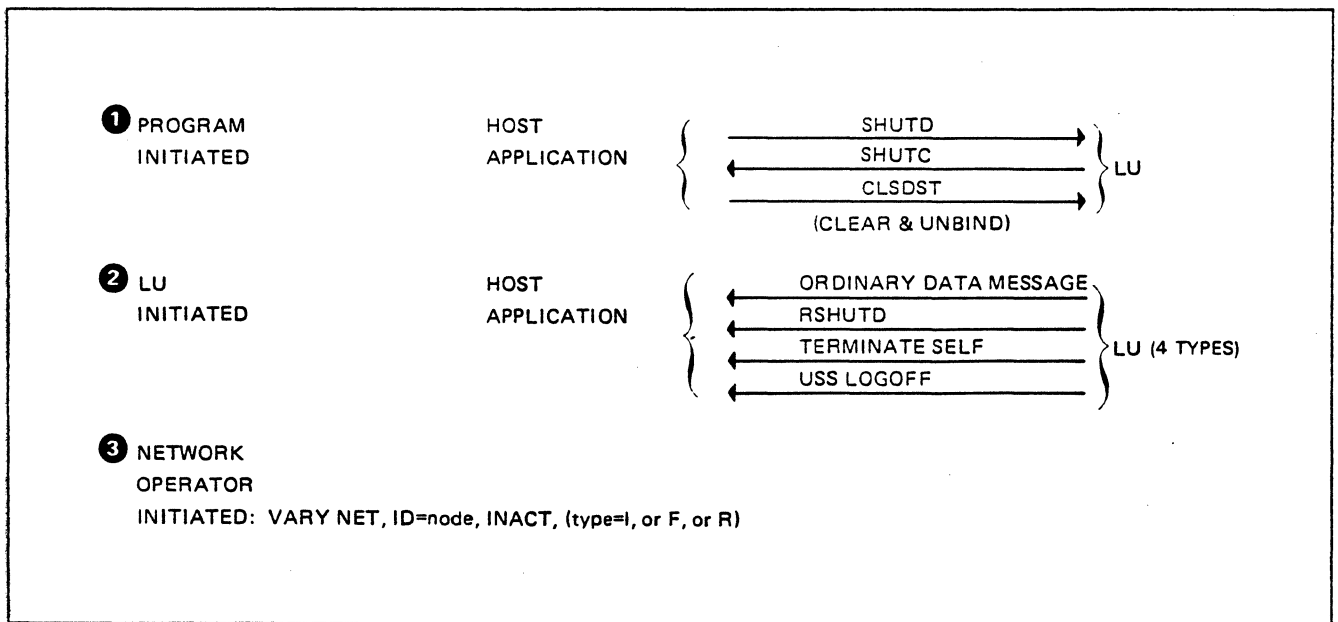


Figure 10-4. VTAM Disconnections

VTAM Application Program Initiated Disconnections

The application program can force a disconnection at any time by issuing the CLSDST macro. When the CLSDST macro is issued, VTAM generates an UNBIND command which terminates the connection between the program and an LU in an abnormal manner. This type of abrupt disconnection generally is unacceptable. For example, if the LU is programmable, it is unlikely that the LU program can terminate normally when the VTAM program has "pulled the plug."

If the network node is a non-programmable SNA device or a non-SNA device, the terminal operator might not even know that the connection was terminated.

If the program must disconnect the LU, a better approach is for the program to use orderly shutdown procedures (see Figure 10-4 item #1). The program sends to an LU the SNA command SHUTD (shut down). When the LU is ready to terminate, it sends SHUTC (shutdown complete) back to the program. The program then issues the CLSDST macro which sends an UNBIND to the LU. (The UNBIND command may be preceded by CLEAR in some cases.)

Before disconnecting a non-programmable LU or a non-SNA terminal, it is useful for the program to send a *Goodbye* message, so that the terminal operator knows that the connection is going to be broken.

LU Initiated Disconnections

When a request for disconnection originates with the terminal or LU, it is usually termed a logoff request. An LU can request a logoff in four different ways (see Figure 10-4 item #2).

- The LU can send an ordinary data message to the application program which the program recognizes as a request for disconnection. This method can be used by any terminal or LU. A non-SNA terminal cannot force a disconnection; it can only request one. The application program must cause the disconnection.
- An intelligent LU can issue an SNA command called RSHUTD (request shutdown). In effect, RSHUTD requests that the program terminate the connection. The program, at its option, may continue the session. RSHUTD is the recommended way for a programmable LU to request a disconnection.
- A programmable LU can issue a TERMINATE-SELF command. There are two kinds of TERMINATE-SELF commands: conditional and unconditional. An unconditional TERMINATE-SELF causes immediate termination of the session. A conditional TERMINATE-SELF requests that the session be terminated, but it is the application program which determines whether the session continues. Use of a conditional TERMINATE-SELF command is functionally equivalent to issuing an RSHUTD. (There are some differences in the coding of the VTAM application program.)

Use of the TERMINATE-SELF command, whether conditional or unconditional, results in an error condition in the VTAM application program, and the LOSTERM (lost terminal) exit is invoked.

If the TERMINATE-SELF command is unconditional, the application program is not given an opportunity to send any end-of-connection data to the LU, such as statistics. An unconditional TERMINATE-SELF usually should be used only as a last resort by the LU. For example, assume that the LU has attempted to disconnect from the program, but the program has refused. If the LU operator must perform some other task, then TERMINATE-SELF may be sent.

- The LU can send to VTAM's SSCP a USS logoff message which the SSCP interprets as a request for a logoff. Any LU can make this type of request; but in practice, it is the non-programmable LUs and non-SNA terminals which use this facility. This logoff request requires a USS table.

Like the TERMINATE-SELF command, a USS logoff may be either conditional or unconditional. Therefore, a USS logoff request functions exactly as if the LU had sent a TERMINATE-SELF SNA command. (Recall the earlier discussion of how a USS table converts a character-string from the LU into the equivalent of an INITIATE-SELF or TERMINATE-SELF command.)

Network Operator Initiated Disconnections

The VTAM network operator can use the VARY command to force an LU or terminal to be deactivated (see Figure 10-4 item #3). For example, assume NODE3 is an LU connected to a VTAM application program. If the network operator enters a forced inactivation:

```
VARY NET, ID=NODE3, INACT, F
```

NODE3 is immediately deactivated, the session is disrupted, and VTAM notifies the application program that the session is terminated abnormally.

The VARY INACT, when type R (reactivation) is used, will cause VTAM to first go through a forced (type=F) inactivation then immediately attempt to reactivate the node. This might be used in an attempt to free a terminal when an existing session is in a locked condition.

When the operator requests an inactivation with an immediate (type=I), VTAM disrupts the session but allows the program to issue the CLSDST macro to terminate the session. Since VTAM waits for the CLSDST to be issued, the network operator may choose to use the force option to clean up the session.

Logoffs

The purpose of a USS LOGOFF command is to generate a logoff message as shown in Figure 10-5. (As you might expect, a USS table logoff has the same relationship to a TERMINATE-SELF command as the USS table logon has to an INITIATE-SELF command.)

```
LOGOFF APPLID(appl-name) TYPE(COND|UNCOND|FORCE) HOLD(YES|NO)
```

Figure 10-5. Output Format of USS LOGOFF

USS LOGOFF Command

Any USS LOGOFF commands for an LU are specified in the same USS table in which the USS logons for that LU are coded. In the USSCMD macro, specify `CMD=name`, where name is the user-defined equivalent for LOGOFF. For example, if you want to allow the user to enter OFF in order to issue a USS LOGOFF command, code:

```
[name] USSCMD CMD=OFF,REP=LOGOFF,FORMAT=PL1
```

When the LU operator keys OFF, the USS table converts that command to the standard LOGOFF command. The USSPARM macros are coded following the USSCMD for the LOGOFF command.

For a LOGOFF command, values may be specified (or the defaults accepted) for the three parameters: APPLID, TYPE, and HOLD.

Figure 10-5 shows the possible values for each of these parameters.

USSPARM Parameters

APPLID(appl-name)

Normally, you will not code this parameter, in which case appl-name defaults to the name of the application program with which the LU is in session.

TYPE(COND)

This coding specifies that the request for a logoff is conditional. Conditional logoff means that the LU is requesting the logoff, but the application program determines whether the session will be terminated.

TYPE(UNCOND)

UNCOND is the default for TYPE. This coding causes an unconditional termination of the session. The application program is notified in its LOSTERM exit of the end of the session and must accept the termination. The program should issue a CLSDST macro instruction.

TYPE(FORCE)

With this coding, VTAM's SSCP immediately terminates the session and drives the application program's network services exit (NSEXIT) with a termination code. If the program has no NSEXIT, the LOSTERM exit is driven as with TYPE(UNCOND). FORCE may be tried when UNCOND fails to break the session.

HOLD(YES|NO)

The HOLD value tells VTAM what to do about deactivating the PU if the LU logging off is the last active LU on the PU. The default is NO. NO indicates that the PU is to be deactivated; that is, resign the connection between PU and VTAM. YES indicates that the PU is not to be deactivated; that is, hold the connection between the PU and VTAM). If HOLD(YES) is specified, the LU operator can issue a logon request to another (or the same) VTAM application program.

However, if the DISCNT (disconnect) parameter of the PU statement specifies DISCNT = YES, the PU will be deactivated, regardless of the HOLD value specified in the LOGOFF command. As a rule, therefore, the systems programmer codes DISCNT = NO in the PU statement, thus giving the LU operator the option of specifying whether the PU connection should be broken.

Sample USS LOGOFF Table

Here is a portion of a USS table showing a LOGOFF command:

```
USSCMD  CMD=OFF,REP=LOGOFF,FORMAT=BAL
USSPARM PARM=P1,REP=TYPE,DEFAULT=COND
USSPARM PARM=P2,REP=HOLD,DEFAULT=YES
```

With this code included in the USS table for a given LU, the LU operator, by keying the word OFF, can cause VTAM to log the LU off the VTAM application program. TYPE = COND, specified in the table, means that the OFF logoff request causes a conditional logoff request; that is, the application program may ignore the logoff request. The table default of HOLD = YES indicates that (if the PU definition statement allows it) the PU to which the LU is attached is not to be deactivated even if this logoff means that there are no LUs currently in session.

Overriding any of the parameters by the terminal operator is allowed for USS LOGOFFs, just as they are for USS logons. So, if the operator wants to log off using the OFF command shown above, but wants the logoff request to be unconditional, the USS table may be keyed as follows:

OFF UNCOND

In this example, since the operator does nothing to change the HOLD default, it remains YES.

The rules for overriding logoffs are identical to those which were considered for overriding logons. For example, there are two formats for the USSCMD: BAL and PL1. Also, there are two types of parameters—positional or keyword. Thus, there are four possible combinations for logoff, just as there are four possible combinations for logon.

■ ■ ■

Please turn to Mini-Course 10, Exercise 10.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 11

Basic VTAM to NCP Requirements

Mini-Course 11. Basic VTAM to NCP Requirements

Introduction

The function of VTAM is to manage the network. VTAM's operating system facilities initiate all network activity at the host. When using a channel-attached communications controller with NCP, the NCP takes no action except when requested by the host VTAM. For example, if the NCP definition of initial status for a PU is ACTIVE, this does not cause the NCP to send it an activate physical unit (ACTPU) command. First, VTAM must instruct NCP to send the ACTPU.

The NCP is always prepared to receive a request from the host. When the NCP has data for the host (or a response to a host command), the NCP issues an attention interrupt to the host. The NCP then must wait for VTAM to issue a read channel program. If the NCP has additional data to send at the end of the read, it may set read attention asking for an additional read channel program. There is an additional way that the NCP can ask for service; it can set the status modifier bit at the end of a host channel write program.

VTAM and Channel-Attached NCPs

Figure 11-1 illustrates how VTAM communicates with a channel-attached NCP over the single channel interface.

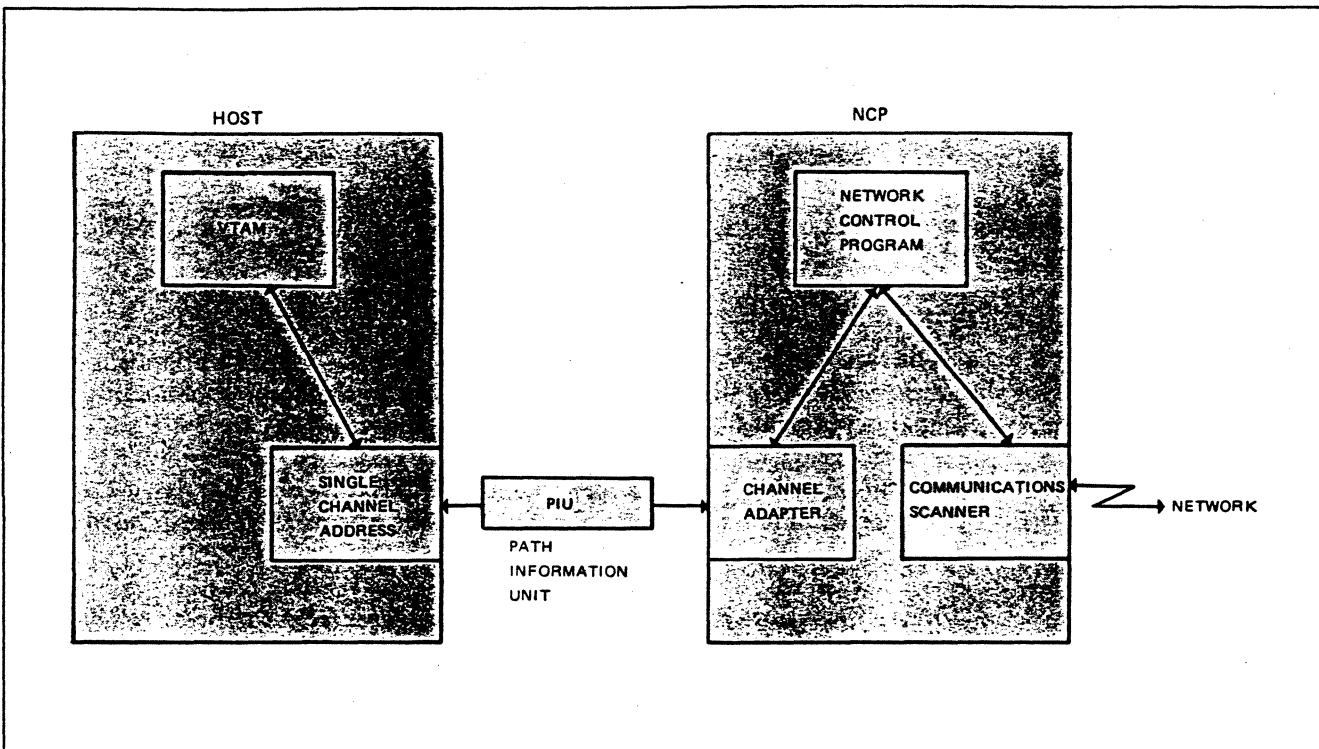


Figure 11-1. Host to NCP Communication

The unit of communication between the host and the NCP is the *Path Information Unit (PIU)*. For traffic outbound from the host, VTAM creates a PIU request and sends the PIU across the channel to the NCP.

Some PIUs from the host contain normal data, intended for an LU. Other PIUs request the services of the NCP to initiate, maintain, or restart some part of the network.

Inbound traffic comes into the NCP from a terminal, an LU, or from a link-attached NCP. The inbound traffic may consist of SNA commands (such as an INITIATE-SELF), a response to an earlier VTAM PIU, or a PIU intended for VTAM's SSCP or an application program.

When the channel-attached NCP receives inbound data from the network, it analyzes the data, determines what to do with it, creates an appropriate PIU (for example, conversion to a format identifier type 4 (FID4) transmission header (TH)), and issues an attention interrupt to notify the host that it has data.

As soon as possible, VTAM issues the read channel program. The PIUs are read into VTAM's buffers (IOBUF in MVS or VM, LFBUF in VSE). VTAM then takes the necessary action by scanning the PIU. For example, if the destination address is that of an application program, VTAM will look to see if the application program has an outstanding request for inbound traffic. If VTAM finds a request, it will then move the data to the work area specified by the application program.

Link-Attached NCP Communications Controllers

A *link-attached* communications controller is, by definition, one that is not directly attached to the host. A link-attached NCP resides in a communications controller that is connected by an SDLC data link to a channel-attached (local) communications controller or to another link-attached communications controller.

Communications between VTAM and the link-attached NCP must be directed (routed) through the local NCP. The procedure is the same whether the communication is inbound to VTAM or outbound to the link-attached NCP.

First, the local NCP analyzes the PIU for a destination subarea number. If the destination subarea number is the local NCP, it will search its tables to find the proper node to forward the PIU. If the destination subarea number is for another NCP, the local NCP searches its PATH table for an entry for that subarea. When the subarea is found, the NCP then checks the explicit route number (ER#) in the PIU to see if it is defined for the destination subarea. If the route is available, the NCP adds the SDLC framing characters and forwards the PIU to the link-attached NCP.

If there are intermediate subareas involved, such as another link-attached NCP or perhaps a host, the NCP forwards the PIU to the next subarea (adjacent subarea) in the route. The intermediate subarea would follow the same procedure to determine where the PIU is destined and how to forward it.

Note: In a multisystem environment, a channel-attached NCP may be an intermediate routing node (IRN) for data flow. In that case, the NCP forwards PIUs not destined for its channel-attached host directly to adjacent subareas, bypassing its own host.

NCP Program Support

NCPs Prior to NCP Version 3

The NCP is defined with two levels of macros, Stage 1 and Stage 2. The Stage 1 generation macros are user coded and assembled. NCP contains a 37xx assembler that may be used for this purpose. (Assembler H may be used in an MVS environment.) The output of Stage 1 assembly is a job stream of multiple assemblies with Stage 2 macros, constants, and linkage editor steps. Stage 2 assemblies generate the 37xx instructions.

NCPs at Version 3 and Higher

Although NCP definition coding at Version 3 and higher remains the same as with prior versions, the generation process has been simplified. Advanced Communications Function for System Support Programs (ACF/SSP or SSP) Version 3 includes an NCP/EP Definition Facility (NDF) which directly produces an NCP object module. The module is link-edited into the NCP load module library.

NCP Loading

A communications controller can be loaded directly from the linkage editor output file. In VSE, the linkage editor output may be moved from the phase library to a sequential disk file before loading if there is insufficient VTAM storage to contain the entire NCP load module.

Figure 11-2 on page 11-5 illustrates the generation, loading and dumping of an NCP.

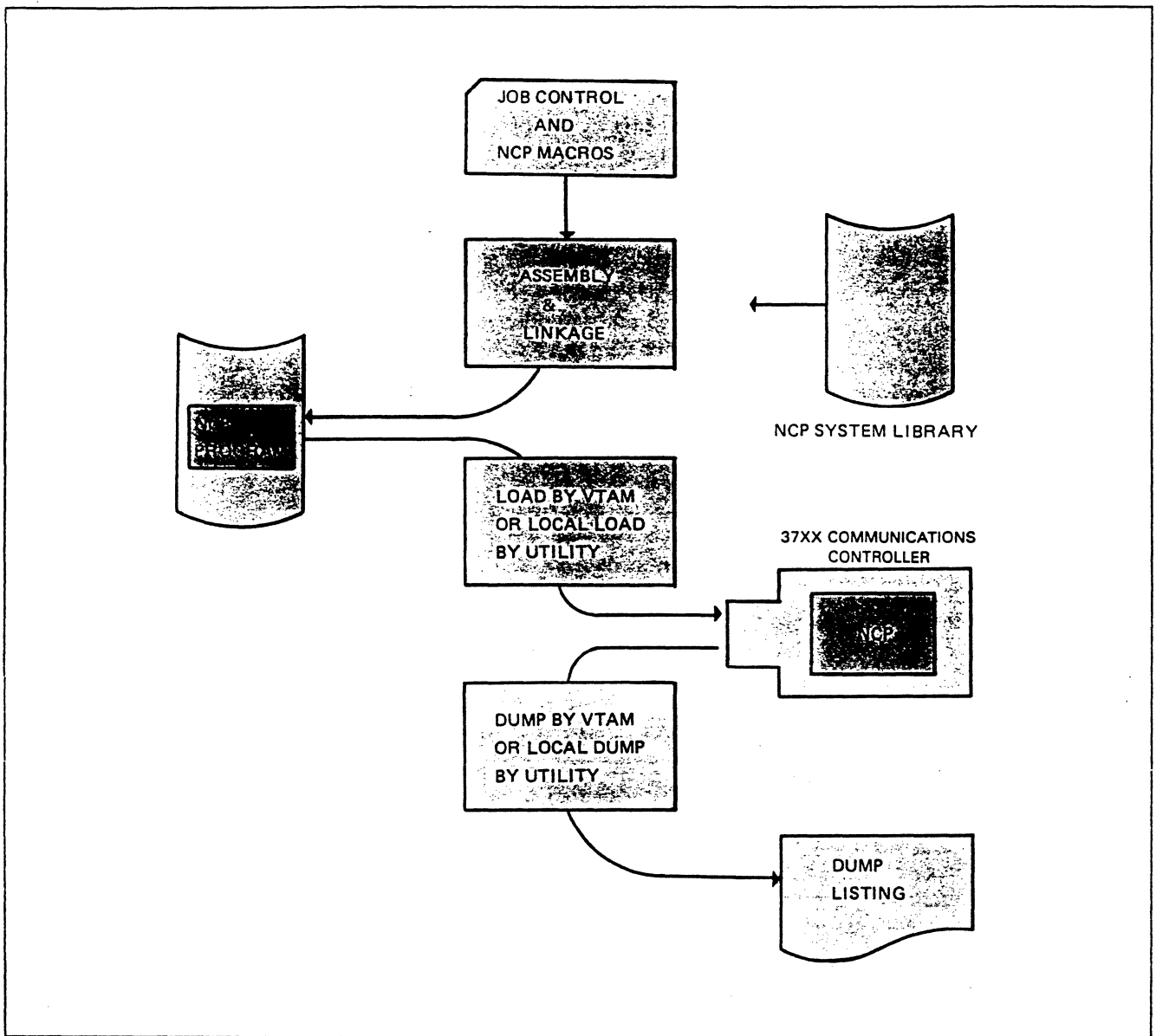


Figure 11-2. NCP Generation and Utilities

The input to the 37xx assembler or NDF is the job control and the NCP user coded macros. The macros are expanded from the NCP system library to create the NCP executable code.

A channel-attached 37xx can be loaded in one of two ways:

1. If VTAM is active, the network operator can load the NCP with the following command:

```
VARY NET,ACT,id=ncpname,LOAD=YES
```

LOAD=YES causes an unconditional load of the id=ncpname NCP regardless of the current status of the communications controller. This is the most common way to load a channel-attached communications controller, and the only way to load a link-attached communications

controller. Automatic activation of the NCP via a CONFIG list is comparable to the VARY ACT command; however, the opportunity to override coded parameters is not available with CONFIG.

2. The NCP loader utility program can be executed in the host operating system.

The NCP load module always remains available as long as the communications controller is powered on. Therefore, once loaded, the NCP is ready for communications regardless of the status of VTAM. For example, if the 37xx has an NCP loaded and VTAM is terminated, VTAM may be restarted and establish communications with the NCP without the necessity of reloading the NCP.

The following are examples of when an NCP might require reloading.

- A different named NCP load module is needed.
- A newly generated NCP with the same name is needed.
- There has been some form of system failure.
- Whenever the communications controller is powered off or reset.

A channel-attached or link-attached 37xx can be dumped to provide a hexadecimal storage listing, with formatted control blocks, registers, and mnemonic operation codes. Link-attached NCP dumps are via a local 37xx and VTAM, while a channel-attached NCP dump can be obtained either through VTAM commands or with an operating system NCP dump utility.

VTAM/NCP Channel Operations

Once loaded and ACTIVE to VTAM, the NCP is available for data transfer. When the NCP has data to transfer, and presents the attention interrupt, VTAM may respond with a WRITE channel program along with the requested READ. Therefore, the NCP must be ready to accept either command sequence at all times.

An NCP plans channel transfer according to VTAM requirements. Every new PIU from the NCP must start on a VTAM buffer boundary, due to the way that hardware commands and responses are used. If the incoming PIU will not fit in a single I/O buffer, VTAM will use additional buffers until the entire PIU is read.

VTAM will dynamically allocate a fixed number of inbound buffers for each channel-attached NCP according to a user-defined NCP generation operand (MAXBFRU). After an inbound channel operation has terminated, VTAM separates the PIUs for analysis and releases all unused buffers which had been allocated. A new set of buffers are allocated for the next attention interrupt from the 37xx.

Two different channel commands are used, alternately, for transfers between the host and NCP. This odd/even approach is used by the intended receiver to discover if a transfer has been lost. The READ or WRITE channel program can then be repeated.

NCP Macros

There are three types of user coded macros with operands required in NCP generation for a VTAM system. They are VTAM-only, VTAM and NCP, and NCP-only.

- VTAM-only macros and operands are read by VTAM when the NCP major node is activated. These operands are allowed to be in the NCP generation source deck; however, the NCP assembly process disregards them and generates no NCP code. (Selected VTAM-only operands were discussed in Mini-Course 7.)

IMPORTANT: Since these macros and operands are read and used only by VTAM, changes to any of these operands can be made as follows, with one exception as discussed later. Change the source deck and refile it in the VTAM definition library. Inactivate the NCP if it is active to VTAM, then reactivate it (no load is required), and the changes will be in effect.

- VTAM and NCP macros and operands generate code in the NCP generation process and are used by VTAM when the NCP is activated. Changes to any of these operands will require a new NCP generation.
- NCP-only macros and operands are used during the NCP generation process and result in NCP code. When the NCP major node is activated (that is, the NCP source deck is read from the definition library), VTAM ignores all of the NCP-only macros and operands as it builds the resource definition table.

■ ■ ■

Please turn to Mini-Course 11, Exercise 11.1, in your PRG and answer the exercise questions.

VTAM-Only Code in NCP

In addition to VTAM-only operands which may be coded on many of the NCP macros, there is one NCP macro, PCCU, that is used entirely by VTAM and produces no NCP generated code.

PCCU

The PCCU macro is coded in the NCP to identify the VTAM functions to be provided for the NCP. It must appear as the first macro in the NCP generation deck. PCCU operands define to VTAM selected user options that VTAM is to use when the NCP is activated.

If more than one VTAM host is to activate the same NCP, then multiple PCCU macros would be coded in that NCP. Each VTAM would then select the proper PCCU macro by searching for its own SUBAREA number (a PCCU macro operand) at activation time.

PCCU Macro

Each host VTAM connected to a channel-attached 37xx that will activate the NCP and any of its resources must have a PCCU macro included in the NCP source deck.

As described earlier, with one exception, any changes that the user wants to make in VTAM-only operands do not require a regeneration of the NCP load module. The exception is SUBAREA = which may require regeneration. All other PCCU operands do not incur this requirement.

PCCU operands which are used by VTAM in a single-domain environment with a channel-attached NCP communications controller are:

SUBAREA = n

This identifies the SUBAREA number which is used by VTAM to identify its own PCCU macro at NCP activation time. It must be coded with the same number used in the VTAM start list parameter HOSTSA = when VTAM is initialized.

SUBAREA = may be changed in the PCCU macro if there is a need to change the host subarea number; that is, if HOSTSA = is changed for some reason. The NCP regeneration exception occurs only when there is no NCP HOST macro with a corresponding SUBAREA = number for a channel-attached host. Changes to the host subarea number and the effect on NCP regeneration are as follows:

Channel-attached NCP:

HOSTSA start option = NCP PCCU and HOST SUBAREA operands

Regenerate NCP if the new subarea number is not in an existing NCP HOST macro or if there is no valid PATH statement in the NCP for the new host subarea number.

Link-attached NCP:

HOSTSA start option = NCP PCCU SUBAREA operand

Regenerate NCP only if no existing PATH statement in the NCP contains a valid route to the new host subarea number.

For the above reasons, VTAM network design should attempt to establish unique subarea numbers which will not require future changes.

Note: If SUBAREA= is not coded in a PCCU macro, VTAM can not find a match for its own subarea number. In this case, VTAM will use the PCCU macro with no SUBAREA= operand. This allows multiple hosts to use the same PCCU macro. However, this procedure may require the VTAM operators to override certain operands when activating the NCP. *It is recommended that a PCCU macro with SUBAREA= be coded for all hosts that will activate the NCP.*

CUADDR = cua

This identifies the channel unit address of the attached communications controller for this NCP. All channel programs, including loading the NCP, are directed to this address by VTAM. The address may be supplied (or overridden) by the VTAM network operator when issuing the VARY ACT command for this NCP.

MAXDATA = size

This tells VTAM the size of the largest PIU that it may send to this NCP in a single WRITE channel program. This operand applies to outbound traffic from VTAM to a channel-attached NCP.

AUTOSYN = YES|NO

Tells VTAM whether to automatically activate the NCP if the name of the NCP in the controller matches the name of this NCP major node. (A loaded NCP informs VTAM of its name during the initial contact procedure.)

If AUTOSYN=NO is coded, the VTAM operator is asked whether the NCP is to be reloaded, even though it is already loaded with an NCP of the same name. For example, a new NCP load module with the same name has been generated and it needs to be loaded.

Another example: AUTOSYN=NO might be used in a test environment, then later changed to AUTOSYN=YES for a production environment.

VFYLM = YES|NO

This operand is used by VTAM when the name of a previously loaded NCP does not match the name of this NCP major node. Coding YES or NO determines whether VTAM notifies the operator before performing a reload of the NCP.

When VFYLM = NO is coded, or allowed to default, VTAM will reload the NCP without operator intervention.

In an environment with more than one named NCP load module for the same communications controller, a better choice might be to code VFYLM = YES. In this case, if the two names do not match, VTAM will ask the operator to confirm the NCP reload.

INTEST = YES|NO

Tells VTAM whether to load a diagnostic routine (supplied with the NCP) into the communications controller before loading the NCP load module. This operand only applies to channel-attached 3705 controllers. The test routine checks for any machine malfunctions in the 3705.

CHANCON = COND|UNCOND

UNCOND allows VTAM to force NCP to break contact with another host if the other host has the same subarea number as this VTAM (PCCU) host.

The default value of COND allows NCP to reject a contact request from this host if it is in session with another host with the same subarea number.

Normally, a VTAM network will be designed to have unique subarea numbers in all hosts (even if there are other access methods, such as ACF/TCAM, in the network). With unique subarea numbers, CHANCON has no effect during the contact procedure and CHANCON is left to default to COND.

