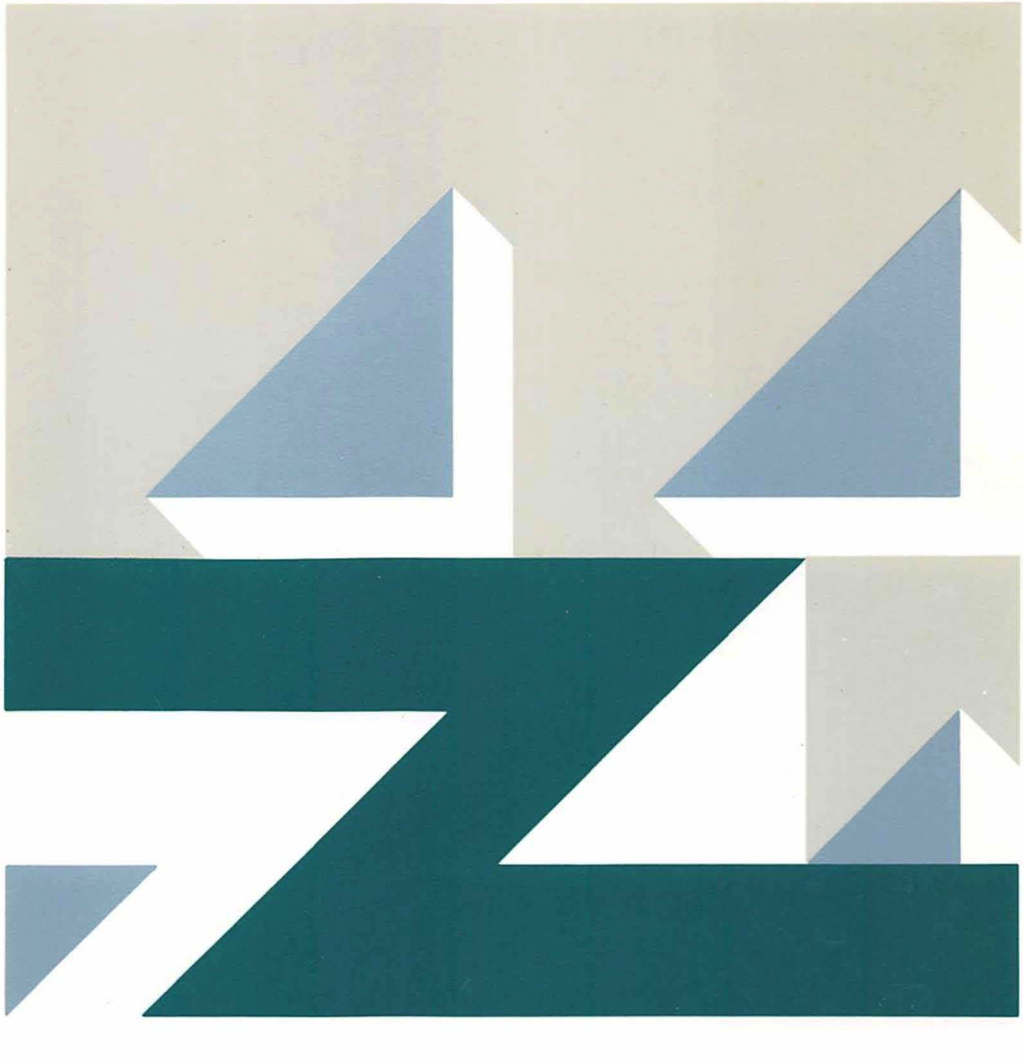
LY43-0046-1

VTAM™

Customization

Version 3 Release 3 for MVS, VM, and VSE/ESA

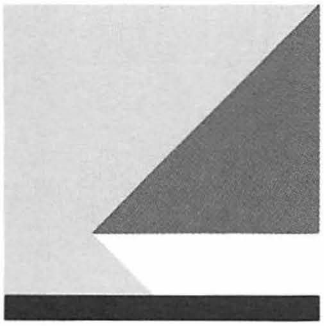


Licensed Materials-Property of IBM
Restricted Materials of IBM
©Copyright IBM Corp. 1984, 1990