Other PCCU Macro Operands

Several other operands which may be coded on the NCP PCCU macro can be grouped into general categories for user selection:

AUTODMP =
AUTOIPL =
DUMPDS =
DUMPSTA =

These four operands tell VTAM what to do in case of an NCP failure that requires a dump of NCP storage. For example, if the user selected operand values as follows:

PCCU ,AUTODMP=NO ,AUTOIPL=NO ,DUMPDS=xxxxxx

the network operator would be prompted before a dump of NCP storage was attempted automatically due to AUTODMP=NO. Also, VTAM would not try to reload the NCP after any dump (requested or automatic) due to AUTOIPL=NO. If the operator decided to have VTAM dump the NCP, the storage dump would go to DUMPDS=xxxxxx where the xxxxxx would be a data definition name in an MVS system, a filename in a VM gobal loadlib, or a DLBL filename in VSE. Without DUMPDS, no NCP storage dump could be taken.

For an IBM 3725 Communications Controller, two additional dump files may be setup. Data from the communications scanner processor (CSP) may be directed to a file pointed to with the CDUMPDS= operand. Maintenance operator subsystem (MOSS) dump data may be sent to a file pointed to with MDUMPDS=.

LOADSTA=

The LOADSTA= operand would normally be coded for a link-attached communications controller to provide VTAM with a link station name in an adjacent NCP to load that controller's NCP. When the station name is not provided, VTAM will attempt to choose a default link station. The operator may provide a LOADSTA name with the VARY ACT command for the NCP.

RNAME=

RNAME is coded in link-attached NCPs to provide VTAM with contact points (link stations) for communications. RNAME must indicate a valid link station name coded in one or more NCPs which are adjacent to the link-attached NCP. For example, a channel-attached NCP, LOCNCP, contains a PU (PUTYPE=4) macro to represent a link-attached NCP, LNKNCP. The symbol (or name) on the PU macro is coded LOCLNK. This provides VTAM with a contact point, LOCLNK in LOCNCP, to communicate with LNKNCP. In this case, RNAME=LOCLNK would be coded in LNKNCP's PCCU macro.

An alternative to coding RNAME is to provide the link station name on the VARY ACT command for the link-attached NCP.

CONFGDS=

CONFGPW=

The configuration restart operands are coded only if the user has chosen to have VTAM attempt to reinstate the network status after a system failure. Configuration restart will only try to recover the ACTIVE/INACTIVE status of network components, and not attempt to recover any LU-LU sessions that may have been broken.

NCPLUB = for VSE

In a VSE environment, the NCPLUB operand must be coded. VSE needs to know where to look for the NCP load module phase when the NCP requires loading. NCPLUB is the NCP phase or sequential file lubname (SYSxxx) from the EXTENT or ASSGN statements for the load module.

OWNER =**BACKUP =**

The OWNER and BACKUP operands apply when more than one VTAM is to own resources belonging to this NCP. Ownership applies to all resources on single lines or groups of lines. The OWNER name, which is any user chosen host identifier, would be compared to all OWNER names coded on LINE or GROUP macros within the NCP. Those resources that have a matching OWNER name would belong to that VTAM host. Therefore, the user can pre-allocate resources to a particular VTAM.

If the user wanted the other hosts to be aware of those resources, BACKUP= YES would be coded on their PCCU macro. Then, the backup host could become the owner of the resources via the VARY NET,ACQ command if the original owning host lost the resources for some reason.

If OWNER and BACKUP are not coded on the PCCU macros, all of the NCP resources are available to each host based on the share limits of the resource.

Share Limits for NCP Resources:

An NCP itself is sharable by up to eight hosts; that is, it may be ACTIVE (owned) by more than one host at a time. Communications links connected to the NCP may also be shared resources.

At the PU/LU level of resources, only one host at a time may activate the PU and its LUs for ownership. This does not mean that the LUs cannot be in LU-LU sessions with multiple hosts concurrently, since they may be defined in a multisystem environment as cross-domain resources.



Please turn to Mini-Course 11, Exercise 11.2, in your PRG and answer the exercise questions.

The NCP HOST Macro

The NCP must know the characteristics of each access method with which it will communicate. A HOST macro is provided for each channel-attached VTAM host that will activate the NCP and own any of its resources.

Note: Link-attached NCPs may contain HOST macros for other hosts in a multidomain environment.

The HOST macro information includes:

The VTAM subarea number

VTAM's I/O buffer structure

The number of NCP buffers reserved for VTAM outbound PIUs.

SUBAREA =

At activation time of the NCP major node, each VTAM host will search for its own HOST macro by looking for a match between its own subarea number and SUBAREA = on all HOST macros in the NCP.

Once VTAM finds its HOST macro in the NCP major node, the coded values are sent to the NCP during activation processing. In turn, the NCP will send to VTAM its equivalent information. This is referred to as the XID exchange.

UNITSZ =

The length of the VTAM I/O buffers is specified in the UNITSZ = operand of the HOST macro and it must be equal to or less than the size supplied to VTAM in the start list at initialization time. For example, if the VTAM I/O buffer size value in the start option is specified as

```
IOBUF=(...,128,,,,,.....)      in MVS or VM
                                or
LFBUF=(...,128,,,,,.....)      in VSE
```

UNITSZ=128 would be coded in the HOST macro of the NCP.

MAXBFRU =

The NCP also needs to know how many total buffers VTAM will set aside for each READ from the NCP. This number is specified as MAXBFRU = in the HOST macro.

The product of MAXBFRU times UNITSZ must be large enough to hold the maximum size PIU that will ever flow from the NCP to VTAM. For example, if the HOST macro for HOST1 were coded as:

```
HOST    ...,MAXBFRU=10,UNITSZ=100,SUBAREA=1,...
```

the NCP could send, in one READ, ten 10 PIUs to HOST1 if they were all equal to or less than 100 characters long, or it could send one PIU that was 1,000 characters long.

If a HOST macro were added for HOST2, in the above example, MAXBFRU and UNITSZ could be different than those for HOST1.

Outbound PIUs from VTAM to NCP

MAXBFRU and UNITSZ determine what NCP can send to VTAM in a single READ. The MAXDATA operand determines what VTAM can send to the NCP in a single WRITE. Recall that the PCCU macro specifies VTAM functions with respect to the NCP.

INBFRS =

Since the NCP must be prepared to receive data from VTAM at any time, it must also reserve a number of buffers for the expected WRITE. This number of buffers is specified in the HOST macro operand INBFRS=. If VTAM attempts to send more data than the NCP can hold in the first set of INBFRS, the NCP will allocate an additional set of buffers equal to INBFRS. In other words, NCP does not have to have buffers reserved for a total of MAXDATA characters, it will dynamically add buffers as needed.

Other VTAM Hosts

Other VTAM hosts may own resources, such as application programs, that are in cross-domain sessions with this NCP's resources and not be represented by a HOST macro. These other hosts in the network would communicate with the NCP resources via cross-domain resource managers belonging to one or more of the NCP's owning hosts.

VTAM/NCP Operands on Other NCP Macros

In addition to the VTAM-only operands described in Mini-Course 7, there are other operands that are used by VTAM and NCP. These operands appear in the NCP generation deck depending on such things as line control, SNA or non-SNA terminals, switched connections, and NCP polling. In some cases there are VTAM restrictions on the operand values that may be coded. The NCP system programmer should become familiar with any restrictions imposed by VTAM.

■ ■ ■

Please turn to Mini-Course 11, Exercise 3, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 12

VTAM Communications Adapter Support in VSE and VM

Mini-Course 12. VTAM Communications Adapter Support in VSE and VM

VTAM and the Communications Adapter

Support of a communications adapter (CA) is limited to the IBM 4321, 4331, and 4361 Processors running under VSE or VM. Unlike VTAM with an NCP, the support for devices attached to CA lines is a combination of the communications adapter hardware and VTAM program modules. *All CA lines and devices belong to the VTAM domain in which they are defined; that is, in the VTAM subarea.* All communications lines and devices are defined in a VTAM channel attachment (CA) major node.

The CA Major Node

The channel attachment major node is defined with the VTAM statement as follows:

```
[name] VBUILD TYPE=CA
```

There are four types of minor nodes within a CA major node for remote SNA terminals:

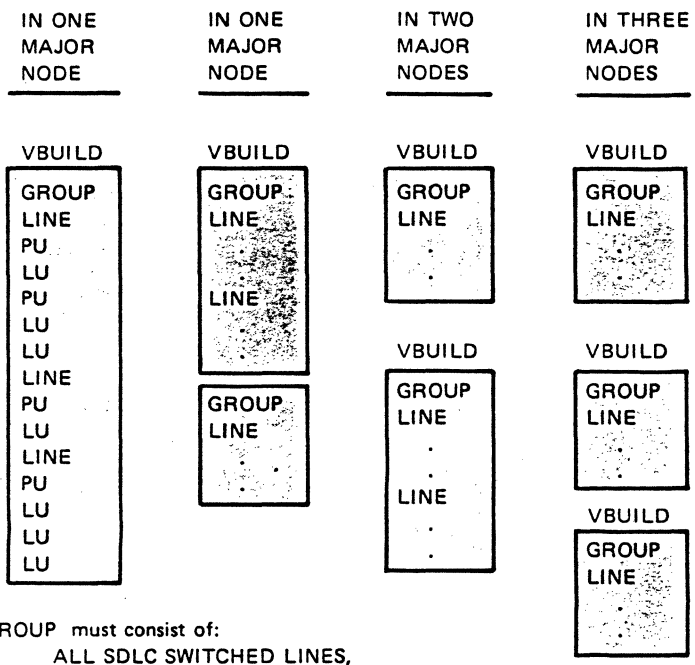
- Groups of lines
- The lines themselves
- The physical units
- The logical units.

The minor node names are assigned with the name fields on the GROUP, LINE, PU and LU statements. There may be multiple GROUP statements within a single major node, each followed by one or more LINE statements. This allows all communication adapter lines to be in a single major node definition. In addition, they may be set up by grouping one or more lines together.

All LINE statements within a single GROUP must represent the same line discipline; that is, either SDLC or BSC. Also, each group of lines must be either switched or nonswitched. The PU statements and their corresponding LU statements (for SNA terminals) would follow each LINE statement shown.

For example, three SDLC lines may be set up in different ways as shown in Figure 12-1. (Note: BSC lines and their corresponding CLUSTER and TERMINAL statements will be discussed in Mini-Course 13.)

DEFINITION STATEMENT VARIATIONS FOR 3 COMMUNICATIONS LINES



Note: A GROUP must consist of:
 ALL SDLC SWITCHED LINES,
 ALL SDLC NONSWITCHED LINES,
 OR
 ALL BSC NONSWITCHED LINES.

Figure 12-1. Variations of CA Major Node Definition

LINE Definition Notes

The LINE statement defines the minor node name for the communication line. It must have at least the ADDRESS operand coded unless the line address is 030 (the default). Up to eight lines may be installed on the CA. The lines have fixed channel addresses from 030 to 037.

MAXBFRU

MAXBFRU describes the number of I/O buffers needed to read data from the line. Two values are specified for the particular device:

- The normal/minimum I/O buffers needed.
- The maximum buffers required.

The second value should be large enough to handle the maximum amount of data that any PU on the line can transmit at one time. VTAM will attempt to allocate the maximum number of buffers depending on availability.

A reasonable choice for MAXBFRU is the average message size transmitted from the PU to VTAM divided by the I/O buffer length (specified in the VSE LFBUF or VM IOBUF start option) rounded up to an integer.

PAUSE REPLYTO RETRIES SERVLIM

These four operands tell the communications adapter and VTAM how to service the PUs on the line.

A *data poll cycle* is a request for each active PU on a line to send data. The requests are made in the order of PU statements coded under the LINE statement.

After the CA has completed a data poll cycle with no station having any data to send, it will wait a specified time before starting the next cycle. The wait time is defined by the user in tenths of a second with the PAUSE operand. For example, if there is heavy traffic expected on the line, a very short PAUSE would be appropriate (0.1 or 0.2 seconds). If perhaps there is only one station on the line with a low volume of traffic, a longer PAUSE could be given.

As stations become active, they are serviced as part of the data poll cycle, leaving the inactive stations still to be contacted. A contact poll cycle addresses those stations that are not yet active in the SDLC polling list.

The user may specify that active stations (data poll cycle) be serviced more frequently than inactive stations (contact poll cycle) with the SERVLIM operand. For example, a SERVLIM value of 5 would cause active stations to be serviced every cycle through the table and one inactive station to be polled every fifth time through the table.

Another form of CA waiting time is the *reply timeout value* (REPLYTO) which is the time given to a station to respond to a poll. The CA presents to VTAM an error condition (idle detect timeout error) when the REPLYTO time has expired without a response. VTAM will reissue the request as many times as specified in the

RETRIES operand. In a practical situation, an active station is expected to respond in a matter of tenths of a second (reply *no data to send*, for example). Therefore, nominal values such as the defaults (REPLYTO of one second and 7 RETRIES) are sufficient. RETRIES also applies to other types of errors during transmission. Normally, if an error, such as a line error, isn't corrected after the second or third retry, it is a permanent type of error. Although RETRIES can number as many as 255, a value between 4 and 7 is usually sufficient.

PU Definition Notes

The PU statement requires at least a minor node name and the SDLC station address.

MAXDATA

The MAXDATA operand specifies the maximum number of bytes that the PU can accept in one data transfer including the TH and RH. MAXDATA is not the device buffer capacity; it is the device buffer size plus the TH length and RH length. Most, but not all, devices have a 256-byte buffer. The defaults correspond to the PU type: 261 bytes for PU__T1 and 265 bytes for PU__T2.

For example, the 3767, a PU__T1, can accept 261 bytes (up to 256 data/RU bytes plus a two-byte TH and a three-byte RH), while an SNA 3274 control unit PU__T2 can accept 265 bytes (up to 256 data/RU bytes plus a six-byte TH and a three-byte RH). An exception with PU__T2 is an SDLC 3276 control unit which must have MAXDATA defined as 262.

MAXOUT PASSLIM

The MAXOUT operand specifies the maximum number of SDLC frames (a PIU or a PIU segment) to be sent to a PU before requesting a response from the PU. MAXOUT must be a value between 1 and 7. If the total number of PIUs to be sent is less than the MAXOUT value, the request for response will be sent at that point. That is, MAXOUT is used only when the total number of PIUs to be sent is greater than the specified MAXOUT value.

A value for MAXOUT may be specified according to line quality. With a high quality line, MAXOUT may be set to 7 if the terminal can accept 7 frames, or it may be set to a lower number if line errors require frequent retransmissions.

Along with MAXOUT, PASSLIM specifies the number of SDLC frames to send before servicing the next PU in the list. For example, an IBM 3790 can accept several SDLC frames at a time. If MAXOUT is specified as 7 and PASSLIM is specified as 3, only three frames would be sent before servicing the next PU, and the 3790 would wait a full cycle to receive three more frames. After 7 frames, VTAM would request a response from the 3790.

PASSLIM may be any value from 1 to 254. The user may want to make PASSLIM a value greater than MAXOUT to allow special

servicing of a particular PU, however, other PUs in the list would be serviced less frequently.

VTAM will default PASSLIM to the value of MAXOUT if it is not specified.

LU Definition Notes

The LU statement requires the minor node name and the local address of the logical unit. The LU local addresses are assigned at installation time and correspond to the control unit addressing scheme.

LOGAPPL

When the logical unit becomes active, VTAM will automatically log it on to an application program if LOGAPPL is specified. The name specifies the application program minor node name given on an APPL statement.

PACING

Pacing controls the flow of data in the network, specifically for LU to LU sessions.

The important thing about pacing is that it prevents a sender from transmitting data faster than the receiver can handle it. It is a form of acknowledgment by the receiving LU that it is ready for the next transmission.

Pacing applies to a logical unit. The PACING value specified is the number of normal flow requests (PIUs or PIU segments) to be sent by VTAM before waiting for a pacing response from the LU.

Note: The MAXOUT value discussed previously applies to a PU. For example, a physical unit, such as a cluster controller may receive PIUs for more than one LU before sending the MAXOUT response. Also, since PACING is a request for response, the PASSLIM parameter could be set equal to PACING, rather than MAXOUT, which would allow other PUs to be serviced while waiting for the pacing response.

SSCPFM

The SSCPFM parameter can sometimes cause confusion. Code SSCPFM = FSS for programmable LUs (such as the IBM 8100) which can transmit the SNA commands INITIATE-SELF and TERMINATE-SELF. Code SSCPFM = USSSCS for nonprogrammable LUs (such as on the IBM 3274 Control Unit) which cannot transmit those SNA commands.

SNA 3270 display stations and printers (the logical units) require SSCPFM = USSSCS while 3790 logical units require SSCPFM = FSS. SSCPFM was also discussed in Mini-Course 7.

CA and NCP in a Single-Domain

A communications adapter (CA) may coexist in the same domain with 37xx NCP communications controllers with the following restrictions:

- An NCP cannot be link-attached to a CA line in the same domain.
- An NCP cannot be loaded or activated via the CA .

For example, in a migration from VTAME with a CA to VTAM and NCP, nondisruptive testing is possible since both the CA and the 37xx are supported.

If both the CA and an NCP are link-attached to an NCP in another domain, the link-attached NCP cannot be contacted through the CA and active through the channel-attached NCP at the same time.

Multisystem Networking and the CA

VTAM support of a communications adapter includes participation in a networking environment with other domains managed by the following communications access methods:

- VTAM with a CA (peer to peer connection)
- VTAME and a CA
- VTAM and NCP
- TCAM and NCP.

In all of these cases, communications via the CA will always be cross-domain sessions.

A link between the CA and an NCP in another domain has the following restrictions:

- There may be only a single link between the CA and the 37xx.
- The data link configuration must be half duplex (HDX), point-to-point, nonswitched.

Routing functions are provided for ER/VR protocol over the single link.

Note: Routing will be discussed in Mini-Course 15 and cross-domain definition in Mini-Course 16.

Sample Coding - CA Major Node

An example configuration is shown in Figure 12-2.

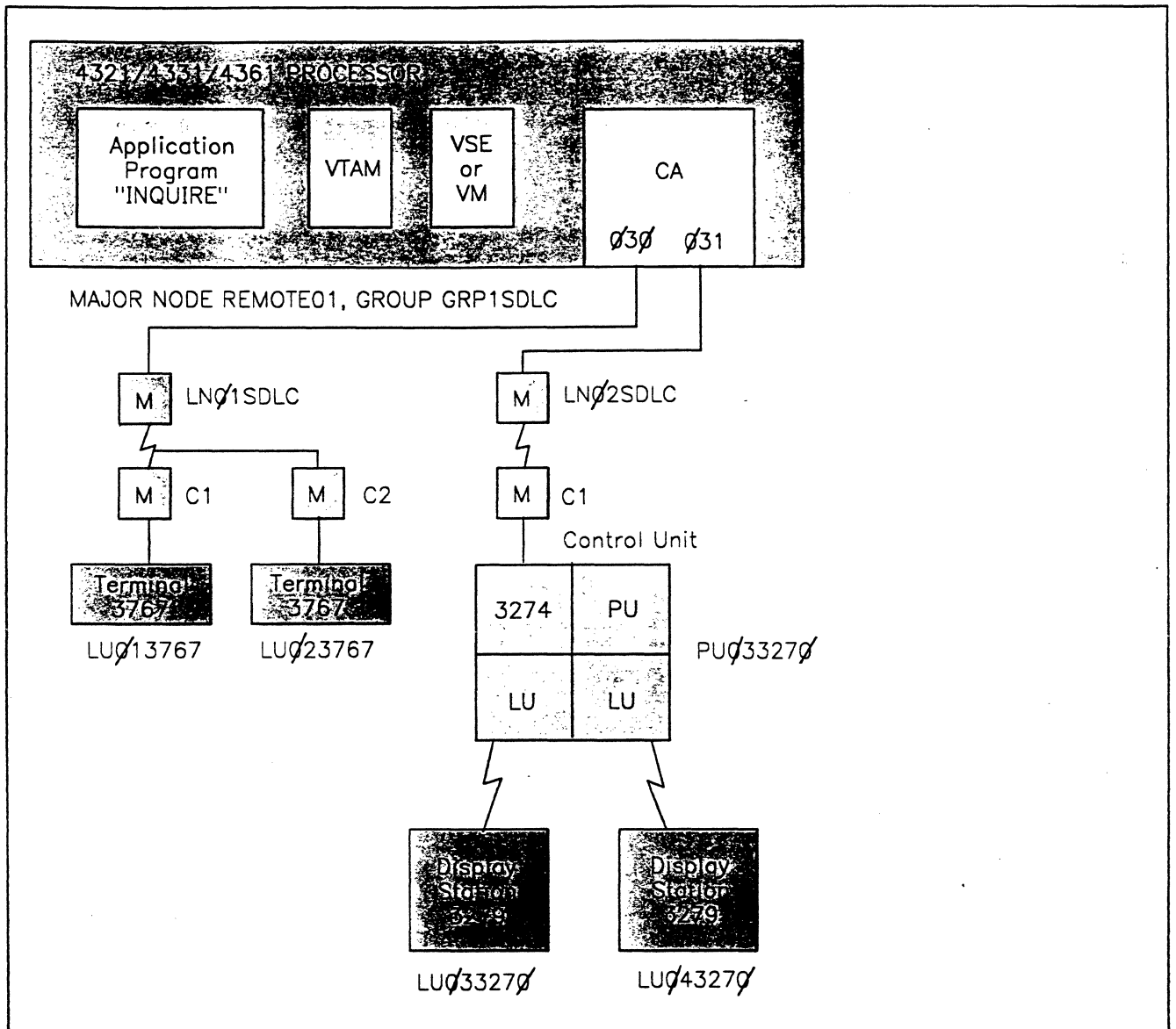


Figure 12-2. CA Example Configuration

The CA example configuration consists of:

- The IBM 4300 Processor with the communications adapter
- Two SDLC nonswitched communications lines
- Two IBM 3767 Communication Terminals (PU__T1)
- A 3274 Control Unit with two 3279 Display Stations
- VTAM running under VSE
- A VTAM Application Program with an APPL-name of INQUIRE.

The configuration diagram should be marked with information to be used during coding of the major node. In this case, line addresses, the 3767 and 3274 PU and LU addresses, PU type, major and minor node names, and line discipline should be added. A user may wish to include other information, such as physical locations, VSE or VM library names, and other notes relative to the environment.

Example Configuration Assumptions

For this mini-course, some assumptions have been made in order to keep the number of considerations at a minimum. The assumptions are:

- There is a single application program with an APPLID of INQUIRE in the access method control block (ACB).
- The logical units are to have user-coded USS tables available for logons to the application program. The table names are USS3767 and USS3279.
- Two logon mode tables are available, each having single entries, and are link-edited with names MODE3767 and MODE3279.
- The VSE or VM VTAM definition and VTAM module libraries are to be used.

The VTAM node definitions based on the configuration diagram in Figure 12-2 are coded in the CA major node shown in Figure 12-3.

REMOTE01	VBUILD	TYPE=CA		
*				
GRP1SDLC	GROUP	LNCTL=SDLC		x
		ISTATUS=ACTIVE	default	
*				
LN01SDLC	LINE	ADDRESS=030,	default	x
		MAXBFRU=(1,2),	default	x
		ISTATUS=ACTIVE,	default	x
		PAUSE=0.5,		x
		REPLYTO=1,	default	x
		RETRIES=4,		x
		SERVLIM=5		
*				
PU013767	PU	ADDR=C1,		x
		ISTATUS=ACTIVE,	default	x
		MAXDATA=261,	default for PU_T1	x
		MAXOUT=1,	default	x
		PUTYPE=1		
*				
LU013767	LU	LOCADDR=0,		x
		ISTATUS=ACTIVE,	default	x
		PACING=1,	default	x
		SSCPFM=USSSCS,USSTAB=USS3767,	(SSCP-LU session)	x
		MODETAB=MODE3767	(LU-LU session)	
*				
PU023767	PU	ADDR=C2,		x
		PUTYPE=1		
*				
LU023767	LU	LOCADDR=0,		x
		SSCPFM=USSSCS,USSTAB=USS3767,		x
		MODETAB=MODE3767		
*				
LN02SDLC	LINE	ADDRESS=031,		x
		PAUSE=0.5,		x
		MAXOUT=7		
*				
PU033270	PU	ADDR=C1,		x
		PUTYPE=2,	default	x
		MAXDATA=265,	default for PU_T2	x
		SSCPFM=USSSCS,USSTAB=USS3279,	for both LUs	x
		MODETAB=MODE3279,	for both LUs	x
		RETRIES=4		
*				
LU033270	LU	LOCADDR=2		
*				
LU043270	LU	LOCADDR=3		

Figure 12-3. Remote SNA - Sample CA Major Node

Notes on the Sample CA Major Node

- The CA major node definition for the example configuration is coded as a single major node.
- The design structure is based on the example requirements, although it could have been arranged in other ways, as shown previously in Figure 12-1.

- The default values shown in the example minor node LN01SDLC are for documentation and reference; coding them is not required. For example, all minor nodes will be activated with ISTATUS = ACTIVE (either coded or defaulted) when the major node is activated.
- The SSCP-LU session operands, USSTAB and SSCPFM, are shown in both PU and LU definitions as an example of sifting of operands.
- The PAUSE value of 0.5 means that VTAM will wait one-half second at the end of a normal (data) poll cycle if all stations report no data to send. PAUSE for a communications adapter works differently from an NCP SDLC line PAUSE. An SDLC line PAUSE in NCP refers to the entire service cycle time for all stations on the line.
- Another CA and NCP difference is in the service order. Definition for NCP includes a service order table when there are multiple stations on an SDLC link. No service order table is coded in a CA major node; the stations are serviced in the order they appear under each LINE macro.
- LOCADDRs of 2 and 3 on the 3270 LU definitions refer to port numbers A0 and A1 respectively on the 3274 Control Unit.

■ ■ ■

Please turn to Mini-Course 12, Exercise 12.1, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 13

Remote Non-SNA Support

Mini-Course 13. Remote Non-SNA Support

Introduction

VTAM will support the IBM 3270 Information Display System cluster controllers and terminals that use binary synchronous control (BSC) line discipline. It can support BSC 3270s in EBCDIC character format.

Other BSC terminals running in BSC 3270 mode are also supported by VTAM. For example, the IBM 8100 Information System may be remotely attached as a BSC terminal and supported by VTAM as a 3270 cluster controller (3271 Control Unit, Models 1 or 2).

The remote stations must be on nonswitched communications lines attached to a communications controller or a communications adapter (CA). Several 3270 cluster controllers may be connected to the same line. The exact number depends on the type and model of the control unit.

Although BSC line discipline is used for message transfer, VTAM provides functions which make the transmissions appear to be to (and from) logical units. The non-SNA devices are supported as PU_T1 physical units. VTAM becomes aware of this requirement when a remote non-SNA major node is activated. This function is transparent to any network considerations by the user.

Remote Non-SNA Definition

The remote non-SNA BSC lines, cluster control units, and terminals are defined in NCP or VBUILD TYPE=CA major nodes. VTAM makes the BSC 3270 cluster control units appear as physical units to the rest of the network. Once definition is completed, VTAM treats BSC cluster control units as physical units. VTAM also provides functions to make BSC 3270 terminals appear as logical units to the rest of the network.

Note: If an access method other than VTAM is supporting start-stop communications lines via a communications adapter, other lines must be either all BSC or all SDLC lines. The CA supports installation of only two of the three line disciplines at a time.

Major Node Definition

The BSC communications lines for non-SNA major nodes are defined with the same VTAM statements as for remote-SNA major nodes, namely: the GROUP and LINE statements in either an NCP or VBUILD TYPE=CA major node.

CLUSTER and TERMINAL statements are used to describe the cluster control units and terminals rather than the PU and LU statements.

In order to define the remote non-SNA configuration, certain information must be obtained from the equipment planner/installer. This information includes:

- The types of cluster control units.
- The control unit addresses on the line(s). Addresses are used to determine the general polling characters for each control unit.
- The types and features of terminals attached to the various control units.
- The terminal device address (used to determine the device selection character) by which it is known to the control unit.

CUTYPE in the CLUSTER Macro

Depending on the particular control unit installed, the CUTYPE is specified as either 3271 or 3275.

The 3275 is a single display station with its own control unit, which may have the optional printer attached, an IBM 3284, Model 3.

In addition to the BSC 3271 Control Units (Models 1 and 2), VTAM supports the BSC 3274 (C Models), and the BSC 3276 Models 1 through 4 Control Units. All are specified as CUTYPE = 3271.

TERM in the TERMINAL Macro

The TERM operand in the TERMINAL statement specifies the type of devices attached to the control unit. As with the control units, some common protocols are used, and the device number specified may not be the actual IBM device type number. For example, 3278 and 3279 display stations are specified as 3277. An exception to defining every physical device with a TERMINAL statement is when a 3284 printer is attached to a 3275 display station. The 3284 is specified with an operand, FEATUR2=PRINTR, on the TERMINAL statement for the 3275.

POLLING

Polling refers to a request for any inbound traffic from a BSC 3270 control unit. There are two types of polling in the BSC 3270 environment. *General polling* refers to a request asking the control unit if it has sense and status information or a manually entered message to send for any of its devices. The second type of polling, *specific polling*, refers to asking the control unit if a specific device has sense and status information or messages to send.

When not executing a command operation (text mode), the control units monitor the communication line for a valid selection or poll addressing sequence (control mode). Upon receiving a general poll with its address sequence, the control unit examines each attached device in sequence (starting with a random address) to determine whether the device has sense and status or a manually entered message waiting for transfer. (The 3274 control unit running in BSC mode examines its devices in the order that an ENTER key was pressed.)

The current status of each device is recorded in the associated device adapter in the control unit. When an I/O pending condition is detected at a device the following occurs:

1. Internal examination stops.
2. The reply to the general poll is made positive.
3. The sense and status or message for the device is sent to the host.
4. The control unit communicates solely with that device until transfer is completed, that is, additional general or specific polls are not needed even though multiple transmissions may occur for the device. When transmission is ended, the control unit examines the remaining devices for pending messages.
5. When no device has any pending message, the control unit returns to control mode.

The operation of the BSC 3270 control units is essentially the same when a specific poll is received. The control unit will automatically send sense and status or a manually entered message on behalf of the specified device.

Selection Addressing

When VTAM needs to send to a BSC 3270, it identifies the device to the 3270 control unit with a *selection addressing sequence*. This sequence is like a specific poll except the control unit address character differs from that used in the specific poll. This sequence is used for write-, control-, or read-type commands. In practice, reading is normally done with polling while address selection must be used for write- and control-type commands.

GPOLL in a CLUSTER Macro

Each 3270 control unit on a line is given a unique numeric address at installation time. The numeric address is in the range of 0 to 31 starting with 0 for the first control unit on the line. This hardware address is used to determine an address character to be used in command and data transmission. The address character for the 3270 control unit is selected from a published table according to the control unit address (0 - 31) on the line. It is in the form of a single EBCDIC or hexadecimal character.

Note: The device addressing tables appear in the device component description manuals, (description and programmers guides), or for a CA, in the *VTAM Installation and Resource Definition* manual.

The GPOLL operand in the CLUSTER statement for a particular control unit specifies the EBCDIC character found in the table opposite the hardware address. When general polling, the general polling characters (X'7F') are appended to the address by NCP (or by VTAM when the BSC 3270 is on a CA BSC line).

The polling address sequence for general polling is:

```
GPOLL - GPOLL - X'7F' - X'7F'
```

The BSC 3270 control unit understands the X'7F's to mean, "examine the status of all attached devices."

GPOLL is specified a bit differently in an NCP source deck (major node) and a VBUILD TYPE=CA major node. For example, if a 3270 control unit was installed at physical address 1 on a BSC line, the GPOLL character from the table would be X'C1'. The definitions would then be:

```
GPOLL=C1C17F7F      in an NCP
and
GPOLL=C1             in VBUILD TYPE=CA
```

ADDR in a TERMINAL Macro

The devices attached to the control units have unique numeric addresses which are related to the port numbers of the control unit. They also are in the range of 0 to 31. These device numbers are also converted by table lookup to EBCDIC device address characters when defining the major node. For BSC 3270 control units, the device addresses normally begin with 0. They are selected by the planner/installer.

The definition for ADDR is different between an NCP and a VBUILD TYPE=CA major node.

First, the format of a specific poll address is as follows:

```
GPOLL      CU General Polling Address
GPOLL      (repeated)
X'dd'      Device Address-character
X'dd'      (repeated)
```

Second, for host to CU write- and control-type commands, the selection address is as follows:

```
X'cc'      CU Selection Address
X'cc'      (repeated)
X'dd'      Device Address-character
X'dd'      (repeated)
```


NCP BSC Addressing

The ADDR operand in an NCP TERMINAL definition refers to the selection address and is defined with four hexadecimal characters.

The specific poll address in an NCP is defined with an additional operand in the TERMINAL macro, the POLL operand. As with ADDR, POLL is defined with four hexadecimal characters.

CA BSC Addressing

The ADDR operand in a VBUILD TYPE=CA major node, on the other hand, refers to the specific poll address and is defined with a single hexadecimal character. VTAM creates the proper device selection address when necessary, therefore no additional TERMINAL operand is needed; that is, no POLL operand.

■ ■ ■

Please turn to Mini-Course 13, Exercise 13.1, in your PRG and answer the exercise questions.

GPOLL and ADDR Example

Refer to Figure 13-1 for the following discussion. For this example, assume that the BSC stations are on either an NCP BSC communication line or on a BSC communication adapter line. Both definitions will be shown.

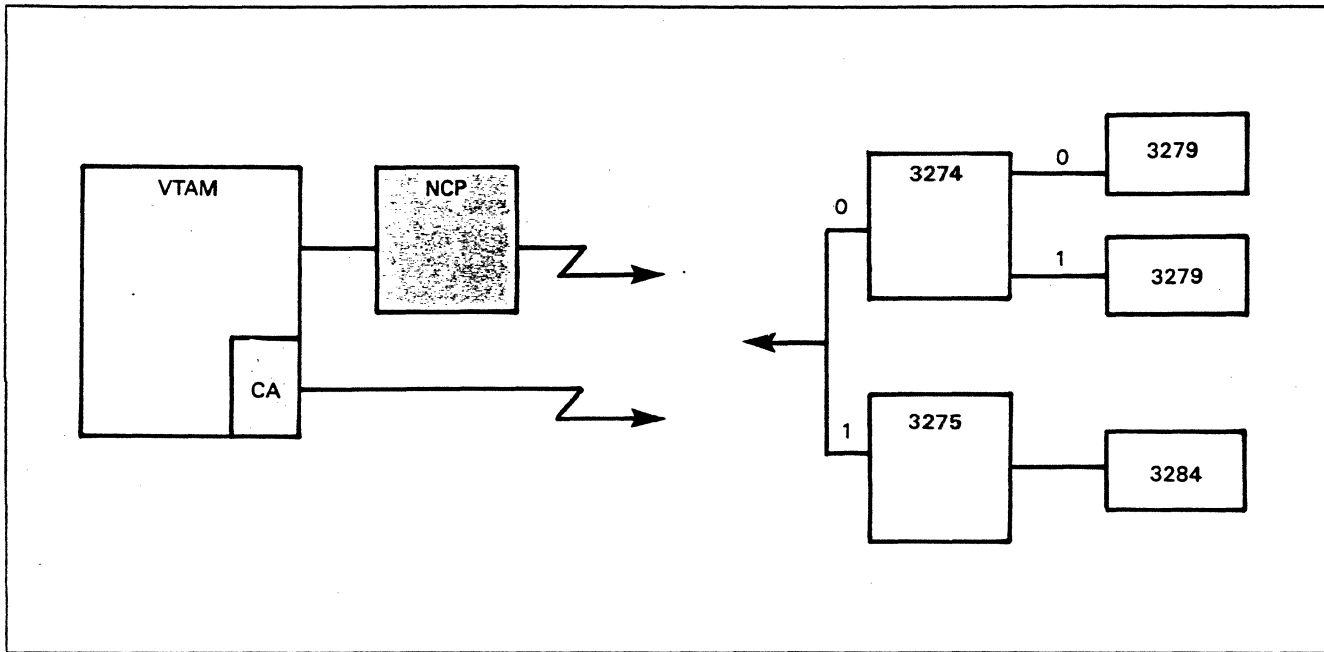


Figure 13-1. Example BSC Configuration

A device addressing table for the control unit character, the device address character, and for NCP, the selection address character, are shown in Figure 13-2.

CU Number or Device No. -----	GPOLL= NCP POLL= CA ADDR= -----	NCP-Only ADDR= -----
0	40	60
1	C1	61
2	C2	E2
3	C3	E3
4	C4	E4
5	C5	E5
6	C6	E6
7	C7	E7
8	C8	E8
9	C9	E9
10	4A	6A
11	4B	6B
12	4C	6C
13	4D	6D
14	4E	6E
15	4F	6F
16	50	F0
17	D1	F1
18	D2	F2
19	D3	F3
20	D4	F4
21	D5	F5
22	D6	F6
23	D7	F7
24	D8	F8
25	D9	F9
26	5A	7A
27	5B	7B
28	5C	7C
29	5D	7D
30	5E	7E
31	5F	7F

Figure 13-2. General Poll (GPOLL) and Device Addressing Table

A 3274 Control Unit is at numeric address 0 on the line, and a 3275 is at numeric address 1. The 3274 Control Unit has two 3279 Color Display Stations attached at its port addresses of 0 and 1. The 3275 has an attached 3284 Printer. The 3275 does not have a port address and the 3284 is the only device that may be attached to it. (The 3284 Printer is defined with the FEATUR2 operand in a TERMINAL macro as described above.)

These addresses may now be used to find the corresponding EBCDIC characters in the table (see Figure 13-2).

For the 3274 CU poll address character, GPOLL will be found opposite the CU number 0, its numeric address. The GPOLL character is 40.

The 3275 GPOLL address character is found in the same manner opposite its numeric address of 1. The GPOLL character is C1.

Figure 13-3 shows the CLUSTER macros in both an NCP and a VBUILD TYPE = CA major node for the 3274 and 3275 control units.

NCP Source -----	VBUILD TYPE=CA -----
●	●
CLUSTER CUTYPE=3271,....., GPOLL=40407F7F	CLUSTER CUTYPE=3271,....., GPOLL=40
TERMINAL,	TERMINAL,
●	●
CLUSTER CUTYPE=3275,....., GPOLL=C1C17F7F	CLUSTER CUTYPE=3275,....., GPOLL=C1
TERMINAL,	TERMINAL,

Figure 13-3. GPOLL in the CLUSTER Macro for BSC 3274 and 3275

Note that in the NCP definition, GPOLL is coded with the four-character General Poll address including the X'7F's.

NCP Specific Poll Address - POLL

As described earlier, ADDR has a different meaning between an NCP and a VBUILD TYPE = CA definition. Therefore, the table lookup for the TERMINAL macro uses two columns in the table to determine specific poll (POLL) and selection address (ADDR) values. Refer to Figure 13-2 for the following discussion.

The specific poll address is determined by adding the device address character (repeated) to the GPOLL character (repeated). For the example 3274, the 3279 addresses are determined from the same column as for the CU GPOLL character. Device addresses for the 3279s convert from ports 0 and 1 to address characters 40 and C1 respectively.

The 3275 has no ports, therefore, the device number is always 0 which converts to address character 40.

The coding for POLL in an NCP is shown in Figure 13-4.

NCP Selection Address - ADDR

The selection address (ADDR) in the TERMINAL macro for an NCP is determined by replacing the GPOLL characters for the control unit with the CU selection character. The character appears opposite the CU number in the selection character column of the table (see Figure 13-2).

For the 3274 the selection character is found in the third column opposite the CU number 0 as character 60.

The 3275 selection character, found in the same manner, is character 61 opposite the CU number 1.

Since the device address characters were found when POLL was determined, they are appended to the CU selection character to form the ADDR value. The NCP TERMINAL macro with the POLL values for the 3274 and the 3275 are shown in Figure 13-4.

NCP3274	CLUSTER	CUTYPE=3271,....	(for the 3274)
		GPOLL=40407F7F	(3274 General Poll)
		●	
NCP79CDO	TERMINAL	TERM=3277,....,	
		ADDR=60604040,	(3279 Selection Address)
		POLL=40404040	(3279 Specific Poll)
		●	
NCP79CD1	TERMINAL	TERM=3277,....,	
		ADDR=6060C1C1,	
		POLL=4040C1C1	
		●	
NCP3275	CLUSTER	CUTYPE=3275,....	(for the 3275)
		GPOLL=C1C17F7F	(3275 General Poll)
		●	
NCP32750	TERMINAL	TERM=3275,....,	
		FEATUR2=(PRINTR,..	(the 3284-3 Printer)
		ADDR=61614040	(3275 Selection Address)
		POLL=C1C14040	(3275 Specific Poll)
		●	

Figure 13-4. Example ADDR and POLL in NCP TERMINAL Macro

ADDR in CA TERMINAL Statement

As described previously, the ADDR operand in the TERMINAL statement for a BSC communications adapter (CA) device definition refers to the specific polling address for the device and not to the selection address as in an NCP. Since the GPOLL character has been determined (see Figure 13-3), the only further reference to the table is to find the device addresses for the ADDR operand.

The ADDR character is found in the same column of the table (see Figure 13-2) used for the CU number to find the GPOLL character.

The first 3279 will have a device address character (ADDR) of 40 (opposite its 0 port address) and the second 3279 will have a character of C1 (opposite its 1 port address).

In the same manner, ADDR is determined for the 3275. With the 3275, since there are no port addresses, 0 is assumed and the device address character is always specified as 40.

When used for specific polling, the control unit address character (GPOLL) and the device address character (ADDR) are sent as a combined address by VTAM to the control unit. It is sent as a 4 hexadecimal character address in the form GPOLL-GPOLL-ADDR-ADDR (as it is with an NCP). If the

control unit is to be general polled, VTAM substitutes X'7F's in the ADDR positions. Selection addressing is done by VTAM without user definition.

The TERMINAL statement for the example VBUILD TYPE = CA major node is shown in Figure 13-5.

CA3274	CLUSTER	CUTYPE=3271,..... GPOLL=40	(for the 3274) (3274 'General Poll')
		●	
CA79CD0	TERMINAL	TERM=3277,..... ADDR=40	(3279 'Specific Poll')
		●	
CA79CD1	TERMINAL	TERM=3277,..... ADDR=C1	
		●	
CA3275	CLUSTER	CUTYPE=3275,..... GPOLL=C1	(for the 3275) (3275 'General Poll')
		●	
CA32750	TERMINAL	TERM=3275,..... FEATUR2=(PRINTR,.. ADDR=40	(the 3284-3 Printer) (3275 'Specific Poll')
		●	

Figure 13-5. Example ADDR in a CA TERMINAL Macro

VTAM Procedural Options

Information chosen by the designer/coder for the remote non-SNA major node entries include:

- Resource names for the control units and terminals.
- Procedural options used by VTAM to communicate with the control unit and the terminals (refer to Mini-Course 7, *VTAM-Only Operands in Device Major Nodes*).
- The choice of grouping of the communications lines if there is to be more than one line.

Note: When defining communications adapter major nodes, more than one VBUILD TYPE = CA major node may be defined; therefore, as with SDLC lines, BSC lines may appear in a BSC-only major node.

In an NCP major node, all lines are, by definition, in the same major node. BSC lines are always defined before SDLC lines and have many of the same parameters as SDLC lines. BSC lines in an NCP are, unlike SDLC lines, nonshareable. In a multihost environment all CLUSTERS and TERMINALS on one BSC line may be owned by only one host at a time.

The VTAM procedural options appear on the GROUP, LINE, CLUSTER, and TERMINAL statements as has been discussed in prior mini-courses. Consult with the NCP installer as to VTAM requirements for the BSC lines. In the NCP BSC line definition, there are several additional parameters to consider compared to BSC lines in a communications adapter major node.

BSC in VBUILD TYPE = CA

GROUP and LINE statements for BSC lines appear in a CA major node when installing VTAM and a communications adapter.

The GROUP must contain only BSC lines. The only operands in the GROUP statement for BSC are ISTATUS and LNCTL = BSC.

In the LINE statement, the operands to consider are: ADDRESS, ISTATUS, RETRIES, and SERVLIM.

RETRIES must be specified with the LINE statement for BSC lines, and a value of 4 is usually sufficient.

SERVLIM, for BSC lines, specifies the maximum number of output WRITE operations to be performed before a polling operation is started. Consideration must be given to the terminal types on the line. For example, too many writes to a 3270 screen may cause overwrites before the terminal operator has an opportunity to respond, even though the writes are application program dependent.

■ ■ ■

Please turn to Mini-Course 13, Exercise 13.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 14

Switched Network Considerations

Mini-Course 14. Switched Network Considerations

Introduction

An important consideration for switched networks is that establishment of the data link is different from the nonswitched environment. Once the link is connected, session establishment proceeds as if it were a nonswitched connection.

Switched network communications is similar to ordinary telephone conversation in that the host or remote terminal dial a phone number for connection. Either the host or remote terminal may perform the dialing.

Special dialing equipment at the host, and also at the remote terminal, make use of ordinary telephone circuits (switched circuits) to establish the data link. There are various types of dialing equipment available for dial-out (host to terminal) or dial-in (terminal to host) operation. The type of equipment and how it is to be used is dependent on the user application requirements. This mini-course will describe the types of operation and how a switched network is defined to VTAM.

VTAM definition of lines, PUs, and LUs is handled differently for switched networks than other types of major node definitions. There is a major node definition for each end of the telephone circuit—one for the communications line(s) at the host, and one for the remote SNA PUs and LUs. The two major nodes are:

- A definition for the switched communications lines connected to dialing or answering equipment at the host site, either in an NCP major node or in a communications adapter (VBUILD TYPE = CA) major node.
- A switched network major node (VBUILD TYPE = SWNET) for SNA PUs and LUs connected to telephone equipment at the remote locations.

Type of Operation

User considerations for switched networks include the type of operation required for the applications. VTAM supports three types of operation of a switched network:

- Dial-in operation from terminal to host
- Manual dial-out operation from host to terminal
- Automatic dial-out operation from host to terminal.

These types of operation are explained below.

Dial-In Operation

The remote terminal operator calls a host site dial port telephone number. The dial port is equivalent to a communications line at the host and is associated with one of the communication line addresses defined in a major node. The communications controller (NCP) or a communications adapter (CA) informs VTAM of the incoming call. VTAM requests a physical unit ID, then locates the PU definition for the calling terminal in another major node—the switched network major node. VTAM activates the PU and its LUs and the connection from there on works like a nonswitched connection.

Manual Dial-Out Operation

When an application program requests a session with a remote terminal, VTAM looks at both the switched network major node and the NCP or CA major node. First, it searches the switched network major node for the terminal, using the minor node name supplied by the application program. When the minor node definition is found, it looks at an associated path definition to find the terminal telephone number and an NCP or CA major node GROUP name assigned to the path. The GROUP name is found in the NCP or CA major node and used to select an available dial communications line from the group. VTAM then sends a message to the domain operator with the telephone number and the NCP or CA line address selected.

The domain operator manually dials the number and NCP or the CA notifies VTAM when the connection is complete. At connection the remote PU sends a PU identification to the local PU. VTAM searches the switched network major node to verify the PU identification. The connection is then ready for session establishment.

Automatic Dial-Out Operation

When an application program requests a session, VTAM selects a dial port and finds the telephone number as it does with the manual dial-out operation. However, in automatic dial-out operation, VTAM passes the telephone number directly to the NCP or CA. The NCP or CA, using its auto-call unit interface (AUI) and an external automatic calling unit (ACU), dials the remote station number and notifies VTAM when the connection is complete.

Note: A 37xx communications controller may be configured with an integrated auto-call unit (ACU) function.

Switched Network Definition

Definition Statements

Figure 14-1 shows, in skeleton form, the definition statements and the corresponding operands to describe a switched network to VTAM. Operands which apply only to switched networks are shown; other operands which are common to other PU/LU definitions have been omitted for clarity.

STATEMENTS AND OPERANDS FOR DIAL-IN ONLY OPERATION

* NCP OR VBUILD TYPE=CA MAJOR NODE *

GROUP DIAL=YES
LNCTL=SDLC

LINE ANSWER
CALL=IN

PU MAXLU

* VTAM SWITCHED MAJOR NODE *

VBUILD TYPE=SWNET

PU IDBLK
IDNUM

LU

STATEMENTS AND OPERANDS FOR DIAL-OUT OPERATION

* NCP OR VBUILD TYPE=CA MAJOR NODE *

GROUP DIAL=YES
LNCTL=SDLC

LINE ANSWER
AUTO (not coded for manual dial-out)
CALL=OUT or INOUT
REDIAL (NCP only)

PU MAXLU

* VTAM SWITCHED MAJOR NODE *

VBUILD TYPE=SWNET
MAXGRP
MAXNO

PU IDBLK
IDNUM
MAXPATH

PATH DIALNO
GID
GRPNM
PID
REDIAL
USE

LU

Figure 14-1. VTAM Switched Network Definition

Defining Operation Type

The type of switched network operation is defined with the CALL operand of the LINE definition statement in the NCP or CA major node. The choices are:

- LINE ...,CALL=IN,... Dial-in only
- LINE ...,CALL=OUT,... Dial-out only
- LINE ...,CALL=INOUT,... Dial-in and Dial-out

Automatic dial-out operation requires automatic calling unit (ACU) equipment installed for each switched line.

Note: For automatic dial-out operation using a communications adapter (CA), a maximum of two lines may have an automatic calling unit (ACU) installed via an auto-call unit interface (AUI) feature.

If an ACU is to be used, it is defined with the AUTO operand on the LINE definition statement. The value of AUTO is the address of the line on which it is installed. Note that for automatic dial-out, it would be either CALL=OUT or CALL=INOUT.

Switched Line Definition

Switched lines are defined in an NCP or VBUILD TYPE=CA major node with GROUP, LINE, and PU macros.

GROUP

The NCP or CA major node GROUP statement must specify DIAL=YES and LNCTL=SDLC. One or more GROUP statements may be defined. These GROUP names are referenced in the switched network (VBUILD TYPE=SWNET) major node PATH statements.

LINE

The LINE macro describes the communications line in the NCP or CA major node as if it were a nonswitched line, with the addition of the ANSWER, AUTO, and CALL (plus REDIAL in NCP) operands to specify how the line is to be used as a switched line.

ANSWER specifies whether the line will accept a terminal dial-in. If ANSWER=OFF is specified, no calls will be accepted on the line. However, the network operator may use the VARY NET, ANS command to change the status to ANSWER=ON which will allow incoming calls on the line.

CALL and AUTO are used as described in *Defining Operation Type* above.

In NCP only, REDIAL may be specified to have the NCP repeat the dialing sequence in a dial-out operation. The number of retries in one dial sequence is specified in VTAM's SWNET major node PATH statement (also with a REDIAL operand). The NCP redialing allows pauses between retries, additional sequences, and pauses between sequences. The NCP REDIAL operand format is:

REDIAL (,t1,n,t2)

where:

, = skip 1st value (defined in VTAM PATH REDIAL)
t1 = pause in seconds between dials (retries)
n = number of additional sequences to try
t2 = pause in seconds between sequences

An example NCP REDIAL sequence is shown in Figure 14-2.

Example:

```
LINE ...,REDIAL=(,3,2,10) in NCP
PATH ...,REDIAL=2,... in VTAM
                          SWNET Major Node
```

The NCP attempts to dial would be:

Number of dials in a sequence = ONE + 2 REDIALs = 3
Number of sequences = ONE + 2 additional (n) = 3

```
Seq. 1 - dial(1) pause(3) dial(2) pause(3) dial(3) pause(10)
Seq. 2 - dial(4) pause(3) dial(5) pause(3) dial(6) pause(10)
Seq. 3 - dial(7) pause(3) dial(8) pause(3) dial(9)
```

Figure 14-2. Example REDIAL Sequence in NCP

PU

The MAXLU operand in the PU statement of an NCP or CA major node is used to specify a maximum number of LUs for any single PU to use over the link. In NCP, the value could be made greater than the actual number of LUs defined so that the PU can handle dynamically added LUs.

In NCP, PUTYPE=(1,2) may be specified if both physical unit types 1 and 2 will be in the switched network.

VBUILD TYPE = SWNET

The PU, PATH, and LU definitions for the remote stations are placed in an VTAM switched network major node. Values are defined with operands used only in switched network definitions along with operands used in other major nodes. Refer to Figure 14-1 on page 14-4 which shows the SWNET major node with the definition statements and the operands for dial-in or dial-out networks. Only the operands that apply to switched networks are shown. Further consideration for the operands will be covered later in this mini-course.

Path Definition

PATH definition statements are coded in the switched network major node for each PU which is to have dial-out capability.

Note: The switched PATH statement should not be confused with networking PATH statements which define routing in the network.

Switched PATH statements include:

- The telephone number assigned to the PU.
- The name of a major node GROUP which contains switched SNA/SDLC lines.
- The initial status of the path.
- The number of times to retry dialing.
- Two identifiers for the network operator's use.

More than one PATH statement may be included with each PU/LU definition since a PU may have more than one telephone number. For example, a PU may have a long distance number and a tie-line or a Wide Area Telephone Service (WATS) number.

Also, the NCP or CA major node may have more than one switched SDLC GROUP, in which case, multiple PATH statements could be used for the same PU even if it had only one telephone number.

If more than one PATH statement is included for a PU, VTAM will try the first defined path. If no line in the group is available, it will try the next path until an available line is found or until no more paths exist.

■ ■ ■

Please turn to Mini-Course 14, Exercise 14.1, in your PRG and answer the exercise questions.

VBUILD TYPE = SWNET Notes

VBUILD Notes

MAXGRP

The MAXGRP operand of VBUILD TYPE = SWNET specifies the total number of GROUP names that will be pointed to with path GRPNM operands. For example, if one NCP or one CA major node GROUP contains all of the switched line LINE definitions, MAXGRP = 1 regardless of the number of times it is referenced in path GRPNMs.

MAXNO

The MAXNO operand specifies how many unique telephone numbers will be specified in the SWNET major node. For example, if there are 3 PU (switched) definitions, each followed by two PATH statements, and each with a unique DIALNO, then MAXNO = 6 since there are six unique telephone numbers regardless of the number of PUs.

VBUILD TYPE = SWNET - PU Notes

ADDR

The SDLC station address is a one-byte address selected when the station (PU/LU) is installed. This address is required in the PU definition as ADDR = address. The character must be obtained from the equipment installer.

IDBLK

The IDBLK number is a 3-hexadecimal-character (12-bit) device dependent number. It may be found in the component description manual for the control unit. For example, the IDBLK for a 3274 control unit would be coded: IDBLK = 017.

IDNUM

The third PU identification, IDNUM, is a 5-byte (20-bit) Terminal ID chosen at device installation time. As with ADDR, the IDNUM must be obtained from the equipment installer.

MAXPATH

The MAXPATH PU operand specifies the number of PATH statements which follow this PU definition only. It is not a count of all PATH statements in the major node. Note that MAXPATH and PATH statements themselves only apply to dial-out (host to terminal) connections.

DISCNT

The DISCNT operand works differently for switched connections than for nonswitched. It specifies what VTAM is to do when the last LU session for this PU terminates. With a switched connection, VTAM does not deactivate the physical unit when DISCNT= YES is specified. VTAM simply hangs-up the phone line. A new request for a session with the PU/LU causes the session to be re-established after the dialing connection procedure. In this way, telephone line use costs are minimized.

VBUILD TYPE= SWNET - PATH Notes

DIALNO

The DIALNO specifies the terminal telephone number in exactly the same format as the number on a standard telephone. For example, some localities must dial 1 before the area code. In this case, the 1 is coded in front of the number. Some types of equipment allow a dialing pause to be coded.

GRPNM

As described earlier in the section *Manual Dial-out Operation*, GRPNM points to an NCP or CA major node GROUP name for selection of a SDLC switched line. If an available line cannot be found in this GROUP, VTAM will try the next PATH statement, if one is available.

USE

PID

GID

The USE operand is similar in operation to the ISTATUS operand used with other network components. It specifies the initial usability of the path. It may be changed by the network operator.

The PID identifier is unique to the PU for which the PATH is coded. That is, each PATH statement for a single PU requires a different PID. For example, the operator may enter a VARY command to change the usability of a particular path from a defined USE=NO as follows:

```
VARY NET,PATH=USE,ID=PUname,PID=identifier
```

The GID is an identifier for multiple PATHs; in the same PU or in different PUs. Its use is the same as for the PID; that is, for operator control. For example, if all direct dial long distance PATHs are given the same GID (including different PUs), the operator could make all of them usable by entering:

```
VARY NET,PATH=USE,GID=identifier
```

An Example Configuration for Switched Networks

The following are assumptions for the example configuration used in previous mini-courses. We will use these requirements to consider the switched network definition.

1. The INQUIRE program has been updated to give it the capability to acquire two IBM 3767 Communication Terminals to produce summary reports of the day's activities at two remote locations. Since it is a batch type operation at the end of each day, the terminals need not be active throughout the entire VTAM execution. For this reason, and others to follow, a switched network environment is chosen. Each terminal will have a single telephone number, and the network operator is not to be involved with manual dialing when the reports are to be produced. Direct dial numbers assigned for the two 3767s are 202-833-7409 and 202-833-7576 at station addresses C5 and C6 respectively.
2. An additional requirement is for a display station attached to a 3274 Control Unit to allow the end user to log on to either the INQUIRE or another program. There is also to be a 3287 printer which may be used to receive detail reports from the host location. Due to the expected low amount of activity, these terminals are also to be installed in a switched network.

The primary operation is for the end user to dial-in to a switched line connection at the host site (either an NCP or a communications adapter line in this example). After the connection is complete, the end users will key either INQ or a different USS command.

A secondary requirement is for the INQUIRE program to be able to call the 3274 (when there is no current session) to produce a detail report on the printer. In addition, the dialing process may be done either automatically or by the VTAM network operator.

The remote 3274 is to have two telephone numbers assigned to its auto-answer equipment: (1) a long distance direct dial number, 703-810-5675, and (2) a tie-line number, 8-444-5675. The station address chosen is C4.

3. Two switched communications lines are to be installed. The lines are to have addresses 033 and 035 (NCP or CA). Line 033 is to have an automatic calling unit (ACU) function.

Switched Network Definition Considerations

Now that the VTAM example requirements are known, there are considerations for the switched network definition.

The GROUPS and LINES

The two switched lines could be in the same GROUP or in separate GROUPs (in either an NCP or CA major node).

In the requirement for the 3767s, the operator is not to be involved with manual dialing. Since there is only one ACU, the definition must be made so that the outgoing call to the 3767s is made over line 033 (the ACU line). Recall that the PATH statement references a group name to select the lines to be used. Therefore, in this case, each line will be defined in a separate group. The PATH statement for the 3767s will both point to the group name SW01GRP1 for line 033 which has the automatic calling equipment. The GROUP, LINE, and PU definitions are shown in Figure 14-3 for either an NCP or a VBUILD TYPE = CA major nodes. (Other operands common to other GROUP and LINE definitions are not included in this example.)

* NCP or VBUILD TYPE=CA				
SW01GRP1	GROUP	ISTATUS=ACTIVE, LNCTL=SDLC, DIAL=YES		
SW01LN33	LINE	ADDRESS=033,		x
		ANSWER=ON,	default	x
		AUTO=033,		x
		CALL=INOUT,		x
		PAUSE=0.3,		x
		RETRIES=4		
	PU	MAXLU=3		
		PUTYPE=(1,2)	(NCP-Only)	
SW01GRP2	GROUP	DIAL=YES, LNCTL=SDLC		
SW01LN35	LINE	ADDRESS=035,		x
		ANSWER=ON,		x
		CALL=INOUT,		x
		PAUSE=0.2,		x
		RETRIES=4		
	PU	MAXLU=3		
		PUTYPE=(1,2)	(NCP-Only)	

Figure 14-3. NCP or CA Major Node Definition for Switched Line Groups

The Switched Network Major Node

Several pieces of information need to be determined before the switched major node can be coded. The information includes:

- The telephone numbers for the remote stations and for the host location (NCP or CA) dial ports.
- The IDBLKs and IDNUMs for the various PUs. These numbers may be obtained from the component description manuals for the devices or from the person(s) who planned the device installation.
- The PU type and the number of LUs.
- The PU addresses and the LU local addresses.
- The data handling characteristics of the devices to define MAXDATA, MAXOUT, etc.

Once the information is collected, the nodes may be coded. Choices that the designer/coder would make include:

- The node names
- The PATHs and the corresponding group references
- The LU logmode table entries
- The proper USSTAB for each LU
- The group and path identifications for the domain operator's use.

The switched major node for the example configuration is shown in Figure 14-4. (Some fictitious numbers were included for the information gathering steps mentioned above.) Note that the PATH statements follow the system requirements for telephone numbers and group identifications.

SWITCH01	VBUILD	TYPE=SWNET,MAXGRP=2,MAXNO=4	
PU043274	PU	ADDR=C4,	x
		IDBLK=017,	x
		IDNUM=19834,	x
		DISCNT=YES,	x
		MAXDATA=265,	default for PU_T2 x
		MAXOUT=7,	default x
		MAXPATH=4,	x
		PUTYPE=2	default
SW01PATH	PATH	DIALNO=988105675	x
		GID=5,	x
		GRPNM=SW01GRP1,	x
		PID=1,	x
		REDIAL=3	default x
		USE=YES	default
SW02PATH	PATH	DIALNO=97038105675,	x
		GID=6,	x
		GRPNM=SW01GRP1,	x
		PID=2	
SW03PATH	PATH	DIALNO=988105675,	x
		GID=5,	x
		GRPNM=SW01GRP2,	x
		PID=3	
SW04PATH	PATH	DIALNO=97038105675,	x
		GID=6,	x
		GRPNM=SW01GRP2,	x
		PID=4	
LU043279	LU	LOCADDR=2,	x
		MODETAB=MODE3270,	x
		DLOGMOD=M3279,	x
		USSTAB=USS3270,	x
		SSCPFM=USSSCS	
LU053287	LU	LOCADDR=3,	x
		MODETAB=MODE3270,	x
		DLOGMOD=M3287	
PU053767	PU	ADDR=C5,	x
		DISCNT=YES,	x
		IDBLK=007,	x
		IDNUM=19835,	x
		MAXDATA=261,	default for PU_T1 x
		MAXPATH=1,	x
		PUTYPE=1	
SW05PATH	PATH	DIALNO=92028337409,	x
		GID=5,	x
		GRPNM=SW01GRP1,	x
		PID=5	

Figure 14-4 (Part 1 of 2). VTAM Switched Major Node Example

LU053767	LU	LOCADDR=0,	X
		MODETAB=MODE3767,	X
		SSCPFM=USSSCS,	X
		USSTAB=USS3767	
PU063767	PU	ADDR=C6,	X
		DISCNT=YES,	X
		IDBLK=007,	X
		IDNUM=19836,	X
		MAXPATH=1,	X
		PUTYPE=1	
SW06PATH	PATH	DIALNO=92028337576,	X
		GID=5,	X
		GRPNM=SW01GRP1,	X
		PID=6	
			X
LU063767	LU	LOCADDR=0,	X
		MODETAB=MODE3767,	X
		SSCPFM=USSSCS,	X
		USSTAB=USS3767	

Figure 14-4 (Part 2 of 2). VTAM Switched Major Node Example



Please turn to Mini-Course 14, Exercise 14.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 15

VTAM Class of Service and Routing

Mini-Course 15. VTAM Class of Service and Routing

Introduction

VTAM provides the capability to define communication networks of any size, from single-domains with perhaps a single NCP communications controller, to large networks which include multiple hosts and multiple interconnected NCPs. Regardless of the size of the network, VTAM can be defined to fit the device configuration and the procedure designed to handle session traffic in that configuration.

This mini-course describes the elements of interconnection between VTAM and NCPs in a single network. In addition, it introduces the coding requirements to define VTAM tables that control the traffic flow through all of the network nodes and interconnections.

Parallel Links

Multiple Synchronous Data Link Control (SDLC) links can be used to connect adjacent NCP communications controllers as shown in Figure 15-1.

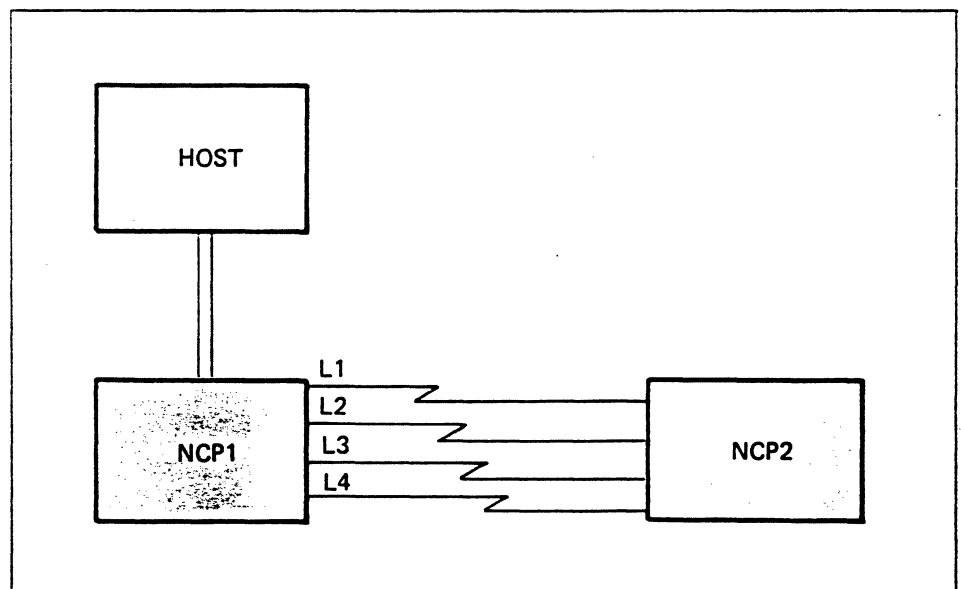


Figure 15-1. Parallel Links

Any or all of the links can be active and carrying traffic simultaneously. For example, the four parallel links in Figure 15-1 can be defined as one logical link. Path information units (PIUs) flowing on a session between the host and NCP2 will use all four links between NCP1 and NCP2. Each parallel link is controlled (activated/inactivated) independently of the

others. The maximum number of parallel links between communication controllers is limited only by the hardware.

There are two significant advantages of parallel links:

- There will be increased availability when there is an individual link failure; that is, sessions will not be disrupted.
- There will be a higher effective bits-per-second transmission rate (increased session traffic bandwidth) between two NCPs.

Transmission Groups

Management of parallel links depends on the concept of transmission groups (TGs). This concept allows multiple SDLC links to be treated as one logical link between adjacent NCP subareas.

A transmission group consists of one or more SDLC links between two NCPs. There may be more than one transmission group between the same two NCPs. For example, Figure 15-2 shows three transmission groups.

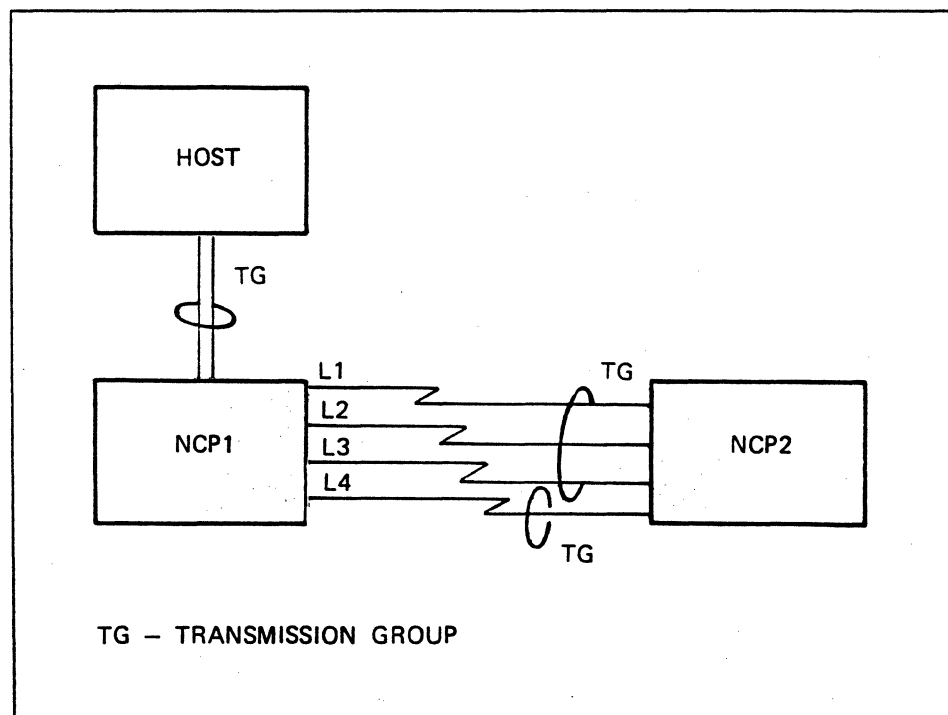


Figure 15-2. Transmission Groups

Links L1, L2, and L3 make up one transmission group, and link L4 makes up a second transmission group. The transmission group between the host and NCP1 in Figure 15-2 is the host channel. Although host channels can never be parallel (or grouped), they are treated as a transmission group for consistency in definition. Specific links are assigned to specific NCP to NCP transmission groups in the NCP definitions.

During a session, data flows between adjacent NCPs across a specific transmission group and not across a specific link in that transmission group. The transmitting NCP dynamically determines which specific link within that transmission group is to be used for transmission of a particular PIU.

For example, assume that traffic for a particular LU-LU session is assigned to a route using the transmission group of links L1, L2, and L3 in Figure 15-2. NCP1 will transmit PIUs for this session over any of the three links depending on which of the three is available at the time. The three links increase the relative communication bandwidth and decrease the impact of a link failure. Session traffic continues even if two of the links become inoperative. Only if all three links fail will the LU-LU session be broken.

If all links in a transmission group fail, the session is not automatically switched to an alternate transmission group. However, if an alternate transmission group is available and is defined as an alternate, the session may be restarted using the alternate transmission group. This is an important advantage of the parallel link transmission group concept.

Transmission groups are assigned a transmission group number (TGN) in the range of 1 to 255 in the NCP definition. For example, link L4 in Figure 15-2 could be assigned TG number 1 (TG1) and links L1, L2, and L3 could be assigned TG number 255 (TG255). The maximum number of transmission groups allowed between adjacent NCPs is eight. That is, regardless of the number of links between the NCPs, they must be placed in no more than eight groups.

The host channel, a special transmission group, is always referred to as transmission group number 1 (TG1).

Explicit Routes (ERs)

Explicit routes (ERs) provide the mechanism to define physical paths between a host subarea and NCP subareas. Explicit routes are ordered sets of network nodes (host and NCP) and transmission groups (TGs) for traffic between two subareas. An ER must be defined between two subareas (including any/all intermediate subareas) before session traffic can flow between the two subareas.

Explicit routes are unidirectional; that is, the ER is defined from an origin subarea to a destination subarea. They must exist in pairs, one ER from the origin to the destination and another ER for the return communication. An ER may be assigned a number from 0 to 7. The pair of ERs do not have to be assigned the same ER number, but they must traverse the same network nodes and the same transmission groups.

VTAM and NCPs exchange ER information at network activation to inform each other of which ERs they have defined. Corresponding ERs are marked operative, that is, the ER numbers are available for session traffic. The ER status changes from operative to active when a session is established over the ER.

Virtual Routes (VRs)

A virtual route (VR) is a bidirectional logical connection between a host subarea and an NCP subarea (or another host subarea in a multisystem environment). A virtual route must be defined between two subareas in order for session traffic to flow between the two subareas. Like explicit routes, they are numbered from 0 to 7.

A VR is defined between the host subarea and the NCP subarea by associating the VR with a particular ER. Virtual routes are defined (mapped to ERs) with VTAM PATH definition sets. Although the VTAM PATH statement associates a virtual route with only one ER, the virtual route is actually associated with, or mapped to a pair of explicit routes. Figure 15-3 illustrates the relationship between a virtual route and a pair of explicit routes.

Note: VRs are not defined in NCPs in a single network. However, in SNA interconnected networks (SNI), VRs are defined in the PATH definitions of gateway NCPs.

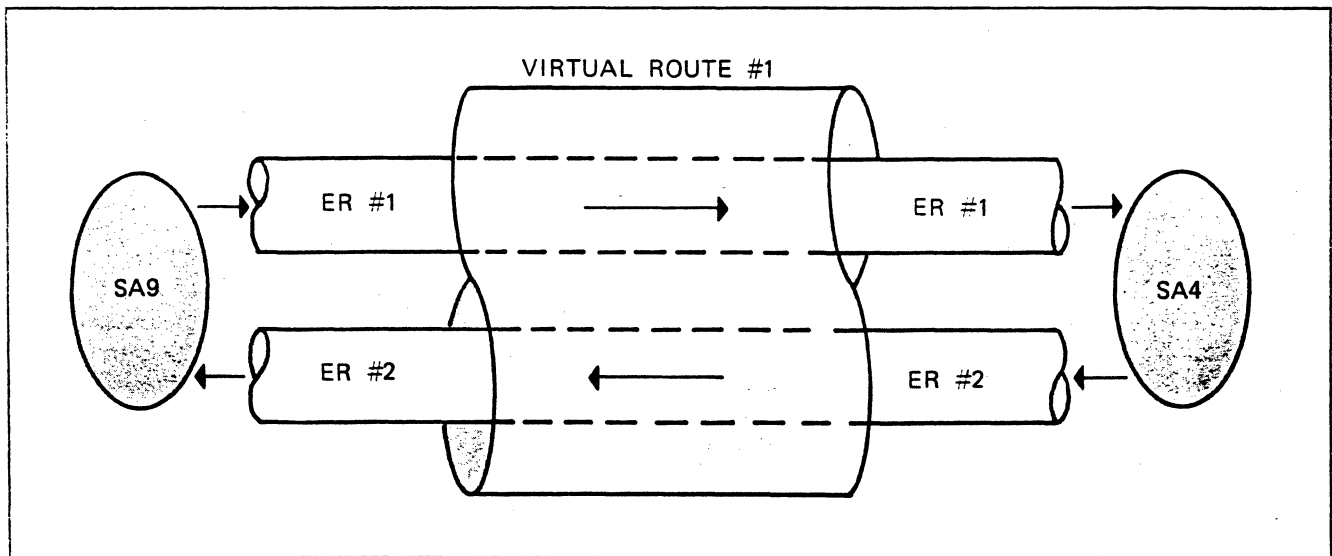


Figure 15-3. Relationship Between Virtual and Explicit Routes

The figure shows ER1 between origin subarea 9 and destination subarea 4, and ER2 between origin subarea 4 and destination subarea 9. There may be a number of transmission groups and intermediate nodes along the path of the virtual route, but it is always associated with the pair of explicit routes ER1 and ER2.

Like explicit routes, eight virtual route numbers (VR0 to VR7) can be associated with ERs between two subareas. A single virtual route number may be associated with a single explicit route number (VR1 to ER1, for example), or multiple virtual route numbers may be associated with a single explicit route number (VR0, VR1, and VR3 to ER5, for example). In any case, only eight associations (0 to 7) can be made between virtual routes and explicit routes between any two subareas.

■ ■ ■

Please turn to Mini-Course 15, Exercise 15.1, in your PRG and answer the exercise questions.

Class Of Service and Transmission Priority

The characteristics of explicit routes between subareas may vary considerably. For example, one ER may map to a transmission group which has multiple links while another ER may map to a transmission group with only one link. The implication is that traffic on one ER would receive service with different physical characteristics than the other ER. VTAM provides the capability to match types of sessions (for example, interactive, batch, and security type sessions) to routes with specific characteristics. This capability provides flexibility when assigning routes for VTAM sessions, including the SSCP-PU, SSCP-LU, and the LU-LU sessions. The particular service chosen for a session is called the class of service (COS).

The choice of a class of service (COS) consists of more than the physical characteristics of the route since a transmission priority can be assigned to the route. Traffic in a session with a high transmission priority is given preference over traffic in sessions with lower transmission priorities.

There are three levels of transmission priority: 0, 1, and 2. Zero is the lowest priority and 2 is the highest.

Transmission priorities are implemented by using the queueing facility associated with transmission groups in NCPs. NCP forms a transmit queue for a transmission group (TG) when every link in the TG is busy. Within a transmission group queue, path information units (PIUs) are arranged in transmission priority sequence for subsequent transmission. Session traffic assigned transmission priority 2 (highest user priority) is placed at the head of the queue, while transmission priority 0 traffic is placed at the end of the queue. Within a specific transmission priority, all PIUs are enqueued FIFO (first-in-first-out). Figure 15-4 illustrates queueing of PIUs (messages) on two transmission group queues.

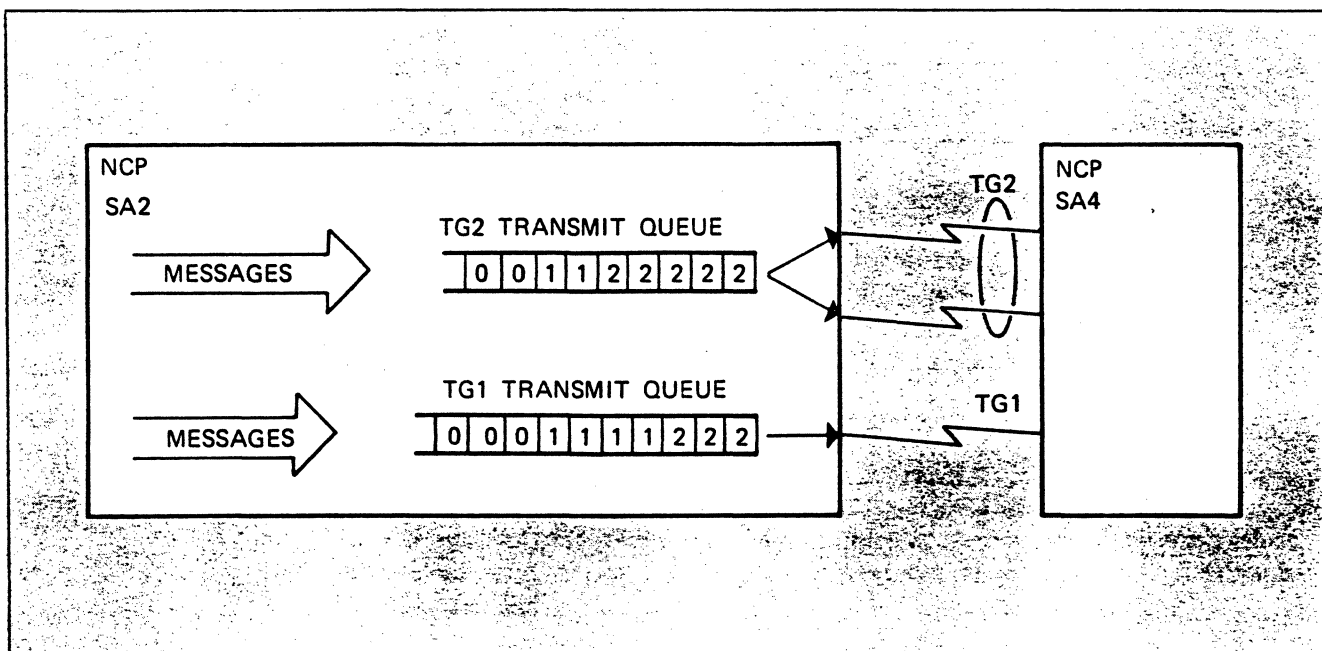


Figure 15-4. Transmission Group Message Queue

The two transmission groups connecting NCP SA2 and NCP SA4 are TG1 and TG2. Messages arriving at SA2 to be transmitted to SA4 are queued in transmission priority sequence on the appropriate transmission group queue. The numbers shown in the queues represent the transmission priority of a PIU.

After each 256 PIUs of any priority, the transmission group queue is flushed to ensure that low priority PIUs don't sit there forever.

Class Of Service Relationships

Class of service (COS) selection is the mechanism for assigning a virtual route number (VR#) and a transmission priority number (TP#) to a session. The virtual route number is used to select a type of route—fast, short, or high quality route—by mapping to an explicit route number (ER#). The paired transmission priority number for the virtual route is used to establish priority for traffic at that class of service level.

Classes of service are defined in a COS table. A class of service consists of a list of virtual route specifications and each specification includes a VR#-TP# pair. The first number identifies the virtual route number and the second number specifies the transmission priority to be assigned to the session. VTAM uses the first VR#-TP# pair in the class of service list to select an ER# for the session. The specified VR# is matched against an appropriate entry—to the session destination subarea—in an VTAM PATH definition. The PATH definition associates the particular VR# to an ER#.

Each virtual route number—transmission priority number pair (VR#-TP#)—in the class of service is called a *class of service identifier*.

Once the selected ER number is found, VTAM determines if the explicit route is operative. If operative, it will be used; if not, the next VR#-TP# in the COS entry list will be used to find an ER in the PATH definitions. If none of the explicit routes selected via the COS table are operative, the session will not be established.

Class Of Service Selection

The class of service for a session is related to the secondary LU in the session. VTAM selects the COS table entry by first searching the logon mode table entry associated with the secondary LU. If the logmode entry for the LU points to a COS entry, that entry in the COS is used for session setup. Figure 15-5 shows the relationship between a logon mode table entry, the COS table, and the VTAM PATH definition set.

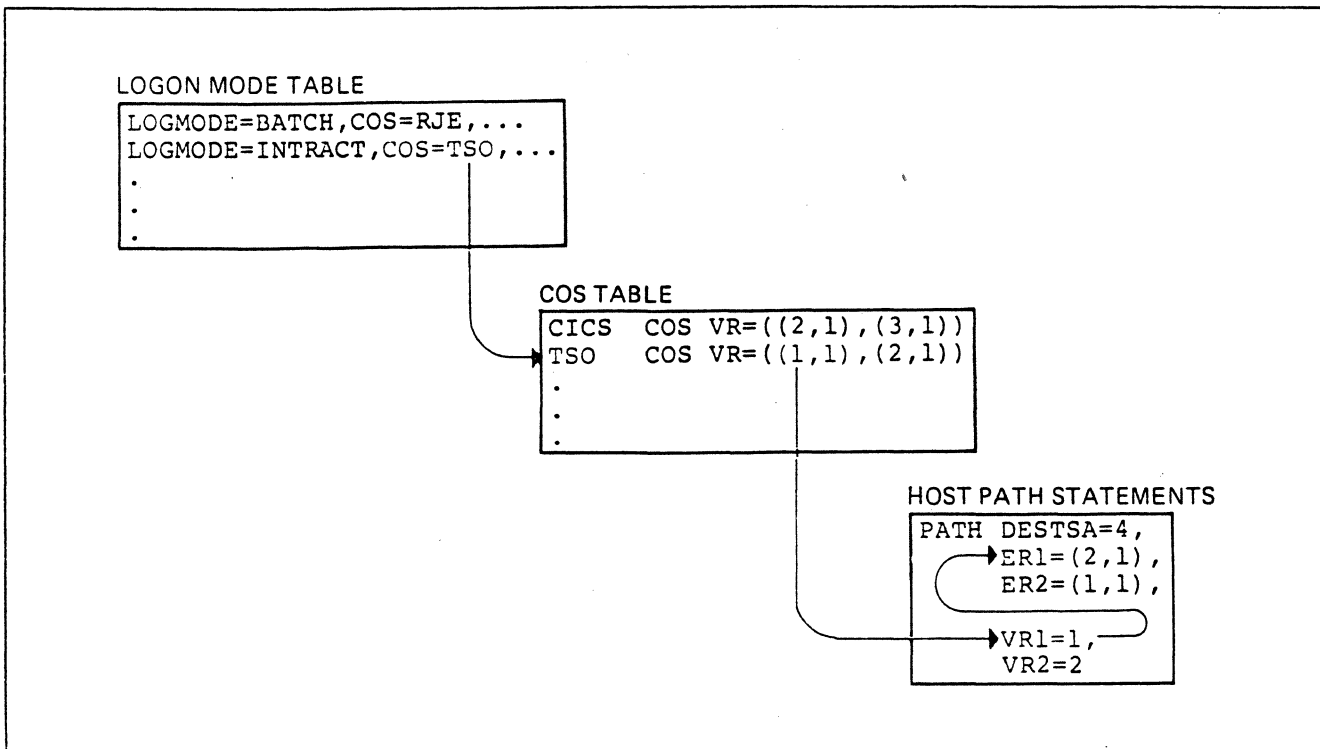


Figure 15-5. Class Of Service (COS) Selection

VTAM uses the following order of selection to find the proper explicit route for the pending LU-LU session:

1. Find the logmode entry in the logon mode table associated with the secondary LU.
2. Get the COS table entry pointer from the logmode entry.
3. Find the COS entry in the COS table.
4. Take the first VR#-TP# pair in the COS entry and search the destination subarea (DESTSA) entry in the PATH definition set.
5. Match the VR# from the COS entry with the VR# in the PATH definitions.
6. Associate the VR# with the proper ER# in the PATH entry.
7. Determine if that ER# is operative (or already active) and, if so, use it.
8. If the ER# is inoperative, go back to Step 4, get the next VR#-TP# pair, and continue the search.
9. If none of the ERs selected are operative or already active, fail session setup.

If there is no user-coded COS entry pointer in the logmode entry (Step 2 in the selection sequence), VTAM will use one of two default possibilities to select a class of service.

- Search the COS table for an unnamed COS entry to use.
- If there is no unnamed entry in the COS table, use the internal COS defaults.

COS Defaults

IBM-specified class of service defaults consist of all 24 VR#-TP# pairs, in the following sequence:

```
VR0,TP0; VR1,TP0; VR2,TP0; ... VR7,TP0;  
VR0,TP1; VR1,TP1; VR2,TP1; ... VR7,TP1;  
VR0,TP2; VR1,TP2; VR2,TP2; ... VR7,TP2;
```

VTAM uses the default sequence in the same manner as user-coded entries in a COS table; that is, an attempt is made with pair VR0-TP0 first. If VR0 maps to an inoperative ER, then VR1-TP0 is tried, and so on.

COS Hierarchy

The class of service identifier (VR#-TP#) can be selected from the COS table or from the IBM-specified defaults for both LU-LU sessions and SSCP sessions. The level of service for SSCP sessions (such as SSCP-PU and SSCP-LU sessions) may be placed in a COS table entry named ISTVTCOS. VTAM will look for the ISTVTCOS entry any time there is a SSCP session request to another subarea.

Figure 15-6 shows the order of search for COS values for SSCP and LU-LU sessions.

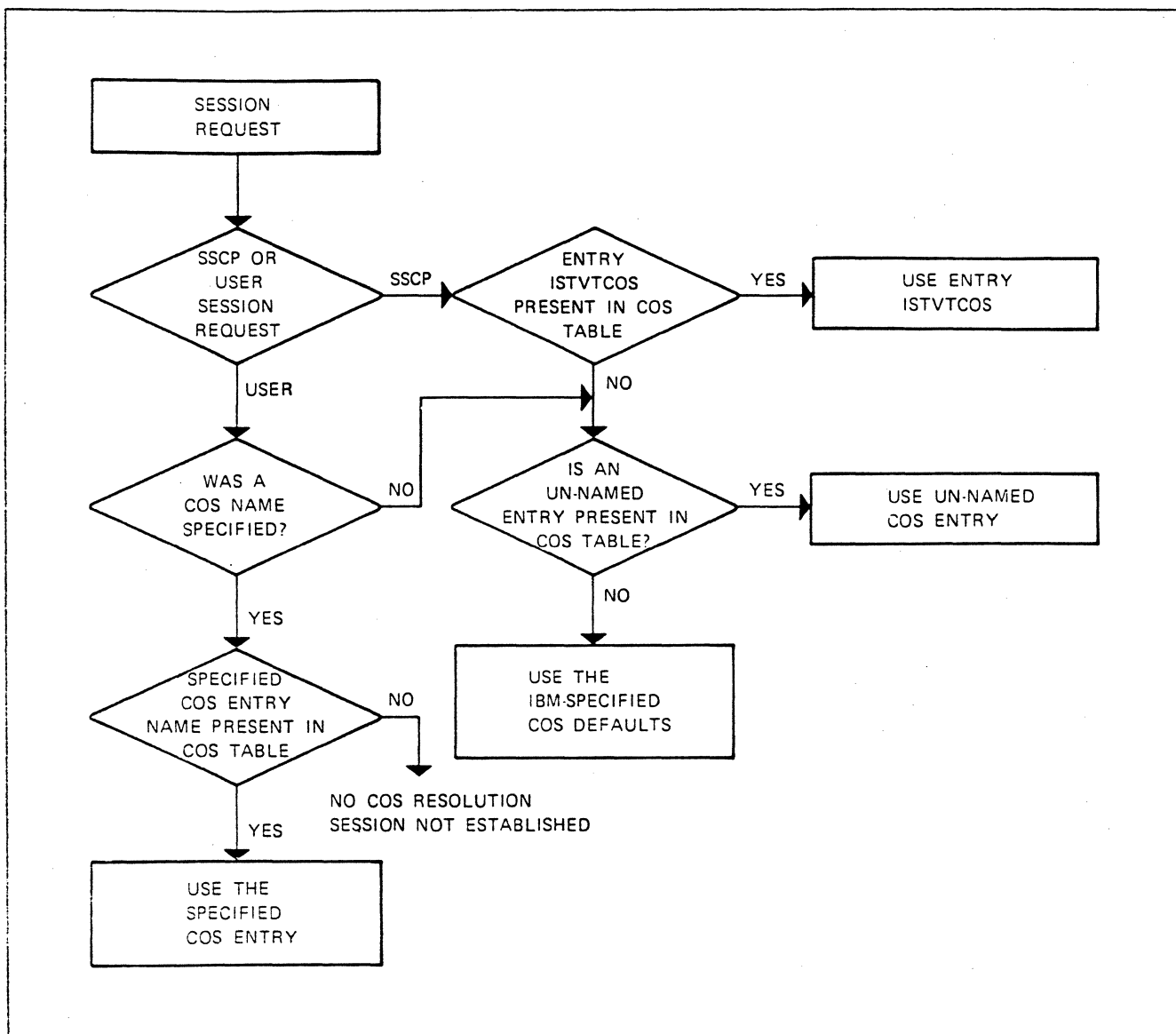


Figure 15-6. Class Of Service (COS) Hierarchy

For an LU-LU session request that specifies a COS name, that named entry in the COS table is used for COS resolution. The session request fails if the COS name is not found in the COS table.

If the session request does not specify a COS name, the unnamed entry in the COS table is used. Then, if there is no unnamed entry in the COS table, the IBM-specified class of service default is used. If class of service resolution is not possible because there is not an available explicit route, the LU-LU session request fails and the network operator is notified so that remedial action can be taken.

The procedure for an SSCP session request is similar to that for an LU-LU session. However, the ISTVTCOS entry in the COS table is used for class of service (COS) resolution. If there is no ISTVTCOS entry in the COS table, the unnamed entry in the COS table is used for COS resolution. Then, if no unnamed entry is found in the COS table, an IBM-specified

default is used for COS resolution. If there is no available route, the SSCP session request is queued for route availability and the network operator is notified.

COS Table Coding

A COS table is not supplied with a VTAM system. VTAM will use the hard-coded COS defaults if a user-coded table is not found. When there is a need to override the defaults (transmission priority, for example), the user may code, assemble, and link-edit a COS table, using the provided COS macros.

A user COS table is link-edited to the VTAM module library. In VSE, the COS table is link-edited into the VTAM module library; in MVS, the table is link-edited into VTAMLIB, and in VM the table is link-edited into VTAMUSER LOADLIB. The name of the link-edited module must be ISTSDCOS, and it must be link-edited as an ONLY LOADABLE data module.

The COS Table Macros

Class of service (COS) table coding uses three assembler macros: COSTAB, COS, and COSEND.

COSTAB Macro

The COSTAB macro defines the beginning of the COS table. The COSTAB macro must be the first macro in the source deck, and is coded as follows:

```
[csectname] COSTAB
```

The csectname specifies a 1 to 8-character name. The name is optional and has no VTAM significance if coded. Since the link-edited load module name must be ISTSDCOS, it is recommended that ISTSDCOS be used for the csectname.

COS Macro

One COS macro must be coded for each entry to be included in the COS table. The macro is coded in one of two formats as shown below:

```
[name] COS VR=(vr#,tp#) or  
[name] COS VR=((vr#,tp#),(vr#,tp#),(...))
```

In the COS macro, the name identifies the name by which this class of service entry is known in the VTAM system. If the name is omitted, it becomes the unnamed class of service (COS) entry. Recall that this unnamed COS entry is used in the event that a logon does not specify a COS name. Only one unnamed entry may be included.

The VR# and TP#, shown in the example above represent ordered pairs. Each pair consists of one virtual route number and one transmission priority number. The VR# parameters can be any virtual route number from 0 through 7; the TP# parameters can be any transmission priority from 0 (low priority) to 2 (high priority).

The sequence of the ordered pairs defines the preference order for route selection; that is, VTAM attempts to select the first specified virtual route number for the requested session. If the indicated virtual route number does not go to the required destination subarea number, or if the specified virtual route cannot be activated, VTAM goes to the second ordered pair and attempts to use that defined virtual route number. This operation continues until VTAM either finds a virtual route number that can be used, or exhausts all the entries in the selected COS entry, in which case the session request fails (LU-LU session request).

COSEND

The last macro of the COS table must be the COSEND macro. There are no parameters. The COSEND macro is followed immediately by the assembler END statement.

■ ■ ■

Please turn to Mini-Course 15, Exercise 15.2, in your PRG and answer the exercise questions.

Path Selection Considerations

Path selection varies according to the complexity of the network. For example, in a single-domain environment, with only one host and a single channel-attached communications controller, there can only be one path; that is, over the channel. Path selection, in this case, is trivial; however, it is required. For single-domain networks with link-attached NCPs and cross-domain communication networks, the path selection task requires a more thorough analysis of the physical links between subareas.

Once the network paths have been selected, they are defined to VTAM in PATH definition sets. VTAM uses the PATH definitions to build the transmission header (TH) for PIUs outbound to the network. Similar PATH definitions must be coded in each NCP for forwarding PIUs in the network.

A review of some of the basic path selection considerations follows:

- Paths traverse transmission groups (TGs) between subareas. A transmission group between the VTAM host and a channel-attached communications controller is, by definition, always TG = 1 over the channel. TGs between communications controllers may have one or more physical links and there may be one to eight TGs. Each TG defines one physical route between two subareas.
- Paths are defined between an origin subarea and a destination subarea. A path between any two end points (subareas) in the network must be unique; that is, even though there may be multiple physical routes between the end points, each path exists as an explicit route between the two subareas. Uniqueness is maintained by assigning a number to the path in the form of an explicit route number (ER#). This uniqueness must be maintained at any intermediate node in the path by using the same ER# to reach the same end point (destination) subarea.
- Each path between the two end points must be reversible. That is, in a multisystem network the participating host access method at the other end point must define a reverse path over the same physical route in the opposite direction. The same is true for link-attached NCPs in a single or multihost system. The ER# in the opposite direction need not be the same as on the incoming path; however, use of the same number is recommended. Again, any intermediate node must maintain the uniqueness for the reversible path.
- Any intermediate node along a selected path must have a PATH definition which has the same ER# to the destination subarea as the one used at the origin subarea. Also, the intermediate node must have a path defined for the reversible route to the origin subarea using the same ER# which was used in the reverse PATH definition.

The intermediate node will be either an NCP or another VTAM host.

- If alternate paths are available, each path must be defined between end points as an explicit route and must be reflected in all intermediate nodes. Up to eight explicit routes (ER0 to ER7) may be defined between any two end point (destination) subareas. The eight paths may traverse different intermediate nodes.

Additional factors that should be considered when selecting the paths are:

- Relative speeds of the communications links in the network
- Volume of data over each link
- Availability of processing cycles and storage buffers in either the host processors or communications controllers.

Coding to Define Routes Between Subareas

All routes used by VTAM to communicate with other subareas in the network, such as an VTAM with one or more NCPs, must be defined in PATH definition sets. VTAM directly to VTAM across a channel-to-channel adapter is another case where PATH definitions are required. VTAM PATH statements are coded to define both explicit and virtual routes to destination subareas.

VTAM PATH definitions are not assembled or link-edited. They are cataloged in the VTAM definition library (member in MVS VTAMLST or VSE source library, filetype of VTAMLST in VM). The PATH definition set name will be the member or file name in the library. There can be multiple sets of PATH definitions and each set may consist of one or more PATH statements.

PATH definition sets may be activated at VTAM startup via the configuration list or with the network operator VARY command. Once active, PATH definitions may not be inactivated; however, additional PATH definition sets may be activated to include new routes.

PATH definitions must also be included in NCPs to define explicit routes that pass through it. Virtual routes are not coded in single network NCPs. Virtual route information used by the NCPs is included in the PIU transmission header (TH).

PATH Statement Format

The format of the PATH statement is shown in Figure 15-7. The PATH statement (one or more) is the only statement used in a PATH definition set.

```

[ symbol ] PATH DESTSA=n|(n1,n2,n3,...)
              [,ER0=(adjsub[,tg#])]
              [,ER1=(adjsub[,tg#])]
              ●
              ●
              [,ER7=(adjsub[,tg#])]
              [,VR0=er#]
              [,VR1=er#]
              ●
              ●
              [,VR7=er#]
              [,VRPWS00=(min#PIUs,max#PIUs)]
              ●
              [,VRPWSvp=(min#PIUs,max#PIUs)]
              ●
              [,VRPWS72=(min#PIUs,max#PIUs)]

where v=Virtual route number
      p=Transmission priority

```

Figure 15-7. The PATH Statement

The coding for the path statement is as follows:

- symbol is any valid assembler language name. It is optional and is ignored if present.
- DESTSA = specifies the subarea number of the destination subarea for the virtual and explicit routes being defined in this PATH statement. More than one destination subarea can be specified in a given PATH statement, provided that all ER and VR definitions apply for each subarea specified. Multiple destination subarea numbers are enclosed in parentheses.
- ER0 identifies explicit route 0 from the host to the destination subarea(s) specified in the DESTSA parameter. ER1 through ER7 are similarly specified.
- adjsub identifies the adjacent subarea which VTAM sends PIUs enroute to the DESTSA. The adjacent subarea, on receipt of PIUs, is responsible for forwarding the PIU on the same ER# to its own adjacent subarea according to its own PATH definitions.
- TG# represents the transmission group over which PIUs are sent to the adjacent subarea.

All channels between VTAM and NCPs are specified as transmission group TG1. VTAM will always assume TG1 is to be used to its adjacent subareas. The primary reason for TG# in the VTAM PATH statement is for coding compatibility with the NCP PATH macro.

- VR# identifies the virtual route number and points to an ER# for this destination subarea.
- VRPWSvp allows the user to override the minimum and maximum virtual route pacing window sizes for any previously defined VR for a particular destination subarea. A virtual route's *window size* is the number of PIUs which flow between subarea nodes before the receiving node responds with a virtual route pacing response. This allows automatic monitoring of network congestion and dynamic change of window size between nodes in the route. The window size is varied between the calculated or VRPWSvp minimum and maximum values. Coding VRPWSvp can be of significant value when tuning a network.

Example VR Mapping

As an example, if a particular virtual route (VR1) is to map to a particular explicit route (ER3), the VR# coding would be:

VR1=3

For this simple example, VTAM would determine a class of service requirement for VR1 from the COS table, locate the appropriate DESTSA in the PATH table, find VR1=3, and use the ER3 adjacent subarea number (and TG1) to route session traffic. The ER number 3 would be included in the TH of the PIUs for message forwarding. This relationship was shown in the discussion of Figure 15-5.

For session traffic bound for a single channel-attached NCP; that is, SLUs defined in the NCP, DESTSA would be that of the NCP. Also, the adjacent subarea number would be that of the NCP. When more than one NCP is channel-attached to the same VTAM host, ERs may be chosen to route traffic through one NCP to the other if the NCPs are interconnected.

Route Selection and Session Setup Example

Refer to Figure 15-8 for an example configuration to review route selection during the activation and logon sequence. A particular LU (L4E79A) defined in a link-attached NCP (NCP SA4) is used for the sequence description. The tables and definitions for the example, beginning with Figure 15-9, represent user coding related to L4E79A.

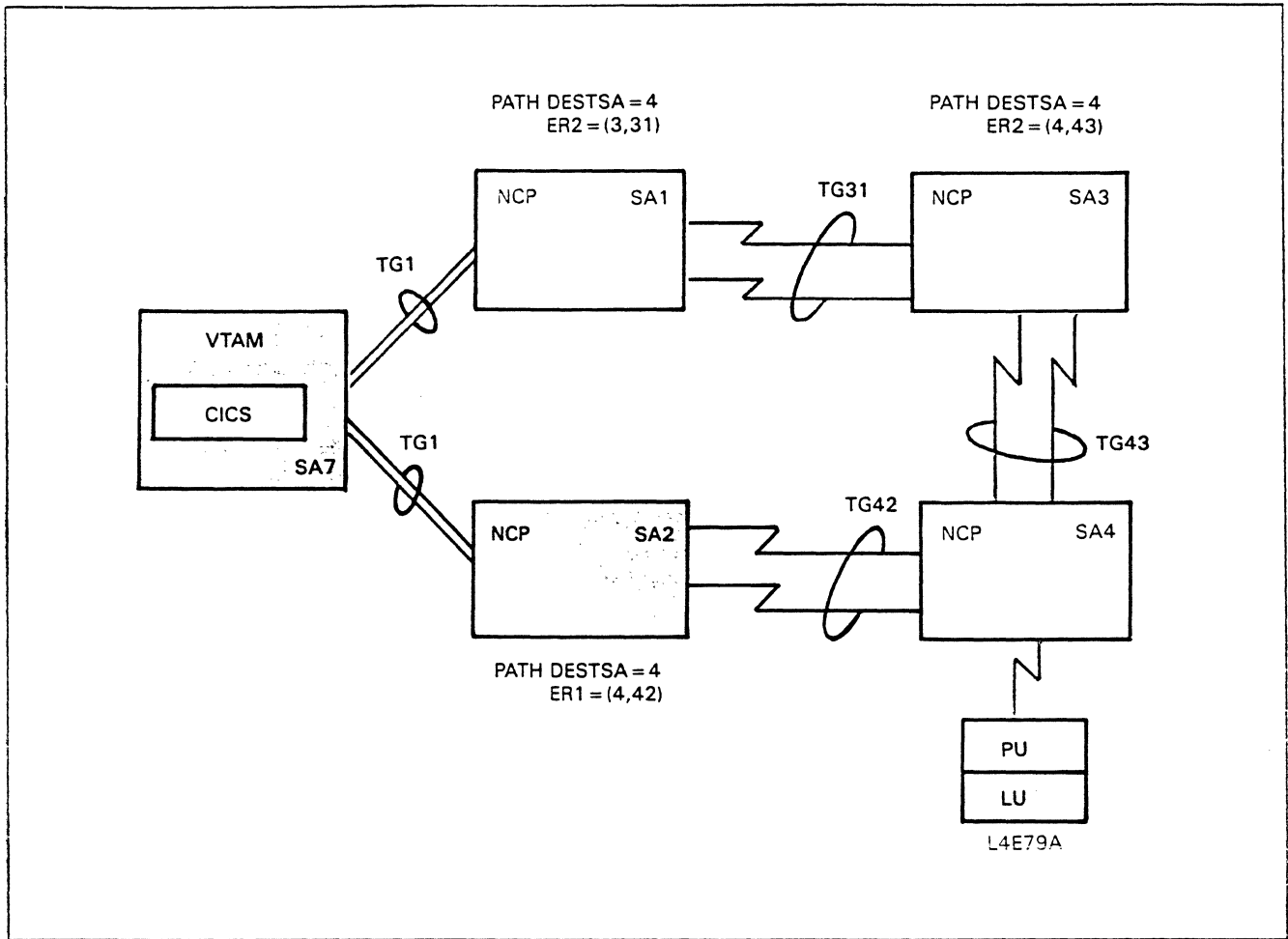


Figure 15-8. Example Route (ER) Selection Configuration

Assume that the network operator has activated the VTAM PATH definitions, NCP SA1, NCP SA3, and NCP SA4 (NCP SA2 is assumed to be inactive). Also, the line and PU for L4E79A are assumed to be active.

1. The network operator enters:

```
V NET,ACT,ID=L4E79A
```

The command asks for an LU activation which requires SSCP to send an activate logical unit (ACTLU - an SNA expedited command) to the LU.

SSCP will search the resource definition table (RDT) to find the L4E79A entry. It is found in the RDT entries for NCP SA4. Figure 15-9 shows the L4E79A definition as coded in the NCP SA4 major node.

```

L4E79A    LU    LOCADDR=2,
           ISTATUS=INACTIVE,
           MODETAB=MODE3279,
           DLOGMOD=LM79,
           USSTAB=USS79,
           SSCPFM=USSSCS

```

Figure 15-9. Sample LU Definition in NCP SA4

Note: The logon mode table (MODE3279) and the USS table (USS79) are already in storage. VTAM loads all LU tables coded in the NCP major node when the NCP is activated regardless of the LU ISTATUS coding.

SSCP now knows which subarea to associate with L4E79A (that is, DESTSA = 4) and the element address to use (found in the RDT entry) to build the transmission header (TH) destination address fields. The four-byte destination subarea field (DSAF) will be 4 and the destination element field (DEF) will contain the element address within subarea 4.

2. Next, the SSCP needs to find the routing information in order to complete the ACTLU PIU.

SSCP will look for the user COS table to find a COS entry ISTVTCOS for the SSCP-LU session.

Note: ISTSDCOS, if present in VTAM's module library, is loaded into storage at VTAM startup time. Figure 15-10 shows the sample COS table. The first ISTVTCOS entry is coded for virtual route 1 with a transmission priority of 2. The next SSCP reference is to the VTAM PATH definition to find the explicit route to use.

```

           COSTAB
ISTVTCOS  COS    VR=(1,2),(2,2)
BATCH    COS    VR=(1,0),(2,0)
CICS     COS    VR=(1,2),(2,2)
           COS    VR=(2,1),(1,1),(0,0)
           COSEND

```

Figure 15-10. ISTSDCOS - Sample COS Table

3. The sample VTAM PATH definition, Figure 15-11, shows the PATH statement for DESTSA = 4.

PATH	DESTSA=4,	x
	ER1=(2,1),	x
	ER2=(1,1),	x
	VR1=1,	x
	VR2=2	

Figure 15-11. Sample VTAM PATH Definition Set

SSCP finds the DESTSA = 4 entries in the PATH definition and locates VR1 which maps to explicit route 1 (ER1). Since ER1 is over TG1 (the channel) to adjacent subarea 2 (NCP SA2), SSCP looks at the status of ER1 to SA4 and finds it inoperative (NCP SA2 is inactive). SSCP takes the second COS entry (VR2, TP2) and again looks at DESTSA = 4 to find VR2 mapped to ER2. ER2 is available since the NCPs (SA1, SA3, and SA4) are active. LE79A's ACTLU PIU is therefore sent over TG1 (the channel) to NCP SA1 for forwarding on ER2.

4. Refer to the configuration in Figure 15-8 on page 15-17 for the NCP PATH definitions used to forward the PIU.
- NCP SA1 looks at the TH to find the DSAF (subarea 4).
 - Subarea 4 is compared to its own subarea number (1), to see if it is to perform the boundary node function for the PIU.
 - Since the PIU is destined for another subarea, NCP SA1 gets the ER number (2) from the TH and looks in its PATH definitions for DESTSA = 4 to find which adjacent subarea (3) and TG (31) to use.
 - NCP SA1 transmits the PIU to NCP SA3 over TG31 in an SDLC frame.

NCP SA3 will perform the same steps as NCP SA1 and send the PIU to NCP SA4 over TG43.

NCP SA4 finds that it is to perform the boundary node function for the PIU (the destination subarea field matches its own). Therefore, it searches its own tables using the element address in the TH. NCP SA4 does a format identification (FID) conversion from FID4 to FID2 and transmits the ACTLU PIU to L4E79A's PU in an SDLC frame.

The PU gives the ACTLU to the LU at local address 2 (L4E79A).

L4E79A accepts the ACTLU and sends a positive response back to VTAM. The response will traverse the same route (the reversible route) as the ACTLU. Processing of the response PIU would be similar to the ACTLU PIU as it is transmitted over the reverse route. To simplify the example, the required NCP PATH definitions for the reverse route are not shown.

5. When SSCP receives the positive response PIU from L4E79A, it marks it active (status ACTIV - SSCP-LU session established). Next, SSCP looks at the LU definition (RDT entry) to see if there is a user USS table pointer. In this case, USSTAB=USS79 associates L4E79A with USS table USS79. The source code used to assemble and link-edit USS79 is shown in Figure 15-12.

USSTAB	
•	
USSCMD	CMD=CICS,REP=LOGON
USSPARM	PARM=P1,REP=APPLID,DEFAULT=DBDCCICS
USSPARM	PARM=P2,REP=LOGMODE
•	
USSMSG	MSG=10,TEXT='GOOD MORNING'
USSEND	

Figure 15-12. USS79 - Sample USS Table

SSCP looks for a coded MSG-10 in USS79, and finding one, it needs to know how to format the RU in the PIU for MSG-10. Looking at the LU definition (see Figure 15-9) it finds SSCPFM coded USSSCS which tells SSCP to use SNA character string (SCS) RUs in the SSCP-LU session.

SSCP builds the MSG-10 RU (GOOD MORNING), completes the PIU, and sends it (using ER2, the established SSCP-LU session ER) to NCP SA1 for forwarding.

6. Once MSG-10 is displayed at L4E79A, an end user may enter a logon request. For this example, assume the LOGON command entered is:

CICS

When L4E79A's PU is polled by NCP SA4, the character-coded message (RU of CICS in a FID2 PIU) is sent in an SDLC frame to NCP SA4. NCP SA4 converts to a FID4 PIU and forwards it to VTAM via NCP SA3.

7. SSCP checks the incoming PIU for information such as:

- Where did the PIU come from?

It looks at the origin subarea field (OSAF) in the TH (4, in this case) and the origin element field (OEF) for the element address (belonging to L4E79A, in this case).

- Is the format indicator in the RH on?

If the format indicator is on, assume it is a character-coded logon request.

When it is a character-coded logon request, as in this example, SSCP looks at the associated USS table (if there is one) to see if there is a user-coded character translation table (TABLE = pointer). Since no TABLE = pointer is in USS79, SSCP uses the default character translation table STDTRANS to translate the characters in the logon request RU.

After translation, SSCP compares the characters (CICS) to the USSCMDs in the associated USS table USS79 (see Figure 15-12).

8. To build the CONTROL-INITIATE (CINIT) command, SSCP finds the following:

- The APPLID name (DBDCCICS) from the DEFAULT APPLID in USS79.

No overriding APPLID appeared in the logon request. If there were no DEFAULT APPLID in USS79, SSCP would reject the logon request with a USS MSG-5 'UNSUPPORTED FUNCTION' and send it to L4E79A with the same routing as MSG-10 (ER2 - the SSCP-LU session ER).

- The symbolic name L4E79A using the origin element field in the TH to search the NCP SA4 major node RDT
- The logmode entry name LM79 from the LU DLOGMOD definition and the logon mode table name MODE3279

SSCP checks the logon mode table (MODE3279) to see that LM79 is a valid entry. The MODE3279 table is shown in Figure 15-13.

Note that no overriding LOGMODE name was in the request from L4E79A and no DEFAULT was coded for LOGMODE in USS79's CICS command.

```

MODETAB
●
MODEENT LOGMODE=LM79, . . . . .
        COS=CICS
MODEEND

```

Figure 15-13. MODE3279 - Sample Logon Mode Table

When the CINIT is ready, SSCP drives the LOGON exit in CICS.

9. CICS will build the VTAM control blocks for L4E79A and issue an VTAM macro (OPNDST) to send the BIND image to L4E79A to establish the LU-LU session.
10. SSCP processes the OPNDST request and builds the BIND command PIU destined for L4E79A.
11. The destination subarea (4) is known for L4E79A, but SSCP needs to determine which explicit route to use for the CICS-L4E79A (LU-LU) session.
12. The logmode entry LM79 is checked for a COS pointer, in this case the COS entry name CICS is found (see Figure 15-13).
13. Using the COS entry name CICS, SSCP searches the COS table ISTSDCOS (see Figure 15-10 on page 15-18) and locates the first VR#-TP# pair (1,2).
14. Looking at the PATH definition for DESTSA=4, SSCP finds that VR1 maps to ER1. Traffic on ER1 is to be sent to NCP SA2. Since both are still inactive, ER1 is unavailable and SSCP will look for a second COS entry.
15. The second VR#-TP# pair (2,2) is used to find VR2 mapped to ER2 in the DESTSA=4 PATH entry. ER2, as it has been for the SSCP-LU session, is available and will be used for the LU-LU session.
16. During the LU-LU session, transmission priority 2 will be in effect. If any traffic which has a lower transmission priority is queued in NCP SA1 or NCP SA3, such as any sessions using the unnamed COS entry which has TP1, the CICS-L4E79A PIUs will be sent first.

This example used some assumptions in order to present the route selection procedure. If NCP SA2 were activated at the same time as the other NCPs, the entire example sequence would result in different routes for the SSCP and LU-LU sessions. The procedure would remain the same as described, however, the session traffic would flow through NCP SA2 to NCP SA4, a more direct route.

■ ■ ■

Please turn to Mini-Course 15, Exercise 15.3, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 16

VTAM Cross-Domain Communication Multisystems

Mini-Course 16. VTAM Cross-Domain Communication Multisystems

Introduction

VTAM has the capability to communicate with logical units which are defined in other VTAM domains.

Communications may be between application programs as well as from application program to device logical unit(s) across domain boundaries. Once data leaves a domain, it is under the control of any intermediate nodes, such as NCP communications controllers, until it reaches a destination subarea in another domain. The destination subarea, another VTAM or an NCP, performs the boundary network node function for the participating LU once an LU-LU session is established.

With cross-domain communications there are several considerations for planning, defining, and operating the network.

Planning and Definition

Planning for cross-domain communications involves a joint effort on the part of the designers/coders for the respective host systems. In some cases it may be beneficial to have a single group of people, perhaps even one person, do the planning and definitions for the network. In any case, certain ground rules must be followed for the network definition.

Figure 16-1 shows a simplified configuration with two VTAM hosts and two NCP communications controllers connected by a multilink transmission group. Refer to Figure 16-1 for the discussion which follows.

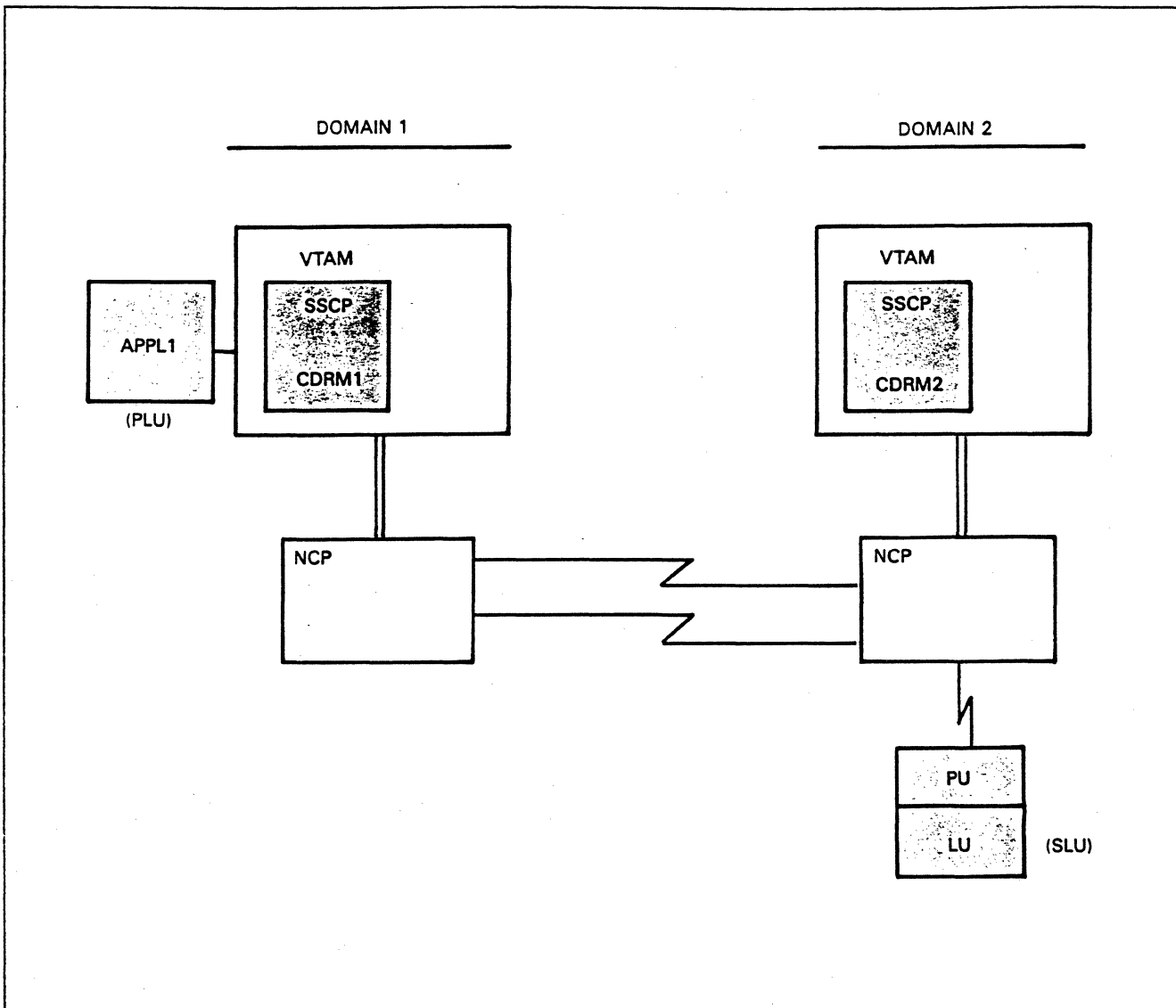


Figure 16-1. A Basic Cross-Domain Configuration

When cross-domain communications are to be established, each host access method must have code in its systems services control point (SSCP) to manage the sessions. This code is called a cross-domain resource manager (CDRM). The SSCP's CDRM is responsible for setting up a session with the CDRM of another domain and for setting up the cross-domain LU-LU sessions.

A CDRM to CDRM session must be established before any LU-LU sessions can be initiated. An application program will always be the primary LU (PLU) in one of the domains, and the secondary LU (SLU) in the other domain will be another application program or a device logical unit. The PLU in one domain will be a cross-domain resource (CDRSC) to the domain that owns the SLU. At the same time, the SLU will be a cross-domain resource to the domain that contains the PLU.

Each host access method would have a single CDRM defined for itself, and a CDRM definition for every other resource owning host with which it will communicate. CDRMs are defined in one or more VBUILD TYPE = CDRM major nodes. Each CDRM, including its own, is defined with a CDRM statement. For example, in Figure 16-1 both VTAM hosts would contain a VBUILD TYPE = CDRM major node which defines both CDRM1 and CDRM2.

VTAM sees the resources in its own domain as it does in a single-domain system, and it sees the resources in other domains that are defined as cross-domain resources. A resource in another domain is defined in one of two ways:

- It is defined in a cross-domain resource major node (VBUILD TYPE = CDRSC) with a CDRSC statement.
- It is defined dynamically at the time of a session initiation request.

Cross-domain resources are always logical units. VTAM application programs and any of the SNA or non-SNA devices supported by VTAM may be defined as cross-domain resources.

Note: This may not be the case with other access methods, such as TCAM; see the introductory manuals for each access method for specific cross-domain terminal support.

An application program in one domain is available to:

- The logical units and non-SNA terminals in its own domain
- The cross-domain resources in another domain.

The application program is not aware of whether a resource is locally attached, remotely attached in its own domain, or in another domain. Also, it is not aware when a resource represents another application program; it is only aware of a logical unit with a particular set of characteristics.

Cross-domain communication implies a link between the access methods. The links in VTAM to VTAM cross-domain communication include:

- Communications links (lines) via two or more communications controllers
- A communications link via channels connected to a single communications controller from two host systems (sometimes referred to as twin-tailing)
- A communications link via a channel-to-channel adapter (CTCA) between two host systems
- A communications link via VSE or VM supported communications adapters.

In certain network configurations, combinations of these links would be valid. For example, a link may connect a VSE VTAM host with a communications adapter to an MVS VTAM host with a communications controller.

Note: Other possibilities exist when one or more of the access method hosts is running in a Virtual Machine/System Product (VM/SP) environment.

Network Definition

There are several areas of consideration for defining the network for each host access method. Definition areas to consider are:

- CDRM major nodes
- CDRSC major nodes
- PATH definition sets
- NCP (or CA) major nodes
- Start options
- Configuration lists.

CDRM Major Node

The CDRM major node begins with a VBUILD statement with TYPE=CDRM. Recall that the major node name will be the cataloged member name, not the VBUILD statement name.

All participating cross-domain resource managers are defined with a CDRM statement. VTAM builds a CDRM in its own domain when it encounters a CDRM subarea number equal to its own HOSTSA number (a start option). Each domain requires a definition statement of its own CDRM as well as a definition for external CDRMs with which it will communicate.

Very little user information is required for the CDRMs. Information is exchanged between CDRMs during the SSCP-SSCP session establishment. The operands of the CDRM statement are: SUBAREA, ELEMENT, ISTATUS, VPACING, CDRDYN, CDRSC, and RECOVERY.

Code a single CDRM statement for each participating access method. The SUBAREA operand value is the assigned subarea number for each CDRM host. The format for the CDRM statement is shown in Figure 16-2.

```
cdrmname  CDRM  SUBAREA=...,ELEMENT=...,ISTATUS=...,      x
              CDRDYN=...,CDRSC=...,VPACING=..
```

Figure 16-2. CDRM Statement Format

The `cdmname` is the name assigned to the CDRM. It must be the same name in all participating access methods to represent this VTAM host. Since all CDRMs are defined in each domain, the same CDRM major node definition can be cataloged in all domains. However, initial `ISTATUS` may present a problem as will be seen later in the section `CDRM ISTATUS`.

ELEMENT

`ELEMENT` forms a part of the network address for a CDRM. For VTAM there is only a single element per subarea; therefore, the default value of 1 is used in every VTAM CDRM definition. TCAM CDRMs may have more than one element per subarea. In those CDRMs, `ELEMENT` specifies a number from 0 to the maximum number of elements in the subarea.

VPACING

`VPACING` establishes the number of inbound messages from other CDRMs before a pacing response is returned. There is no requirement for `VPACING` to be the same for all CDRMs; however, the user may choose to define a common value for all CDRMs. The default value is 63.

RECOVERY

VTAM will automatically attempt to recover a lost SSCP-SSCP session if `RECOVERY` is allowed to default to YES. Both the host CDRM and the external CDRM must be coded for recovery to permit the attempt to reestablish the session.

Dynamic Cross-Domain Resource Definition Reduction

Cross-domain resources must be represented in VTAM with their LU name and their owner's CDRM name before cross-domain LU-LU sessions can be established. The two names can be supplied to VTAM in two ways:

- In a cross-domain resource (CDRSC) major node, VTAM builds control blocks for the cross-domain resources when the CDRSC major node is activated.
- During the cross-domain session establishment procedure, VTAM builds the control blocks dynamically from information in the cross-domain session initiation requests.

Dynamic definition applies to secondary logical units (SLUs) which can request sessions with VTAM application programs across domain boundaries. In the SLU's domain, the application program must be identified in a CDRSC major node in order for its CDRM to know where to send the logon request.

In networks with large numbers of LUs, it is advantageous to use the dynamic definition approach. The major advantages are:

- It can save processor main storage when only a subset of the cross-domain resources are in session with the application program.
- The amount of coding for CDRSC major nodes is reduced.
- When new devices are added to the external domains, CDRSC major node definitions need not be changed to reflect the additions.

If, in the external domain, there are stations that are not to be allowed to logon to this domain's application programs, dynamic definition should not be used. CDRSC major nodes should be coded for the stations (LUs) which will be allowed to initiate sessions.

Parameters in the CDRM definition allow the user to select various combinations between coded CDRSC major nodes and dynamic definition.

CDRDYN in a host's own CDRM statement tells VTAM whether or not it will dynamically define resources belonging to other domains. If CDRDYN=NO (the default) appears in its CDRM statement, it is not allowed to dynamically define resources. If CDRDYN=YES is coded, session requests from other CDRMs will contain information for dynamically building CDRSC control blocks for the session.

CDRSC=OPT|REQ is coded in an external domain CDRM definition to inform VTAM whether it is allowed to dynamically define resources for the external domain. If CDRSC=OPT is coded for the external domain, this VTAM may define its resources dynamically. CDRSC=REQ means that any resources belonging to that domain must appear in a CDRSC major node to have cross-domain sessions with this VTAM domain.

There are four basic possibilities when considering dynamic cross-domain resource definition reduction:

- All resources in the network may be dynamically defined by any VTAM host in the network.

Code CDRDYN=YES in all CDRM definitions.

Code CDRSC=OPT in all CDRM definitions.

- A host is authorized to dynamically define resources for any other VTAM host.

Other hosts are restricted from dynamically defining its resources.

Code CDRDYN=YES in its own CDRM major node definition.

Code CDRSC=REQ in its CDRM definition in all other hosts.

- A host is not authorized to dynamically define cross-domain resources (CDRSC major nodes are required).

Other VTAM hosts may dynamically define its resources.

Code CDRDYN = NO in its own CDRM definition.

Code CDRSC = OPT in its CDRM definition in the other hosts.

- CDRDYN = NO and CDRSC = REQ are coded (or defaulted) in all CDRM definitions.

This choice would mean dynamic cross-domain definition is not authorized for any host. CDRSC major nodes would be necessary for each host.

If there are only two VTAM hosts, these four basic combinations would cover all possible user choices. Once the number of hosts is greater than two, the number of possibilities becomes much larger. For example, CDRDYN is only looked at in a host's own CDRM definition to see if it is allowed to drive its dynamic definition code (YES or NO). For the same host's resources, all other hosts look in their own CDRM major node to see the value of CDRSC in its CDRM definition. Therefore, if its CDRSC is coded OPT in some other hosts and REQ in the rest, more than four combinations occur.

One way to plan for dynamic resource definition, if one of the basic possibilities is not desirable, is to:

1. List each host which will participate in cross-domain sessions.
2. Give each host in the list a name (that is, CDRMxx where 'xx' is the subarea number).
3. Determine, for each host, whether it will be allowed to dynamically define cross-domain resources. Mark the host with YES or NO.
4. List the names for each host, one at a time, of the other hosts which are capable (YES hosts from Step 3) and will be allowed to dynamically define its resources. Mark this list the OPT list.
5. List for each host in the same manner, one at a time, the names of all remaining hosts. Mark this list REQ. They will not be allowed to dynamically define its resources whether they are capable or not.

- Build a skeleton CDRM major node which contains a CDRM statement for all participating hosts. Include the SUBAREA number and code the CDRDYN value YES or NO from the list made in Step 3.

For example:

```

name      VBUILD TYPE=CDRM
CDRM05   CDRM   SUBAREA=05,CDRSC=      ,CDRDYN=YES
CDRM07   CDRM   SUBAREA=07,CDRSC=      ,CDRDYN=NO
CDRM09   CDRM   SUBAREA=09,CDRSC=      ,CDRDYN=YES
CDRM11   CDRM   SUBAREA=11,CDRSC=      ,CDRDYN=YES

```

- Build a CDRM major node for each host by duplicating the Step 6 skeleton major node. Also, name the major node for later cataloging in each host's VTAM definition library.
- Using the OPT list for one host from Step 4, go to every other CDRM major node and fill in CDRSC = OPT for that host's CDRM.
- Using the REQ list for one host from Step 5, go to every other CDRM major node and fill in CDRSC = REQ for that host's CDRM.
- Repeat Steps 8 and 9 for all hosts.

CDRSC Major Node

The CDRSC major node (VBUILD TYPE = CDRSC) is used to define resources that are in domains external to VTAM. An external domain may have resources available for its own use which are not needed by this VTAM. When coding definitions for external domain resources, only the minor nodes needed for cross-domain communication are coded as resources in the CDRSC major node. The other resources of the external domain are not known to this VTAM.

For example, using CDRM05 and CDRM07, the application programs (PLUs) in the CDRM05 domain must appear in a CDRSC major node in the CDRM07 domain in order for the device logical units (SLUs) in the CDRM07 domain to initiate cross-domain session requests.

When dynamic cross-domain resource definition is used, all of the CDRM07 domain's resources may attempt to initiate sessions with application programs in the CDRM05 domain without being defined in a CDRSC major node in CDRM05's domain. The method of coding was shown in the section *Dynamic Cross-Domain Resource Definition Reduction* of this mini-course.

If any restriction is to be made between resources in the external domain, there are three methods of identifying the resources:

- Code a CDRSC major node to include only the names of minor nodes to be permitted access. Also, in this case, code CDRSC = REQ in the external domain's CDRM definition to prohibit dynamic definition.
- Code a VTAM authorization exit routine to disallow logons from restricted terminals.
- Use (or code) authorization facilities in the VTAM application program.

VTAM only needs to know three attributes of a cross-domain resource in order to begin session establishment. They are:

- The minor node name by which a resource is known in its own domain. That would be the APPL name, the LU name, the LOCAL name, or the TERMINAL name coded in the external domain's own major node.
- The name of the CDRM that owns the resource. This would be the name used on the CDRM statement for the external domain.
- The initial status of the external resource as looked at by this VTAM.

Note: This is a logical status to this domain; the actual resource may be either active or inactive in its own domain.

Other information necessary to complete the LU-LU session setup, such as a logmode table entry name, is passed in the CDRM-CDRM session.

The format of the CDRSC statement is shown in Figure 16-3.

```
name CDRSC CDRM=cdrmname, ISTATUS=ACTIVE;INACTIVE
```

Figure 16-3. CDRSC Statement Format

The name must be the same 1- to 8-character name coded in its owning host access method (refer to attribute 1 above).

The cdrmname is the name of the owning CDRM. The name must be the same as one of the CDRMs in the CDRM major node.

Cross-domain sessions can be established only after certain network requirements are met. One of the important considerations is how the initial status is coded for each CDRM and each CDRSC. (CDRM ISTATUS and CDRSC ISTATUS will be discussed later in this mini-course.) Another consideration is whether there are any intermediate subareas between the respective CDRMs.

Shadow Resources

VTAM will allow names of LUs defined in its own domain to appear in its CDRSC major nodes. This feature allows a single coded CDRSC major node to appear in multiple domains.

This is the only case in VTAM where a minor node name may appear in two definitions. When VTAM finds a minor node name in a CDRSC major node that is defined in its own domain, it will treat one as a real resource and the other as a shadow resource depending on the status of the PU associated with the LU. If the LU's PU is active, the LU is real and the CDRSC for the same LU (name) is the shadow. If the PU is inactive, the CDRSC becomes real (VTAM assumes it is a real cross-domain resource) and the LU in its domain becomes the shadow resource.

An application program name CDRSC is always a shadow resource since it is either running/active or not running/not active and has no PU. This is also true for local non-SNA devices since they have no PU. Session requests in the same domain for a resource in shadow state will fail.

■ ■ ■

Please turn to Mini-Course 16, Exercise 16.1, in your PRG and answer the exercise questions.

Definition Preparation

Network Diagram

As with almost any data processing definition, a detailed diagram can help organize a systematic plan, help communicate with others, and help reduce errors in the final product. This is especially true when designing a cross-domain communications network.

Many approaches to building a network diagram can be developed. One possible approach is outlined below. Figure 16-4 shows an example of a completed network diagram for three domains. The numbers in the example network diagram correspond to the steps in the outline which follows.

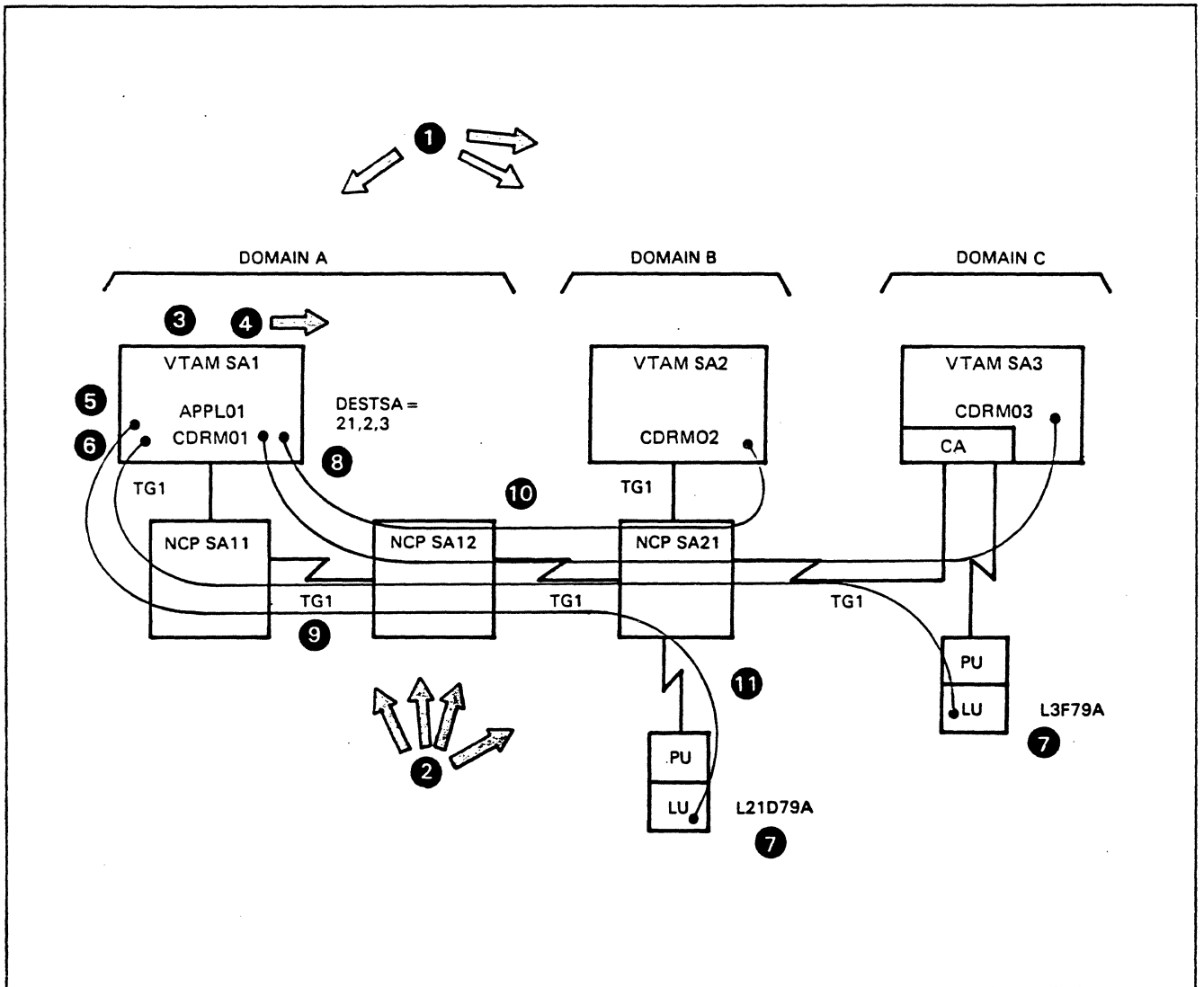


Figure 16-4. Sample Cross-Domain Network Diagram

1. Start with a host CPU location drawing. This may be a schematic drawing or an outline of a state/country map. An important consideration is the relative geographic location of the various host computers for later planning of the communications lines.
2. Draw a block for each host to represent an NCP communications controller or a communications adapter. Also, draw blocks for any link-attached NCPs.
3. Mark each host with its teleprocessing access method.
4. Determine existing subarea numbers, if any, or assign unique subarea numbers for each VTAM host, each non-VTAM host, and each NCP communications controller. Recall MAXSUBA, the maximum number of subareas start option in non-ENA systems. It must be the same in all participating host access methods and NCPs. All subarea numbers must be in the range of 1 to MAXSUBA and need not be contiguous numbers.

In VTAM cross-domain communication with NCP via a communications adapter (CA), the subarea number of the CA will be that of its host.

You may also want to make reproductions of the current diagram to maintain clarity for the next step.

The optional reproductions will be handy at this point if there are several participating application programs. Use one diagram for each program, unless it will include program-to-program communication.

5. Mark the host with the application program which is to take part in cross-domain sessions. Use the name on the APPL statement (the minor node name) from the application program major node in each access method definition.
6. Add a cross-domain resource manager (CDRM) for each participating host access method. Determine the names of existing CDRMs, or select unique 1- to 8-character names for the CDRMs.
7. Draw (or list) all resources for each participating application program (separate diagrams) that are not in its own domain and are not to be dynamically defined cross-domain resources.

The resource may be another application program or a device logical unit.

Identify each resource with the minor node name assigned within its own domain. If the resource is defined in an NCP, its identity should be associated with that NCP. If the resource is an application program, a locally attached logical unit or terminal, or a device attached via a communications adapter, associate the minor node name with its access method host.

8. Make a list, in each application program's own domain, of the subarea numbers of all external domains which will control needed resources. Also, list the subarea numbers of all NCPs which contain the definition of needed resources. Title this list for destination subareas DESTSA = .
9. Draw the communications links that are to connect the various host systems. The communications links will connect either to NCP communications controllers or to communications adapters.

Group parallel links between NCPs into transmission groups and assign TG numbers.

Assign TG numbers to all single link TGs.

Only one CA line (that is, single link TG) may connect its host with another subarea. The transmission group number for a CA line is always TG1.

In a CA to NCP connection, polling will always be done by the NCP. In NCP to NCP connections, polling is done by the NCP with the highest subarea number. Adjacent NCPs determine this relationship during their initial exchange of identification.

10. Draw paths from an application program domain to each domain host which owns needed resources. Ownership, in this case, means the host in which the resource will be activated.

In this step select paths between two host access methods for the CDRM to CDRM session. Application program to resource path selection will be made in the next step.

Up to eight of these paths, or alternate routes, may be drawn between the application program domain and the other end point (destination) domain. This will occur with one or more of the following:

- Parallel links between communications controllers
- Multiple interconnected communications controllers in the network
- A VTAM system acting as an intermediate node.

Note: When multiple paths are available between any two domains, selection and definition of paths becomes more complex. The full scope of route planning and PATH definition are treated in other courses. The intent of this mini-course is to present VTAM cross-domain communications in the light of overall VTAM requirements, rather than explore the details of multisystems and networking. In complex networking situations, it may be appropriate to use a route planning aid such as the IBM Routing Table Generator (RTG) program offering.

11. When the paths have been selected for the CDRM to CDRM sessions, the next step is to select the paths from the application program's domain to the subareas where the end point resources are defined. If a resource is defined in the same subarea as the end point access method CDRM, the paths to the resource are the same ones selected for the CDRM to CDRM session in Step 10.

When the resource is defined in an NCP communications controller in the end point domain (that is, becomes ACTIVE in that domain), the resource is associated with a different subarea number; that is, NCP vs. CDRM subarea numbers. Since paths are defined between subareas, a new set of eight paths may be selected from the application program subarea to the NCP subarea. In most cases, the application program to resource paths will follow the same physical routes as the previously selected CDRM to CDRM routes; however, they may be different.

ISTATUS

The initial active or inactive status of a CDRM (ISTATUS) tells VTAM whether to activate a CDRM when the major node is activated. VTAM will mark its own CDRM active; that is, ready for session establishment with another CDRM. However, the initial status for external CDRMs tells VTAM whether to attempt to establish a session with the other CDRM. Before the CDRM-CDRM cross-domain session can be established, several requirements must be met. They are:

- Communications lines and link stations must be active. A link station is either in another VTAM domain (PU__T5) or in a communications controller (PU__T4).
- The required PATH definition set must be active. If intermediate subareas are involved, they must also be active in their own domains.
- VTAM's own CDRM must be active.
- The external CDRM must be active in its own domain.

CDRM ISTATUS

If an external CDRM statement is coded `ISTATUS=ACTIVE`, VTAM will attempt to establish a CDRM to CDRM session as soon as its CDRM major node is activated. The result of the attempt depends on the four steps listed in the previous subsection `ISTATUS`. However, to satisfy these requirements, a user may choose to have only the VTAM CDRM initially `ACTIVE`, and code all external CDRMs `ISTATUS=INACTIVE`. The domain operator would then activate the external CDRMs after initialization of VTAM and the network.

Note: If there are several external CDRMs to activate, code two CDRM major nodes rather than have the domain operator issue multiple commands.

- For the domain's CDRM major node, code `ISTATUS=ACTIVE` on its CDRM statement and place its major node name in the `CONFIG` list.
- For the external CDRM's major node, code each with `ISTATUS=ACTIVE` and the major node name not in the `CONFIG` list.

VTAM will attempt to activate all external CDRMs after the operator activates the one external major node.

Once all CDRMs are active in their own domains, the request for a CDRM to CDRM session may actually come from either host. Only one request is necessary for the CDRM-CDRM session to activate. When no routes are available (that is, the reverse routes are not active), the CDRM session request is queued.

CDRSC ISTATUS

Cross-domain resources are made available to VTAM when the `CDRSC` major node is activated, and either the resource has been defined with `ISTATUS=ACTIVE`, or the domain operator has issued a command to activate the resource.

Once the cross-domain resources are `ACTIVE` in the VTAM domain, they are ready for attempted session establishment. Before a session can be established with a resource, it must be active in its own domain and its CDRM must be in session with the external CDRM as described previously.

When dynamic cross-domain resource definition is being used, the `CDRSC` is considered active when the definition is completed.

A VTAM start option, `CDRSCTI=`, may be used to specify a time interval for dynamically defined `CDRSCs` to be held active after the LU-LU session completes. It is specified in seconds and defaults to 480 (8 minutes). If `CDRSCTI=0` is specified, VTAM immediately inactivates the `CDRSC` (that is, deletes the control blocks for the `CDRSC` when the LU-LU session completes).

Cross Domain and CONFIG

VTAM activates major nodes and path definition sets in the order they are placed in the configuration list. If any cross-domain major nodes are to be included in the list, they should be placed in the following order:

- Path Definition Set(s)
- NCP Major Node(s) and/or CA Major Node(s)
- CDRM Major Node(s)
- CDRSC Major Node(s).

Communications Adapter (CA) Major Node

VTAM cross-domain communication must be over an SDLC non-switched communication line that is connected point-to-point with either another 4300 Communication Adapter or with an NCP communications controller.

The only cross-domain communication lines in the network that are defined in the CA major node are the lines from the CA to an adjacent subarea. The CA major node definition for these lines consists of the following statements:

- The GROUP statement
- The LINE statement
- The PU statement for a 4300 CA (PU__T5) or an NCP communications controller (PU__T4).

The GROUP statement need only supply a minor node name, LNCTL=SDLC, and DIAL=NO.

LINE Statement

The LINE statement supplies the channel unit address of the line, the minor node name, and certain procedural options. The operands that apply to cross-domain lines are: ADDRESS, ACTIVTO, MAXBFRU, ISTATUS, PAUSE, REPLYTO, and RETRIES.

PAUSE and REPLYTO

PAUSE and REPLYTO only apply with a VTAM to VTAM (peer to peer) connection. PAUSE refers to a communications adapter (CA) that is acting as the primary station (polling station) and is not coded if the station is secondary to an NCP. REPLYTO, the primary station timeout value for response to poll, is only necessary if the one-second default is to be changed.

ACTIVTO

ACTIVTO (activity timeout) applies to the CA and specifies the time that the CA is to wait before signaling an error that it has not received an SDLC frame from an upstream NCP, a peer VTAM CA, or a VTAME CA.

The PU Statement

The PU statement identifies the cross-domain PU as either another CA (PU__T5) or an upstream NCP communication controller (PU__T4). A VTAM CA will always be downstream, or secondary, to an NCP.

The operands that must be coded are PUTYPE and SUBAREA.

SUBAREA

SUBAREA specifies the subarea number assigned to the cross-domain VTAM or NCP that this PU statement defines.

The PATH definition set for this CA link will have all VRs and ERs mapped to this SUBAREA number (adjsub) with a TG number of 1.

ADDR and TADDR

ADDR and TADDR can be coded; however, they may be omitted since they each default to the address character C1 with either PU__T4 or PU__T5. ADDR=C1 will always work.

The TADDR (temporary address) is used when VTAM is acting as a secondary station. As a secondary station it has no real address like a control unit on a line. TADDR gives the secondary VTAM a temporary 1-hexadecimal-character address for a primary (or polling) station to use. TADDR will always be used with an NCP connection, since VTAM is always a secondary (polled by) NCP. The TADDR character specified must be the same address character specified in the NCP generation for the VTAM CA station.

Example Cross-Domain Definitions

In an earlier subsection of this mini-course, a Network Diagram was drawn to illustrate how to prepare for cross-domain definition. The diagram, shown in Figure 16-4 has been updated with definitions for the respective domains and is shown in Figure 16-5. The definition requirements described in the preceding sections have been used.

Note that VR0 maps to ER0 throughout the example to keep class of service (COS) considerations to a minimum.

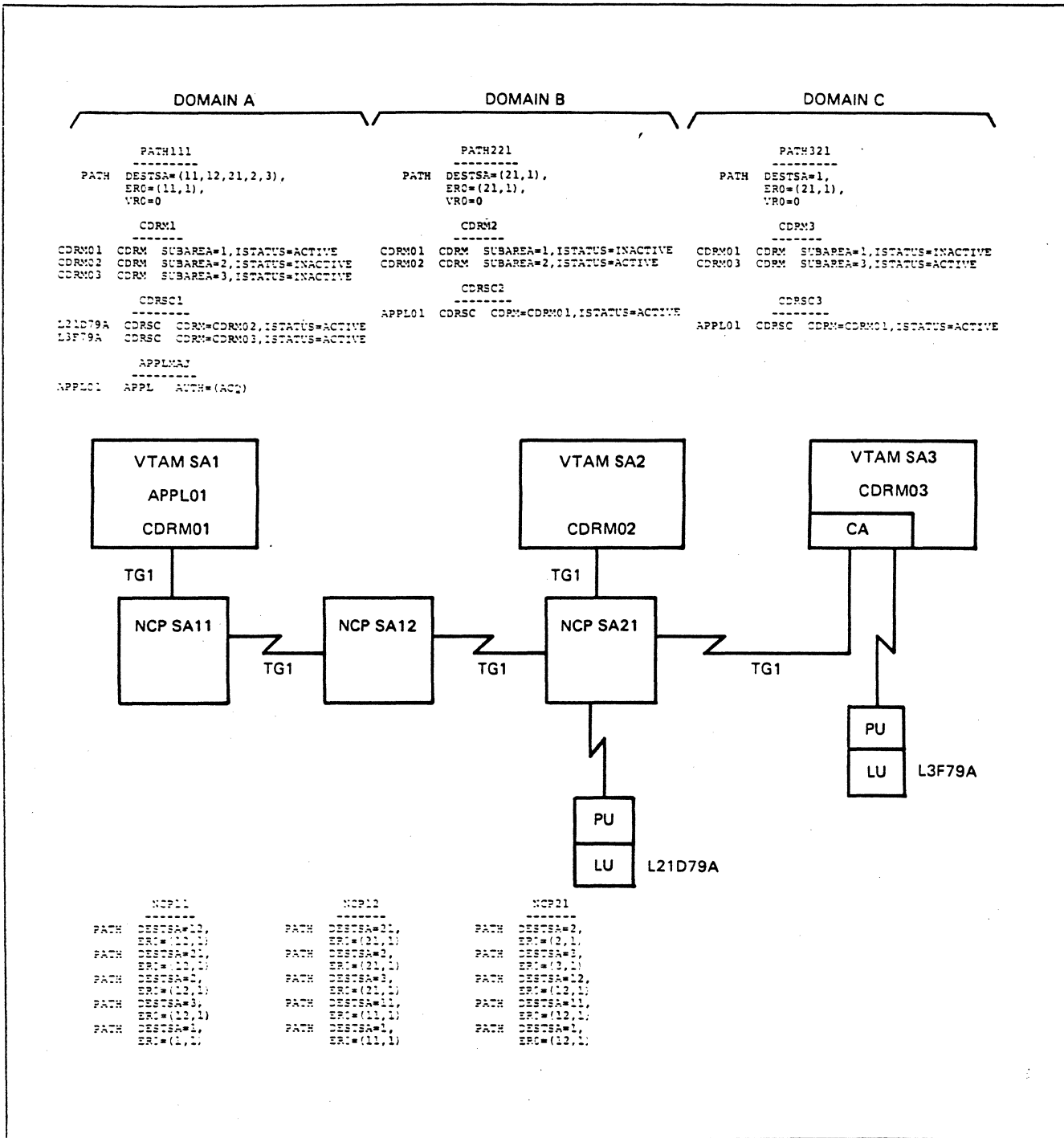


Figure 16-5. Example Cross-Domain Definitions

Operation of the example network shown in Figure 16-5 can be handled in several ways. Each domain may use a different method of bringing-up its system. In this case, the operation will be described using only typical operator commands for each domain.

Note: Assume that each domain's VTAM has been started and that all links and link stations are coded ISTATUS = ACTIVE in the NCPs.

Domain B Operator Commands

The operator of Domain B may use the following series of VARY commands after VTAM has been initialized.

```
V NET,ACT,ID=PATH221
V NET,ACT,ID=NCP21
V NET,ACT,ID=CDRM2
V NET,ACT,ID=CDRSC2
```

Note that these commands activate the major nodes and path definition set in Domain B only; no attempt is made to activate CDRM01 which is coded ISTATUS=INACTIVE.

Activation of NCP21 makes L21D79A ACTIVE in Domain B only. The SSCP-LU session is established and it is marked ACTIVE and ready for sessions with application programs in its own domain.

Activation of CDRM2 causes VTAM to activate CDRM code in the SSCP which makes CDRM02 ACTIVE. An entry is created for CDRM01 but marked INACTIVE.

Activation of CDRSC2 makes APPL01 active in Domain B only, regardless of whether it is active in Domain A.

Domain C Operation

The operator of Domain C will activate only the major nodes and path table of Domain C as follows:

```
V NET,ACT,ID=PATH321
V NET,ACT,ID=CAMN03
V NET,ACT,ID=CDRM3
V NET,ACT,ID=CDRSC3
```

Again, no attempt is made to establish a session other than the SSCP-LU session. CDRM03 and L3F79A are made ACTIVE in Domain C only. CDRM01 is inactive; therefore, no CDRM-CDRM session request will start.

CAMN03 represents the VBUILD TYPE=CA major node for the communications adapter; it contains the definition for L3F79A. The NCP to communications adapter contact procedure will complete since NCP21 is already active and sending exchange identification (XID) sequences to the CA link. The CA will respond with an XID and become secondary to NCP21. Communications could begin between Domains C and B over the link if the example called for it.

Domain A Operation

The Domain A operator will activate the major nodes and path definition set as follows:

```
V NET,ACT, ID=PATH111
V NET,ACT, ID=NCP11,LOAD=YES
V NET,ACT, ID=NCP12,LOAD=YES
V NET,ACT, ID=CDRM1
V NET,ACT, ID=CDRSC1
V NET,ACT, ID=APPLMAJ
```

Activation of NCP12 causes activation of the links and link stations between NCP11 and NCP12 (after the NCP12 load completes). As a result, NCP12 and NCP21 will exchange identification thus allowing cross-domain traffic to flow throughout the network when requested.

The Domain A operator may now start the application program (APPLID=APPL01) and activate the external CDRMs of Domains B and C.

```
V NET,ACT, ID=CDRM02
V NET,ACT, ID=CDRM03
```

The effect of these commands is for the Domain A SSCP/CDRM (CDRM01) to establish sessions with both CDRM02 and CDRM03.

For example, the activation of CDRM02 in Domain A causes an Activate CDRM (ACTCDRM) command to flow to CDRM02 in Domain B. CDRM01 will receive a positive response and send a Start Data Traffic (SDT) command. When CDRM02 responds to the SDT the CDRM-CDRM session is established. As a result, the CDRM01 definition in Domain B, which had been inactive, will become active without any operator command at Domain B.

Once the CDRM to CDRM sessions are established, cross-domain communications may be started.

When APPL01 OPENS its ACB, VTAM will mark the program ACTIVE in the APPLMAJ major node. The actual APPL01-to-resource session establishment will depend on how the program is coded, or whether the cross-domain resources will use a logon procedure.

■■■

Please turn to Mini-Course 16, Exercise 16.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 17

VTAM Traces and Buffer Considerations

Mini-Course 17. VTAM Traces and Buffer Considerations

Introduction to VTAM Traces

A *trace* is a recording of data and/or control information as the information passes a certain point in VTAM. A trace is made during a session between an LU or a terminal and either VTAM's SSCP or a VTAM application program.

The purpose of a trace is to help debug problems with the VTAM-based network. Use a trace when it is necessary to determine exactly what data or SNA commands are flowing back and forth in the network.

There are various types of traces used for debugging different kinds of problems in the network. To assist in understanding when each type of trace might be used, it is important to understand where the different traces are recorded.

Trace Relationships

The following discussion is concerned only with the VTAM traces which are started with the operator MODIFY command or listed as VTAM start options. Figure 17-1 shows that traces are recorded at different places along a VTAM application program to LU session path.

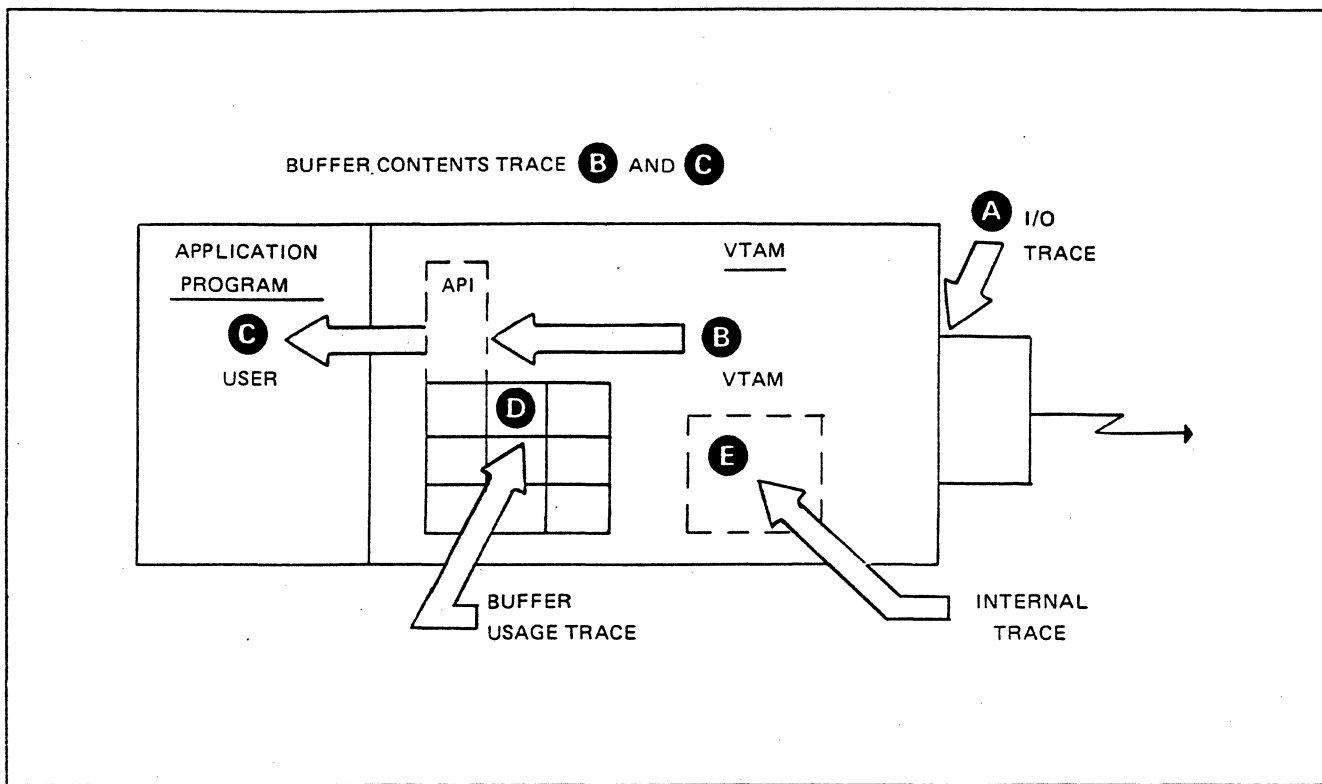


Figure 17-1. Trace Relationships

The following discussion assumes that the buffer contents trace and the I/O trace are "turned on" for a given LU. Assume that a data message comes from a given LU to a VTAM application program, and that a reply to the data message is generated by the application program.

When data is transmitted from the LU to an application program, the first trace point the data passes is the VTAM channel interface (see A in Figure 17-1). The point where the trace occurs is at the path control level of the transmission subsystem. The data is recorded with a VTAM I/O trace record.

Next, the data passes through the VTAM storage pool (buffer pool). The VTAM *buffer contents trace* records the contents of message buffers (up to 256 bytes) as the messages enter VTAM's application program interface (API) routines (see B in Figure 17-1). API routines filter both inbound and outbound information between the application program and other VTAM routines. Before the data is passed from the API to the application program, the VTAM buffer contents are recorded again (see C in Figure 17-1). Thus, the VTAM buffer contents trace captures inbound trace data at two separate points in the data path.

When the VTAM application program sends data to the LU, the process is reversed. When the VTAM application program issues a WRITE or SEND macro, the outbound data is passed across the API to VTAM, and the VTAM buffer contents traces are recorded.

The VTAM *buffer usage trace* (see D in Figure 17-1) records activity within the user-defined VTAM buffer pools. For example, the trace records help to determine if a particular buffer pool is short of buffers.

The VTAM *internal trace* records activity of various user-selected VTAM components in an internal wraparound table or on an external device (see E in Figure 17-1).

Taking VTAM Traces in VM

Taking traces in the VM environment is different than in MVS or VSE. VM CP and GCS commands are issued in addition to the VTAM MODIFY (F) command to start and stop VTAM tracing. The VTAM CP command CPTRAP is used to route the trace records, identify VTAM, and identify which group control system group (GCS group) VTAM resides in.

Rather than a trace data set such as the MVS GTF data set or the VSE TRFILE data set, VM sends trace records to a VM user's reader spool file. The records are later formatted with the TRAPRED facility.

For example, the following commands would be issued to start a VTAM trace:

CP commands:

```
CPTRAP START TO userid      (send (spool) to userid reader)
CPTRAP 3D                   (identify VTAM; that is, 3D)
CPTRAP GROUPID groupname    (identify GCS group)
```

GCS command:

```
ETRACE GTRACE GROUP
```

VTAM MODIFY TRACE command:

```
VTAM F TRACE,TYPE=tracetype,ID=node name (VM command format)
```

This compares to MVS where GTF is started to capture the trace records in the GTF data set; followed by the MODIFY command(s).

At this point, any PIUs from or to "node name" will be captured. After the desired activity for "node name" is completed, the traces may be turned off in VTAM and VM.

For example, the commands used to end tracing are:

VTAM MODIFY NOTRACE command:

```
VTAM F NOTRACE,TYPE=same as started, ID=same as started
```

GCS command:

```
ETRACE GTRACE OFF
```

CP command:

```
CPTRAP STOP
```

Formatting the trace records with TRAPRED will be discussed later in this mini-course.

VTAM Buffer Contents Trace

The buffer contents trace shows the contents of message buffers as the messages enter and leave VTAM. The trace can, therefore, be used to identify any changes made to the messages as they pass through VTAM's buffers and the API.

The buffer contents trace cannot be used to tell the difference between a VTAM application program problem and a problem internal to VTAM. However, it can confirm the order in which data is passed between an application program and a terminal. It can also record up to 256 bytes of data passing to and from an application program.

The buffer contents trace can be started with the MODIFY command or can be defined in a start option. To start a buffer contents trace for the minor node LU023767, the command would be:

```
F NET,TRACE,TYPE=BUF, ID=LU023767
```

VTAM buffer contents trace records have the following general format shown in Figure 17-2.

BUF	aaaaaaaa/bbbbbbbb	yy.ddd/hh:mmtt	LRC(xxx,yyy)	ddddddd
VTAM	TH=hhhhhhhhhhhhhhhhhh	RH=hhhhh		
	nnnnnnnn nnnnnnnn nnnnnnnn nnnnnnnn			cccccccccccccccc
BUF	aaaaaaaa/bbbbbbbb	yy.ddd/hh:mmtt	LRC(xxx,yyy)	ddddddd
USER	nnnnnnnn nnnnnnnn nnnnnnnn nnnnnnnn			cccccccccccccccc

Figure 17-2. VTAM Buffer Contents Trace Format

In Figure 17-2, the fields have the following significance:

BUF	Identifies the buffer contents trace record.
VTAM or USER	Indicates whether the trace is for data out of VTAM's I/O buffers (VTAM) or out of an application program's buffers (USER).
aaaaaaaa/bbbbbbbb	Are the node names of the destination (aaaaaaaa) and the source (bbbbbbbb).
yy.ddd	Is the date when the trace data was recorded (Julian format).
hh:mm:ss.tt	Is the time of day when the trace record was created, where hh is the hours, mm is the minutes, ss is seconds, and tt represents hundredths of seconds.
LRC(xxx,yyy)	Is the number of records lost since the previous first I/O record because of the inability of the trace facility to obtain a VTAM buffer.
xxx	Is the destination's lost record count.
yyy	Is the source's lost record count.
dddddddd	Gives the direction of the message: INBOUND always signifying messages from the network to VTAM application programs and OUTBOUND for messages from the application programs to the network.
TH	Is the transmission header portion of the PIU, expressed in hexadecimal notation. The TH will be 26 bytes for FID4 format.
RH	Is the 3-byte request/response header portion of the PIU, expressed in hexadecimal notation.
nnnnnnnn	Shows the contents of the buffer. Each line contains 32 bytes of user data in standard dump format, that is, expressed first in hexadecimal notation and then in character notation. The cccccccc represents the character notation.

VTAM I/O Trace

The I/O trace can be used to record the order of I/O events that take place between a node and VTAM application programs. It can be used, therefore, to check whether an application program receives all the replies that it should and whether all the commands issued by the application program reach VTAM's path control layer of the transmission subsystems component (TSC).

The I/O trace records activity for the following types of nodes:

- Application Program
- NCP
- PU
- LU
- Non-SNA Terminal
- NCP Line Group
- NCP Communications Line
- CA Communications Line
- Cross-Domain Resource
- Cross-Domain Resource Manager
- A Major Node

The I/O trace can be started by the operator or via a start option. The operator command to start an I/O trace for the terminal node LU033767 would be:

```
F NET,TRACE,TYPE=IO,ID=LU033767
```

I/O trace records have the general format shown in Figure 17-3.

IO	aaaaaaaa/bbbbbbbb	yy.ddd/hh.mm.ss.tt	LRC(xxx,yyy)	INBOUND
	TH=hhhhhhhhhhhhhhhhhh	RH=hhhhhh	RU=hhhhhhhhhhhhhhhh	
IO	aaaaaaaa/bbbbbbbb	yy.ddd/hh.mm.ss.tt	LRC(xxx,yyy)	OUTBOUND
	TH=hhhhhhhhhhhhhhhhhh	RH=hhhhhh	RU=hhhhhhhhhhhhhhhh	

Figure 17-3. VTAM I/O Trace Format

All of the fields in the I/O trace record have the same meaning as described earlier for the buffer contents trace record. (The TH field shown will be 26 bytes long for FID4s.)

The one difference is in the RU field. The RU field contains 7 bytes of basic information if the I/O record is for a non-SNA device, or up to 7 bytes of RU information if the I/O trace is for an SNA device. If bit 5 of the first RH byte is on, the first four bytes of the RU should be sense data.

VTAM Trace Example

Figure 17-4 shows a VTAM trace listing for an inquiry from the SNA terminal LU033767, followed by the reply from the VTAM application program INQUIRE. In this example, both the I/O trace and the buffer contents trace were started for the LU.

The operator commands to start the sample trace are the same as described in the I/O trace and buffer contents trace sections. They are:

```
F NET,TRACE,TYPE=IO,ID=LU033767
      and
F NET,TRACE,TYPE=BUF,ID=LU033767
```

In MVS, the generalized trace facility (GTF) is used to capture and later print the trace records.

There is also a trace analysis program (TAP) available to print a formatted listing of the trace records. TAP is only available in VM if VM/SSP is installed.

In VSE, the trace is printed with the VTAM TPRINT utility and invoked with the command:

```
F NET,TPRINT
```

The reply to the TPRINT utility prompt is PRINT. The TPRINT utility will be discussed later in this mini-course.

The numbers in the sample trace shown in Figure 17-4 correspond to the descriptions that follow the figure.

```

1 IO          /LU023767 79.330/19:60:15.03 LRC(000,000) INBOUND
    TH=40000000 20008006 00000001 00000005 1C00000E 00200001 0009 RH=033080 RU=F3F800000000

2 BUF INQUIRE /LU023767 79.330/19:60:15.08 LRC(000,000) INBOUND
  2 VTAM TH=40000000 20008006 00000001 00000005 1C00000E 00200001 0009 RH=033080
    F3F80000 0000 38....

3 BUF INQUIRE /LU023767 79.330/19:60:15.08 LRC(000,000) INBOUND
  USER F3F80000 0000 38....

4 BUF LU023767/INQUIRE 79.330/19:60:15.09 LRC(000,000) OUTBOUND
  USER F3F8C1D7 D7D3C540 D7C9C540 40404040 4040 38APPLE PIE

5 BUF LU023767/INQUIRE 79.330/19:60:15.09 LRC(000,000) OUTBOUND
  VTAM TH=40000000 00000000 00000005 00000001 1C000020 000E0001 0023 RH=308040
    F3F8C1D7 D7D3C540 D7C9C540 40404040 4040 38APPLE PIE

6 IO          LU023767/ 79.330/19:60:15.10 LRC(000,000) OUTBOUND
    TH=40000000 00000000 00000005 00000001 1C000020 000E0001 0023 RH=308040 RU=F3F8C1D7D7D3C5

7 IO          LU023767/ 79.330/19:60:15.10 LRC(000,000) INBOUND
    TH=40000000 20000007 00000001 00000005 1C00000E 00200000 0003 RH=838000

8 BUF INQUIRE /LU023767 79.330/19:60:15.36 LRC(000,000) INBOUND
  VTAM TH=40000000 20000007 00000001 00000005 1C00000E 00200000 0003 RH=838000

```

Figure 17-4. Sample VTAM BUF and I/O Trace

1. The I/O trace shows the incoming inquiry being passed to VTAM. The request shows (in hexadecimal notation) X'F3F8' for item 38. Note that the destination field is not filled in yet because the PIU has not reached a point where the destination address field (DAF) has been resolved.
2. The buffer contents trace is taken internally within VTAM, shown by the BUF VTAM notation. The data from the RU is shown both in hexadecimal and character notation.
3. This trace record represents the data as it appears just before VTAM passes it to the application program. The BUF USER indicates the buffer contents trace taken at the user application program interface (API). The transmission header (TH) and request header (RH) have been removed from the trace record; only the data is passed to the VTAM application program.
4. After the application program has prepared the reply to the inquiry, it issues the SEND macro to return the reply to the LU. The OUTBOUND trace record shows the text as it was passed from the application program to VTAM. Item 38 turned out to be APPLE PIE.
5. The outbound data is recorded internally within VTAM by another buffer contents trace record (BUF VTAM). The TH and the RH have been added by the transmission subsystem component (TSC) for transmission.

6. The I/O trace record shows the message just before it is passed to the network. Note that all user data is not recorded in the I/O trace record. Only the first 7 bytes of the RU are recorded in an I/O trace record. For this reason, buffer contents traces (TYPE=BUF) should be used if all of the data is needed.
7. The response to the outbound data is received by VTAM. If the leftmost bit of the RH is a 0, the RU is a request. If the leftmost bit is a 1, the RU is a response. In this trace record it is a response since the leftmost bit of the RH is a 1 (X'80'). This is the first response in this example. The response means that when the VTAM application program sent the answer to the inquiry, the program also requested a response. Interpretation of the RH bits in record #6 above show that the program did request a response.
8. The last trace record, BUF VTAM, shows the inbound response being received by VTAM.

VTAM Internal Trace

The VTAM internal trace is substantially different from the other VTAM traces. The purpose of the internal trace is to help the experienced VTAM systems programmer diagnose a possible failure in VTAM itself.

The internal trace is started by the operator command:

```
F NET,TRACE,TYPE=VTAM
```

or as a start option

```
TRACE,TYPE=VTAM
```

In addition, certain trace options may be specified in the command to indicate which internal functions are to be traced.

The internal trace will show various processes of VTAM according to a selection list. One or more of the following process options may be selected:

API	-	Application Program Interface
CIO	-	Channel input/output
PSS	-	Process Scheduling Services
LOCK	-	Locking and Unlocking
SMS	-	Storage Management Services
PIU	-	Path information unit flow
MSG	-	TP Messages
SSCP	-	Request/Response posting
ALL	-	All of the above options

An option is selected by adding the OPTION = parameter to the MODIFY (F) command or the start option. For example, to activate the storage management services (SMS) trace with an internal trace table, the operator keys:

```
F NET,TRACE,TYPE=VTAM,OPTION=SMS
```

To activate both the API and SSCP traces, the operator would key:

```
F NET,TRACE,TYPE=VTAM,OPTION=(API,SSCP)
```

The default internal trace table may be increased by adding a SIZE = parameter on the command, specifying the number of pages for the trace table. The maximum is SIZE = 32. For more extensive traces, the MODE parameter may be used to specify recording of trace records on an external device (MODE = EXT).

To terminate any of the internal traces, the NOTRACE option of the MODIFY command is used. For example, to terminate only the API trace in the above example, the command would be:

```
F NET,NOTRACE,TYPE=VTAM,OPTION=API
```

The storage used by the internal trace table can be freed after all internal traces are completed. To do this, the END option is used as follows:

```
F NET,NOTRACE,TYPE=VTAM,OPTION=END
```

As can be seen from the options, some basic knowledge of VTAM control blocks and data flow is required in order to select the appropriate option(s). Similarly, some knowledge of VTAM internal operation is required in order to interpret the trace records.

The VTAM internal trace is used sparingly, since performance may have some amount of degradation when the internal trace is active due to the resulting increased path length.

■ ■ ■

Please turn to Mini-Course 17, Exercise 17.1, in your PRG and answer the exercise questions.

Buffer Allocation

The VTAM buffer pools have different names and uses between operating systems. The one buffer pool which must receive consideration is the I/O buffer pool. Its name in MVS and VM systems is IOBUF, and in VSE systems the name is LFBUF.

Each buffer pool is managed according to the values supplied at VTAM startup time.

Buffer Pool Specification

Buffer pool values are supplied by operator start options, cataloged start options, or by IBM-supplied default values. The format of the I/O buffer start option is:

MVS and VM IOBUF=(baseno,bufsize,slowpt,,xpanno,xpanpt)

VSE LFBUF=(baseno,bufsize,slowpt,xpanno,xpanpt)

Note the double comma in the MVS format. The omitted value is for fixing pageable pools in storage; the IOBUF pool is a fixed storage pool, by definition. The value (F for fixed) may be coded for other buffer types.

The values supplied for each buffer pool are used by VTAM to allocate buffers in two ways:

- Basic allocation
- Dynamic allocation.

Basic Allocation

Basic allocation is the number of buffers and the length of the buffers that VTAM allocates at startup time. The numbers are either system default numbers for each pool or base numbers supplied in a start option.

The start option values for basic allocation are:

```
poolname=(baseno,bufsize,slowpt,.....)
```

The *poolname* refers to the particular VTAM buffer pool. The *baseno* is the number of buffers allocated to each pool at VTAM startup time, and *bufsize* is the length, in bytes, of each buffer.

The last basic allocation value is the VTAM slowdown point (*slowpt*) and should only be considered for the I/O buffer pool. When the number of available buffers in the pool drops to the value of *slowpt*, VTAM will only accept requests for storage that must be satisfied in order to prevent system interlocking (priority requests). That is, normal requests are rejected or queued until the number of available buffers is again equal to or greater than the value of *slowpt*.

Dynamic Allocation

Dynamic allocation of buffers allows the user to specify that additional buffers be allocated by VTAM during heavy traffic periods. Without dynamic expansion of a pool, basic allocation would have to allow for the peak traffic demand period, resulting in low utilization of storage during normal activity.

Dynamic allocation is specified in a start option or by IBM-supplied default table values. The start option values for dynamic allocation are specified with positional parameters for each pool as follows:

MVS and VM poolname=(...,...,...,xpanno,xpanpt)

VSE poolname=(...,...,...,xpanno,xpanpt)

The expansion number (xpanno) specifies the number of buffers to be added to the pool when the number of available buffers reduces to the value of an expansion point (xpanpt). In normal traffic flow there should be adequate buffers in the base pool (baseno), however, as traffic increases, the number of available buffers will diminish to a point where VTAM can not satisfy buffer requests. With dynamic expansion specified, VTAM will acquire the smallest number of whole pages of storage required to contain the specified (xpanno) number of buffers. For example:

If 5 buffers fit in one page of storage and xpanno is specified as 1, VTAM will acquire one page of storage and expand the pool by 5 buffers.

If 5 buffers fit in one page of storage and xpanno is specified as 6, VTAM will acquire two pages and expand the pool by 10 buffers.

Since the pool goes into slowdown mode when the available number of buffers reaches slowpt, the value for xpanpt must be equal to or greater than the slowpt value. Also, the value of xpanpt must be less than the baseno value.

VTAM will expand the pool repeatedly if the specified expansion point (xpanpt) is reached after prior expansions. Contraction of the pool takes place after an internally calculated number of free buffers is reached.

Dynamic expansion should be used even though there is slightly more processing overhead as compared to having a base pool large enough to handle all traffic volumes. If, for some reason, such as constant data traffic and sufficient storage, the user does not want dynamic allocation, xpanno and xpanpt must be specified as 0 to override the default values.

Buffer Pool Specification Defaults

The values of baseno, bufsize, slowpt, xpanno, and xpanpt may be supplied for each pool from a different source. The overriding hierarchy determines the values that VTAM uses initially to build the buffer pools. Any of the five values may be omitted by placing commas in the appropriate positions. For example, the xpanno for xxBUF may be changed either with a cataloged start option or by the operator with the following:

```
MVS and VM    xxBUF=( , , , , 10)
```

```
VSE          xxBUF=( , , , 10)
```

For any value not given, VTAM uses the following hierarchy:

- A value supplied by the operator at VTAM startup time if operator prompting is used
- A value in ATCSTRyy if the operator uses the LIST = yy option to merge start options with ATCSTR00 options
- A value specified in an ATCSTR00 start option
- If none of the values are specified by the above, VTAM uses a value from an IBM-supplied default buffer pool value table.

Monitoring Buffer Usage

There are guidelines to establish the proper buffer parameters for a particular VTAM system in the *VTAM Customization* manual. The estimates may provide values that produce entirely acceptable performance; however, regardless of performance, the buffer pools should be monitored to determine how each pool is being used in the actual environment.

There are two VTAM traces which provide statistics about the usage of buffers. Both traces provide essentially the same information: one being a recording trace invoked by operator command, and the other being a buffer usage trace record produced after a fixed number of transactions have occurred.

Display Buffer Use

The display buffer use operator command is a convenient method of monitoring buffer use activity. The recording report displays on the operator console and gives an indication of how the buffers are being utilized.

The format of the display command for buffer use is:

```
D NET,BFRUSE
```

A sample of the displayed statistics as it would appear on a VSE network console is shown in Figure 17-5.

```
d net,bfruse
=>
F1 11 5A97I DISPLAY ACCEPTED
F1 11 5D50I VTAM DISPLAY - DOMAIN TYPE= BUFFER POOL DATA
F1 11 5G32I BUFF BUFF CURR CURR MAX MAX TIMES EXP/CONT EXP
F1 11 5G33I ID SIZE TOTAL AVAIL TOTAL USED EXP THRESHOLD INCR
F1 11 5D56I VF 02048 00014P 00009P N/A 00005P N/A N/A N/A
F1 11 5D56I VP 02048 00140P 00067P N/A 00082P N/A N/A N/A
F1 11 5D56I SF 00307 00010 00007 00010 00003 00000 00001/----- 00006
F1 11 5D56I LF 00316 00020 00007 00020 00016 00002 00003/00015 00006
F1 11 5D56I SP 00167 00025 00025 00025 00000 00000 00001/----- 00011
F1 11 5D56I LP 00810 00015 00015 00015 00002 00000 00001/----- 00002
F1 11 5D56I AP 00016 00001 00001 00001 00000 00000 N/A N/A
F1 11 5D56I WP 00144 00010 00010 00010 00000 00000 00001/----- 00013
F1 11 5D56I PP 00016 00001 00001 00001 00000 00000 N/A N/A
F1 11 5D56I NP 00016 00001 00001 00001 00000 00000 N/A N/A
F1 11 5D56I UP 00100 00025 00025 00025 00000 00000 00001/----- 00018
F1 11 5D14I END
```

Figure 17-5. Sample Buffer Use Display

The buffer pool that usually deserves first attention is the I/O buffer pool IOBUF in MVS and VM or LFBUF in VSE. Look at the sample display figures and column headings for LFBUF (LF in the BUFF ID column). If this were an MVS or VM display, the name in the BUFF ID column would be IO and would appear on the first line. The statistics for the I/O buffer in Figure 17-5 are repeated in Figure 17-6 for reference.

Note: The numbers shown in this example are for discussion purposes with all operating systems.

	BUFF ID	BUFF SIZE	CURR TOTAL	CURR AVAIL	MAX TOTAL	MAX USED	TIMES EXP	EXP/CONT THRESHOLD	EXP INCR
MVS or VM	-- IO								
VSE	----- LF	00xxx	00020	00007	00020	00016	00002	00003/00015	00006

Figure 17-6. Sample I/O Buffer Statistics

The following discussion applies to all operating systems.

The I/O pool has been expanded twice (TIMES EXP) by 6 buffers (EXP INCR). Since the current number of buffers in the pool (CURR TOTAL of 20) equals the highest number of buffers used since the last display (MAX TOTAL of 20), VTAM has not contracted the I/O pool. The initial number of buffers supplied at startup time, baseno, must have been 8 (20 minus two expansions of 6).

There are currently 7 available buffers (CURR AVAIL); VTAM will expand the pool again if the number available drops to 3 as shown by EXP THRESHOLD or attempts to contract the pool if the available number reaches 15 (CONT THRESHOLD). The contraction threshold is calculated by VTAM to be $(2 * xpanno) + xpanpt$, in this case, $(2 * 6) + 3$. Contraction occurs only if all of the buffers in a previous expansion block are not in use. In the sample, if CURR AVAIL reaches 15, VTAM will look to see if all 6 buffers of either expansion are free; if so, that block of 6 buffers will be released.

The maximum number of buffers in use at one time was 16 (MAX USED). Note that if the maximum in use had reached 17 there would have been only 3 available buffers left and that would have caused another expansion of 6 buffers.

In a practical situation, several buffer use displays should be taken over a period of operation to monitor the changes in utilization before a decision is made to alter the buffer values.

The most important times to monitor buffer statistics are:

- When VTAM is first installed
- When there is any major change to the network.

At these times, try to monitor buffer usage for the normal traffic period to see if VTAM is spending time expanding and contracting the pool. If so, a larger baseno may be tried if there is sufficient storage. Another alternative would be to increase the value of xpanno to save expansion (and contraction) processing overhead time.

Another time to display buffer use is when terminal response time is lower than expected. VTAM's affect on response time is relatively low compared to other factors; however, available buffers is a factor that can improve response time. Other factors which affect response time are: application program size, number of control units per line, type of logical unit, and, most important, the line capacity. When line utilization reaches 60 per cent and up, response time generally becomes unacceptable. Line utilization is calculated as transaction rate (messages per second) times average message length divided by the line capacity (bytes per second).

If the buffer use display does not provide adequate statistics to adjust buffer pool values, the second type of buffer statistics trace may be of benefit.

Buffer Pool Usage Trace

The buffer pool usage trace provides statistics similar to the buffer use display; however, the trace records are taken at intervals and stored. Where a transaction is defined as one inbound and one outbound message, the interval between records is approximately 1000 transactions. The value may be changed by the user.

The trace is activated by the MODIFY operator command and is used in conjunction with the MVS SMF facility or the VSE SDAID program. The command format to start a buffer pool usage trace, also referred to as an SMS trace, is:

```
F NET,TRACE,TYPE=SMS,ID=VTAMBUF
```

One set of statistics that is not found in the display buffer use is the largest number of requests that VTAM has queued waiting for buffers since the last trace record for the pool was made.

In many VTAM environments the IBM-supplied default buffer values may be sufficient to meet all requirements of network configuration and workload. Use the buffer statistics when there is a suspected problem, or, as mentioned previously, after VTAM installation and after major network changes.

Printing VTAM Trace Records in MVS

In MVS, VTAM trace records are captured via the generalized trace facility (GTF) and printed with the AMDPRDMP print program or ACF/TAP. Refer to the MVS manuals for printing VTAM trace records.

Formatting and Printing Trace Records in VM

The spool reader file created during VTAM tracing contains the trace records ready to be formatted with the TRAPRED program. The TRAPRED program resides on the VM 193 disk after installation of VTAM.

An alternate formatting program is the trace analysis program (TAP) which is supplied with the VM/Systems Support Program (VM/SSP).

An example of formatting with TRAPRED is:

```
ACCESS 193 m      where m is any unused filemode
TRAPRED spoolid  the reader spool file id
3D               identify VTAM
FORMAT          format trace records
PRINTER 999999  direct the output to a printer
QUIT
```

The trace output includes VM-only information in addition to the VTAM trace record fields:

- Time of day (clock) when the record was created
- The length of the trace record including a GTF header
- A formatting module identification
- Information from GTRACE macro instruction's event ID.

Trace Tables, TRFILE, and TPRINT (VSE)

The VTAM print utility TPRINT may be used to print the records of three of the VTAM traces. They are:

- The I/O trace
- The buffer contents trace
- The VTAM internal trace.

The TPRINT utility in VSE may be executed in two ways:

- As a subtask of VTAM and invoked by the domain operator command MODIFY as follows:

```
F NET,TPRINT
```

The operator is then prompted for TPRINT options.

- As a VSE job step in a partition other than the VTAM partition. When the user requires VTAM traces to be taken, there are certain considerations and alternatives regarding where VTAM is to build the trace records. VTAM will build trace records either in main storage or on an external device and in main storage depending on user selection.

The assignment of two VSE files determines the selections possible.

TRFILE

TRFILE is the filename used by VTAM to write external trace records in 2048 byte blocks. TRFILE may be a tape or disk file defined with VSE job control statements and assigned as logical unit SYS001. VTAM will check to see if this file exists when an I/O trace or buffer contents trace is requested. If a VTAM internal trace is requested, the check is only made for TRFILE if the operator specified the MODE=EXT parameter on the MODIFY command when starting the internal trace.

If TRFILE exists, VTAM will write its internal trace table to the file.

SYSLST

The VSE SYSLST file may be assigned to a printer, an unlabeled tape, or a disk file using IJSYSLS as a filename. In order to use the TPRINT utility via the MODIFY command, SYSLST must be assigned in the VTAM partition as part of the VTAM start procedure.

TPRINT in VSE

TRFILE on Disk

TPRINT may be run in a VSE partition when TRFILE is not assigned to disk in the VTAM partition. That is, tracing could be done while VTAM is active and later, after VTAM is stopped, the TRFILE can be assigned to another partition. In this case, TRFILE is assigned as SYS004.

A VSE job step may be set up in the following way:

```
.  
// DLBL   TRFILE,'VTAM.TRACE.FILE',1,SD  
// EXTENT SYS004  
// ASSGN  SYS004,X'cua'  
// ASSGN  SYSLST,X'cua'  
.  
// EXEC   TPRINT
```

TRFILE on Tape

If TRFILE is assigned to an unlabeled tape at VTAM startup, the operator may enter a MODIFY command to start the TPRINT utility. The utility will then prompt the operator for options. An option of NEWTAP specifies that the previous SYS001 tape used by VTAM to record trace records on has been removed, and a new tape has been mounted on the SYS001 drive. The removed tape can now be printed in another partition using the following job control:

```
.  
// ASSGN  SYS004,X'cua'   TAPE  
.  
// EXEC   TPRINT
```

TPRINT via MODIFY Command

TRFILE may be assigned in the VTAM start procedure for external recording of the VTAM trace records and then printed while VTAM is running. TRFILE must be assigned as SYS001.

TPRINT prompts the operator to select either a recorded listing of the VTAM internal trace table for I/O trace and buffer contents trace records or to print records from the external SYS001 device. A recording is not available for the VTAM internal trace.

When the SYS001 file is chosen, TPRINT prompts the operator for options to select all or portions of the file for printing. Some of the options include:

- PRINT - with no additional options, prints all of the trace records
- PRINT BUF=ALL or BUF=nodename(s) - to select buffer contents trace records and print either all or selected node records
- PRINT IO=ALL or IO=nodename(s) - to select I/O trace records to print either all or selected node records
- PRINT INTERVAL=begin time or (begin time,end time) - to select trace records for a particular time period
- CLEAR=YES or NO - whether or not to have VTAM resume trace recording after printing is completed

YES tells VTAM to overlay the trace file from the beginning

NO tells VTAM to resume recording after the last trace record

Note: Any of the PRINT options may be given in combination. For example, the user may select all buffer contents trace records and selected I/O trace records via the following:

```
PRINT IO=nodename,nodename,BUF=ALL
```

To stop TPRINT, the reply to prompting is CANCEL.

If recording is on tape, the tape may be unloaded and a new tape mounted. A reply to prompting of NEWTAP tells VTAM to record on the new SYS001 tape. The removed tape may be printed later or assigned as SYS004 in a DMVSE partition for printing.

VTAM Start Procedure in VSE

If tracing is to be done on an external disk file, the file must be defined in the start procedure for VTAM as follows:

```
.  
// DLBL TRFILE,VTAM.TRACE.FILE  
// EXTENT SYS001  
// ASSGN SYS001,Xcua  
.
```

For an external trace file on tape, SYS001 would be assigned to a tape drive in the start procedure. An unlabeled tape would be mounted on the drive before starting a VTAM trace.

If printing is to be done directly from the VTAM partition, SYSLST must be assigned in the start procedure to a printer, tape, or disk file with a filename of IJSYSLS.

■ ■ ■

Please turn to Mini-Course 17, Exercise 17.2, in your PRG and answer the exercise questions.

VTAM Installation and Coding

Mini-Course 18

VTAM Tailoring with User Exits

Mini-Course 18. VTAM Tailoring with User Exits

Virtual Route Selection Exit

The purpose of the virtual route (VR) selection exit is to have VTAM use the exit to select dynamically at session-establishment time the virtual route number (VR#) and the transmission priority number (TP#) for a requested user session. If this dynamic facility is not required, you will not need to code a VR selection exit.

Determining the Need for a Virtual Route Selection Exit

Before a choice can be made whether an optional user-coded virtual route selection exit should be coded, information must be available on the following:

- The existing virtual routes in the network
- How the available routes are used by user sessions.

This exit allows the opportunity, at the system programming level, to dynamically select the virtual route and, optionally, the transmission priority, to be used by data flowing in a user LU-LU session.

If a user class of service (COS) table has been carefully coded, a VR selection exit may not be required. The COS table entries allow identification, at system definition time, of which virtual routes (and which corresponding transmission priorities) are to be considered when VTAM establishes a particular session.

When analysis of route selection shows that it would be advantageous to change the selection list order based on such things as LU type, time of day, or link availability, a VR selection exit may be coded.

How the VR Selection Exit Functions

The VR selection exit must be reentrant, must be named ISTEVCVR, and must be link-edited into a phase library in VSE, or into the VTAMLIB data set in MVS or in the VTAMUSER LOADLIB IN VM. VTAM looks for ISTEVCVR at startup time. If the link-edited module is found, VTAM drives the VR selection exit for all user session requests.

For a given user session setup, VTAM presents the exit with a set of virtual route numbers (VR#s) and corresponding transmission priority numbers (TP#s). Each VR#-TP# combination represents a pair coded within the COS entry selected for the requested session. However, the only VR#-TP# pairs from that COS entry passed to the exit are those for which the corresponding VR is operative between the origin and the destination subareas.

The exit can reorder the possible VR#-TP# pairs, to indicate a preferred order of VR#-TP# pairs. The exit can also specify that VTAM is not to consider one or more VR#-TP# pairs in selecting the virtual route. (This approach might be taken to balance the number of sessions on a set of virtual routes.) Also, the exit may change any of the virtual route numbers and transmission priority numbers that were presented by VTAM.

When the exit returns to VTAM, VTAM attempts to activate a VR to the specified destination subarea in the exit-specified order. Session traffic will be placed on the first such VR to be activated by VTAM.

Exit Parameters

Register usage within the exit is standard:

- Register 15 contains the address of the exit's entry point.
- Register 14 contains the return address.
- Register 13 contains the address of the calling program's 18-fullword save area.
- Register 1 contains the address of the exit parameter list.

The exit parameter list is shown in Figure 18-1.

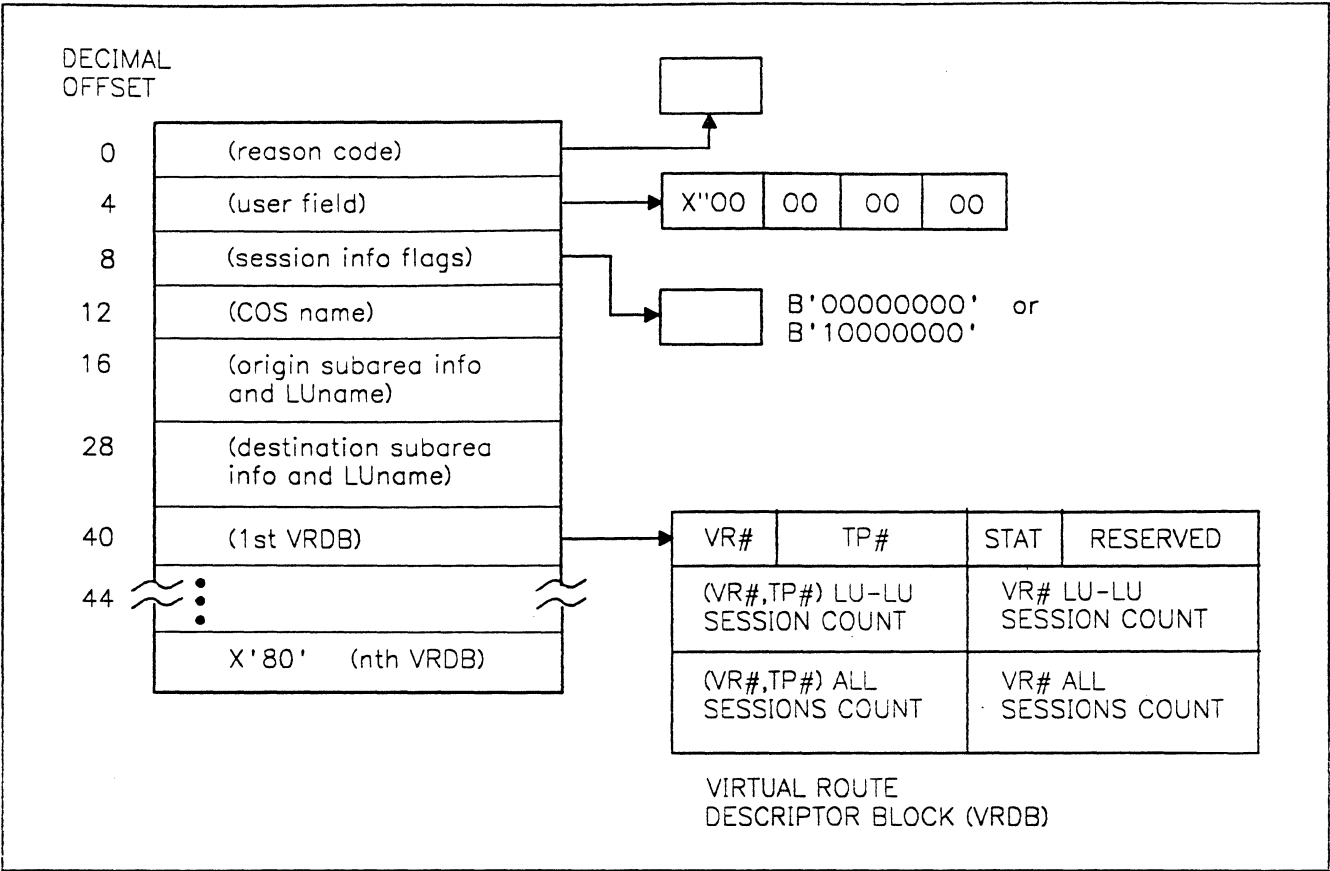


Figure 18-1. VR Selection Exit Parameter List

Word 1

Word 1 of the parameter list points to a 1-byte field reason code indicating the reason the exit was invoked, as follows:

X'00' The exit is being driven for the first time since VTAM was initialized. Any unique user processing (such as opening of data sets) that will be performed on a one-time basis should be executed this time, in addition to normal exit processing.

X'01' The exit is being driven for a session request; however, this is not the first time the exit has been invoked.

X'02' In MVS only, the exit is being driven after a previous abend (abnormal end) of this exit. Of course, an exit shouldn't abend. But, especially during the testing phase, it's possible that user exit code is faulty. If a previous invocation of the exit resulted in an abend, user processing may be required. For example, the exit may need to clean-up control blocks in use or re-open user data sets that had been in use by the exit at the time of the abend.

X'03' The exit is being driven for VTAM termination. Any end-of-job user processing (such as closing of data sets) should be performed at this time, as the exit will not be invoked again.

X'04' In MVS only, the exit is being driven for VTAM termination, after the exit itself has previously abended. This code can be used when testing the exit since VTAM will continue to drive the exit until it abends approximately four times in four minutes (the abend threshold). In this case, VTAM will use the original default list given to the exit and not drive the exit again until VTAM itself terminates.

Word 2

Word 2 contains the address of a 4-byte user field. This field is binary zeros when the exit is invoked for the first time (Reason code = X'00'). The exit can place any 4 bytes of data in the field. Subsequently, when the exit is scheduled, this data will be returned to the user. In effect, VTAM saves the 4 bytes and passes them back to the exit for the next time it is invoked. The exit can use this field for any purpose. For example, the address of a dynamically obtained storage area may be stored here, so that subsequent exit invocations can access that same storage area.

Notice that when the reason code pointed to by Word 1 specifies an X'03' or X'04' (VTAM is terminating), the leftmost bit of Word 2 is set on, indicating the end of the parameter list. In this unique case, the parameter list consists only of Word 1 and Word 2.

Word 3

Word 3 contains the address of a 1-byte field containing session information flags. Currently, only Bit 0 (the leftmost bit) of the flag byte is used. The other 7 bits are reserved.

If Bit 0 is a 1-bit, the pending session requires a virtual route that is mapped to ERO. This requirement occurs only if the route is a migration route. A migration route includes a VTAM or NCP that is at a Version 1 Release 2, or earlier, level.

If Bit 0 is a 0-bit, the above requirement does not exist.

Word 4

Word 4 contains the address of the 8-byte COS name which is the selected COS entry associated with the pending logon request. A name with less than 8 bytes is padded on the right with blanks as necessary to make 8 bytes.

Word 5

word 5 is a field 12 bytes long and contains a 4-byte subarea number of the host node in which the VR selection exit is being driven. This is the node that sends the BIND command. The subarea number is followed by an 8-byte LU name (EBCDIC).

Word 6

Word 6 is a field 12 bytes long and contains a 4-byte subarea number of the destination node. The subarea number is that of the secondary LU, to which the BIND command will be sent. The subarea number is followed by the 8-byte secondary LU name.

Word 7 (and subsequent words of the parameter list) contains the address of an 12-byte field called a virtual route descriptor block (VRDB). The parameter list is variable in length. Its length depends upon the number of entries in the COS entry named in Word 4. The last word in the parameter list contains a 1-bit in the high-order position to indicate the end of the list.

Figure 18-2 shows the format of the virtual route descriptor block (VRDB).

VR#	TP#	STAT RESERVED
(VR#,TP#) LU-LU Session Count		VR# LU-LU Session Count
(VR#,TP#) All Sessions Count		VR# All Sessions Count

Figure 18-2. Format of Virtual Route Descriptor Block (VRDB)

Byte 0 of each VRDB contains a virtual route (VR) number (0-7) from the COS table entry being used for the requested session. Byte 1 contains the transmission priority (0-2) associated with that VR number. Byte 2 contains the status of that virtual route—active or inactive. The status byte is X'01' if the corresponding VR is not active, X'02' if the corresponding VR is active. All other values are reserved.

Remember that only operative virtual routes are passed to the VR selection exit. But, just because a VR is operative doesn't necessarily mean that it is also active. A VR is considered operative if none of the components (such as links and NCPs) are marked as failed. A VR is active only if a session already exists using that VR number to the specified destination subarea number.

Byte 3 of each VRDB is reserved.

Bytes 4 and 5 of a VRDB contain a halfword binary number that is the count of the number of LU-LU sessions between the origin and destination subareas that are currently assigned to the specified virtual route (VR#-TP# pair). That is, the sessions all use the specified virtual route number and the indicated priority.

Bytes 6 and 7 contain a halfword binary number that is the count of the total number of LU-LU sessions between the origin and destination subareas that are currently assigned to the specified virtual route number, regardless of the transmission priority.

Bytes 8 and 9 contain a halfword binary number that is the count of all sessions using the virtual route (VR#-TP# pair).

Bytes 10 and 11 contain the total number of all sessions using the virtual route number, regardless of the transmission priority.

Reordering the VR#-TP# Pairs

As noted earlier, the exit can reorder the VR#-TP# pairs to be used. This reordering is accomplished by switching the addresses of the VRDBs in the parameter list.

Changing VR#-TP# Pairs

The exit can also alter the VR numbers and/or transmission priority numbers within each VRDB. This action is necessary if the exit decides to specify a VR#-TP# pair that is not contained in the original parameter list—that is, a VR#-TP# pair that is not included in the selected COS entry.

Shortening the VRDB List

The exit can shorten the list of VRDBs by setting to X'80' the leftmost byte of the fullword address of the last VRDB you want VTAM to consider. Your exit code, however, cannot lengthen the list of VRDBs.

Returning to VTAM

After altering the parameter list and/or the VRDBs, the exit must set a zero return code in register 15 and return control to the address contained in register 14. (Failure to set a zero return code in register 15 causes VTAM to mark the exit as unusable and to notify the network operator that route selection has been disabled.)

VR Selection Exit Considerations

VTAM will not invoke the VR selection exit under four circumstances:

- VTAM chooses virtual routes for SSCP sessions without using the user exit.
- If the session origin and destination subareas are identical; that is, if the session is application-to-application or application-to-local node within the host, virtual route selection is not required.
- If the origin or destination node does not support VRs and ERs (a migration node), virtual route zero (VR0) is automatically selected for the session traffic, and the exit is not invoked.
- There is no defined or operative virtual route in the COS list.

■ ■ ■

Please turn to Mini-Course 18, Exercise 18.1, in your PRG and answer the exercise questions.

Session Management Exit Routine

Whenever a session between logical units is being established or terminated, VTAM calls the session management exit routine. If more than one VTAM is in the session-setup path, each VTAM will invoke the exit.

The session management exit routine allows a user to code the exit routine for the following four functions:

- Restrict a logical unit from a particular session (authorization function).
- Account for the time logical units are in session (accounting function).
- For MVS cross-network sessions, select alternate gateway NCPs.
- Select adjacent SSCPs in the session-setup path.

If a session management exit routine is not provided, VTAM will allow all sessions, discard accounting data saved for the routine, select a gateway path (MVS and VM only) from the list of generated GWPATH definition statements, and select an adjacent SSCP from the default list.

When VTAM passes control to the session management exit routine, it sets the registers as follows:

Register 1	Address of a variable-length list of virtual storage addresses. Each address in the list points to parameters for one of eight possible function parameter lists.
Register 13	Address of the register save area (18 fullwords)
Register 14	Return address
Register 15	Entry address of the routine

The final register contents vary depending on the function. For example, register 15, the normal return code register, must be set to zero for some functions and set to defined return codes with other functions.

Session Management Functions

There are seven possible session management functions. The function codes and description are as follows:

- **Begin (X'FE')**

A required function that tells VTAM which functions are included in the routine. The begin function is called only once to select the other functions.

- **Initial Authorization (X'00')**

Allow or disallow the session during LU-LU session initiation. If all of the session information is not known at this point, such as with cross-network session setup, this function may defer the request and set a return code to have the Secondary Authorization function called to make the decision.

- **Secondary Authorization (X'01')**

Receives more information than the Initial Authorization function. For example, the destination logical unit (DLU) real name, network identifier, and owning SSCP name may not be known at initial authorization time.

- **Initial and Final Accounting (X'02' and X'03')**

Called at session initiation and session termination to pass accounting information such as session partner information and time of day.

- **Gateway Path Selection (X'04') (MVS only)**

Passes the gateway path selection list for shortening or reordering. The original list passed by VTAM is in the order of coded GWPATH definition statements.

- **Adjacent SSCP Selection (X'06')**

Passes the default adjacent SSCP list for reordering to take effect on the next LU-LU session setup. The function results in one of two choices: proceed with session setup using a modified SSCP Name List, or proceed with session setup with standard VTAM routing.

- **End Function (X'FF')**

Called during VTAM termination to allow any required cleanup to be performed.

Sample Session Management Exit Routine

Refer to Figure 18-3 for the following discussion.

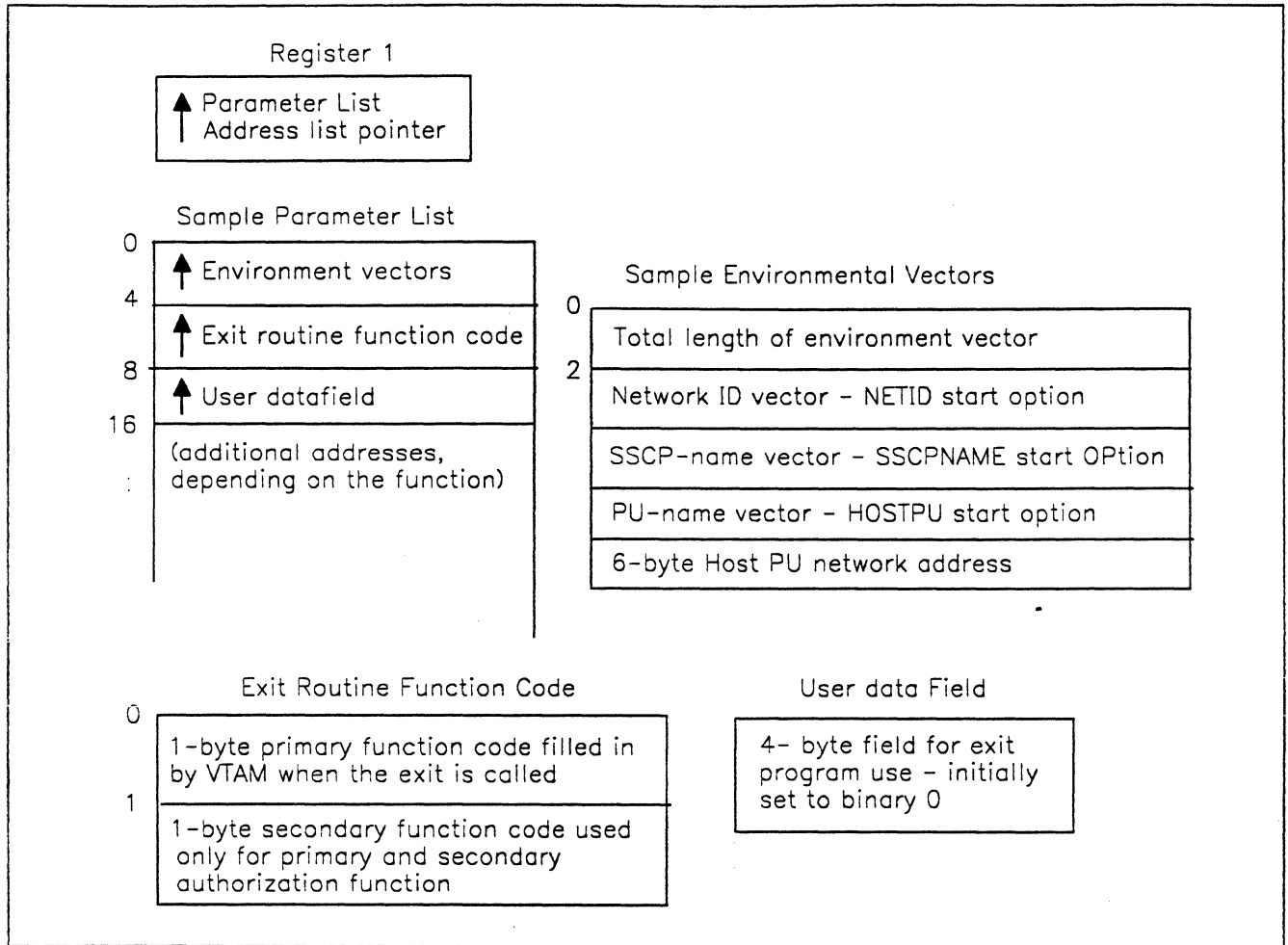


Figure 18-3. Sample Session Management Exit Routine - Passed Information

When VTAM calls the exit it places the exit's entry point address in register 15, and a parameter list address in register 1, and the primary function code in the first byte of the exit routine function field. The user code would access the second word in the parameter list for the function code field address; then compare against the primary code to determine which function was called. Each of the different functions has its own format for the parameter list.

VTAM passes information to the exit through information vectors or information lists. The addresses for each of the vectors or lists would be found in the function parameter list. User code can read the addresses in the list into registers to use for offsets in the function vectors or lists. At the end of function processing, the exit would restore saved registers and branch on register 14, the return register.

The name of the session management exit routine must be ISTEEXCAA and be link-edited to VTAMLIB in MVS or the definition library in VSE or in the VM VTAMUSER LOADLIB.

Things to avoid in the routine are:

- VTAM macro instructions since the application program interface (API) is not used.
- Any function, such as I/O, which would cause system waits to occur.

The routine operates in VTAM's pageable storage as an internal subroutine and it must be reentrant code.

System Exits in VM/SP Release 4

With ACF/VTAM Version 3 Release 1.1, VTAM functions for VM are compatible with those for MVS including exit routine coding.

Prior to VTAM V3.1.1, both the system authorization exit and the system accounting exit must be present in a VTAM V3 for V system. Without them, VTAM is unable to start successfully. Because of this requirement, IBM provides default system exits with each VTAM V3 for VM systems. Depending on user requirements, exits may be coded to replace those furnished by IBM.

These exits are still available for VSE and MVS with VTAM Versions 2 and 3. However, it is recommended that authorization and accounting requirements be placed in the Session Management Exit Routine.

Authorization Exit Function

The purpose of the authorization exit is to allow a user to permit or to refuse connection requests between an application program and an LU or terminal. The IBM-supplied authorization exit authorizes all connection/disconnection requests.

The systems programmer normally codes and maintains the authorization exit. If there is a reason to control or limit connection or disconnection requests on a system basis, an authorization exit should be written.

Assume that a user wants to restrict which terminals are allowed to talk to the VTAM application program named PAYROLL. For example, only the LUs named LU053767 and LU063767 are to have access to PAYROLL. The system authorization exit can be coded in such a way as to permit only those two terminals to connect to PAYROLL. All other connection requests for PAYROLL can be denied by the exit.

There is only one VTAM authorization exit. The exit is invoked for each LU-LU connection or disconnection request.

Coding the Authorization Exit

The user analysis part of the authorization exit is user-dependent. Generally, the user includes a series of tables, listing for each program the nodes which are authorized to connect to it, or perhaps, the nodes not authorized to connect to it. The logic part of the exit code scans through the tables and determines if the particular connection is allowed or not; then sets R15 accordingly.

Figure 18-4 shows one approach to handling an authorization exit.

POSSIBLE HANDLING OF AUTHORIZATION EXIT			
AUTH_TBL	DS	OF	ALIGN
PROG1	DC	CL8'PROG1'	APPL NAME
	DC	A(PROG2)	NEXT ENTRY
	DC	F'3'	NUMBER TERMINALS
	DC	CL8'TERM6'	TERMINALS
	DC	CL8'TERM19'	AUTHORIZED
	DC	CL8'TERM137'	
PROG2	DC	CL8'PROG2'	APPL NAME
	DC	A(PROG3)	NEXT ENTRY
		etc.	

Figure 18-4. Authorization Exit Sample Approach

In this example, each program is listed, followed by the number and names of the terminals and LUs authorized to connect to the program. In the example, program PROG1 has three terminals authorized to connect to it: TERM6, TERM19, and TERM137. All other terminals/LUs are prohibited from connection to PROG1. The logic of the exit (not shown in Figure 18-4) contains the assembler code to scan through the table and determine whether the connection is authorized.

Use of the Authorization Exit

An authorization exit may not be needed because every application program can determine valid connection requests for itself. For example, if the program is a payroll program and only certain terminals are to connect to it, then logic can be written in that program's logon exit to reject logon requests from all other terminals. If such logic is included in the program's logon exit, an authorization exit is unnecessary to keep LUs from logging on to that program.

One use of the authorization exit is to keep a particular program from acquiring specific LUs. However, in a well-run data processing installation it is unlikely that programs will attempt to acquire LUs they are not supposed to acquire.

Another possible use of the authorization exit is to prevent specific connections from occurring at certain times of the day. For example, TERM1 may be allowed to connect to certain VTAM application programs only after 5 P.M. This limitation can be enforced by coding those VTAM application programs to disallow connection requests from TERM1 prior to 5 P.M. However, it may be more convenient to enforce the limitation by coding a user authorization exit. In that way, repetitive coding can be avoided in a number of application programs.

The authorization exit is a systems task; when the exit is executing, no other VTAM system task can execute. This fact means that lengthy input/output operations or WAITs should be avoided in the exit since these operations tie up the entire VTAM system.

Accounting Exit - Function

The accounting exit provides a way for the user to do installation-determined accounting. As mentioned earlier, IBM furnishes an accounting exit with VM VTAM systems. This IBM-supplied exit is a "dummy" exit performing no accounting function but simply returning control to VTAM.

The accounting exit is invoked whenever a terminal/I.U and a VTAM application program are connected or disconnected.

In the accounting exit, the time of day at connect time and at disconnect time can be recorded. The data thus collected may be written to tape or sequential disk, and the installation can sort and process the output to provide user billing based on connect time.

Other statistical information (such as the number of messages or bytes transmitted or received) is not readily available in a VTAM system.

An experienced VTAM systems programmer can obtain useful information from various VTAM internal control blocks. (The exits operate as a systems task and, as such, give the exit programmer access to information in VTAM control blocks.) For example, the exit can obtain information for local 3270s or local 3790s concerning the number of SIOs issued for that local device.

In general, however, the only information available in the accounting exit is the time-of-day when connection and disconnection occurred. From this information the total connect time for a session can be determined and written to a sequential file.

Coding the Accounting Exit

In the accounting exit, the system programmer will probably use operating system time-of-day macros to obtain the time. The programmer may write a tape or sequential disk record every time there is a connection or disconnection request. This tape can be sorted later (offline), the information converted to connect-time for each session, and users charged accordingly.

The accounting exit is a systems task, and lengthy waits due to input or output requests will tie up the rest of the VTAM system. Thus, WAITS should be avoided in the exit code.

Here is another approach to coding an accounting exit. When a connection is established, accumulate the information in a user-defined control block in the exit. This control block contains information such as terminal name, program name, connect time, and a field in which to insert disconnect time later on. When the session is terminated, add the disconnect time to the control block and write all the information about the session in one record. This approach has one major advantage — it reduces the number of input or output operations performed in the exit since only one record is written for each session, not two (one for connection, one for disconnection). A disadvantage is that if the system should crash while the session is active, information in the control block is lost with the rest of storage. A consideration for accounting exits is: the time it takes to write the records versus the amount of accounting information gathered.

Testing the Exits

After the exit(s) have been coded in assembler language, they must be assembled and link-edited to a VTAM LOADLIB file in VM.

Take care in testing the system exits since the exits must be present in order to start up a VTAM system. If a particular exit being tested is defective, the VTAM system cannot be used until the problem is corrected. If, for example, a program check occurs in the assembler code of an exit, the problem must be corrected before any VTAM connections can be made.

One approach to testing these very sensitive system exits is: use an operating system utility to rename the IBM-provided exit. For example, rename the authorization exit 'OLDAUTH', then assemble the new exit, and link-edit it to VTAM LOADLIB with the required member name:

```
Authorization exit name:  ISTAUCAT
Accounting exit name:    ISTAUCAG
```

Next, test the exit by bringing up VTAM and seeing whether the exit does what it is expected to do. If the exit works, fine! If it doesn't, VTAM must be terminated, the defective exit scratched (or renamed to something else), and the saved IBM exit renamed with the required member name. In that way, the VTAM system can continue running while the problems in the system exit are being debugged.

■ ■ ■

Please turn to Mini-Course 18, Exercise 18.2, in your PRG and answer the exercise questions.