VTAM™

Customization

Version 3 Release 3 for MVS, VM, and VSE/ESA



File Number
S370/4300/30XX-50

Program Numbers
5685-085 (MVS/ESA)
5684-052 (VM/9370 for VM/SP)
5684-095 (VM/ESA)
5665-289 (MVS/XA)
5664-280 (VM/SP)
5666-363 (VSE/ESA)

Licensed Materials-Property of IBM
Restricted Materials of IBM
©Copyright IBM Corp. 1984, 1990

Second Edition (September 1990)

This licensed document applies to the following releases of the Advanced Communications Function for VTAM (ACF/VTAM), an IBM licensed program:

ACF/VTAM Version 3 Release 3 for MVS/ESA (program number 5685-085)
ACF/VTAM Version 3 Release 3 for MVS/XA (program number 5665-289)
ACF/VTAM Version 3 Release 3 for VM/SP (program number 5664-280)
ACF/VTAM Version 3 Release 3 for VM/9370 (program number 5684-052)
ACF/VTAM Version 3 Release 3 for VSE/ESA (program number 5666-363)
ACF/VTAM Version 3 Release 3 for VM/ESA (program number 5684-095).

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to IBM Corporation, Department E15, P.O. Box 12195, Research Triangle Park, North Carolina 27709, U.S.A. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1990. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the Agreement for IBM Licensed Programs. Changes are made periodically to the information herein; before you use this document in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program may be used.

This book contains product-sensitive programming interfaces for customers provided by VTAM. Installation exits and other product-sensitive interfaces are provided to allow the customer installation to perform tasks such as product tailoring, monitoring, modification, or diagnosis. They are dependent on the detailed design or implementation of the product. Such interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

IBM may have patents or pending patent applications covering subject matter described in this document. Furnishing of this document does not give you any license to these patents. You can send inquiries, in writing, to the IBM Director of Commercial Relations, International Business Machines Corporation, Purchase, New York, 10577.

Licenses under IBM's utility patents are available on reasonable and nondiscriminatory terms and conditions. IBM does not grant licenses under its appearance design patents. Inquiries relative to licensing should be directed in writing to the IBM Director of Commercial Relations, International Business Machines Corporation, Armonk, New York, 10504.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Some trademarks of the IBM Corporation in the United States and/or other countries appear in this document. These IBM trademarks are:

IBM	VTAM	MVS/XA
MVS/ESA	Enterprise System/9370	NetView

Contents

Chapter 1. Writing VTAM Exit Routines 3
Session Management Exit Routine 3
Virtual Route Selection Exit Routine 48
VR Pacing Window Size Calculation Exit Routine 54
Session Accounting Exit Routine 58
Session Authorization Exit Routine 60
VSE TPRINT Processing Exit Routine 63
Installing VTAM Exit Routines 66

Chapter 2. Defining User Modules and Tables 69
Module for VTAM Constants 69
CNM Routing Table 85
Logon-Interpret Routine Requirements 88
Installing Tables and Modules in VM 89

Chapter 3. Writing VSCS Exit Routines for VM 93
VSCS Data Manipulation Exit Routines 93

Chapter 4. Writing TSO/VTAM Exit Routines for MVS 101

Appendix A. IBM-Supplied CNM Routing Table 111

Appendix B. Sample Session Management Exit Routine 113
Initialization 114
Begin Function 117
Secondary Authorization Function 121
Gateway Path Selection Function for MVS and VM 125
Alias Selection Function for MVS and VM 130
Final Accounting Function 136
End Function 138

Glossary 141

Bibliography 165
VTAM Publications 165
Related Publications 167
Other Related Publications 168

Index 169

© Copyright IBM Corp. 1984, 1990

Figures

1. Sample Cross-Network Session 5
2. Sample Cross-Network Session for Init Other CD 6
3. Session Flow for SLU-Initiated Session 40
4. Session Flow for PLU-Initiated Session 41
5. Session Flow for ILU-Initiated Session 43
6. Session Flow for ILU-Initiated Session 44
7. Session Flow for SLU-Initiated Session with USERVAR name 45
8. Session Flow for DSRLST 47
9. Example of Logic for a TPRINT Exit Routine 65
10. Example of SSCP-Rerouting-Count Use 76
11. IBM-Supplied Skeleton Data Manipulation Exit Routine for VSCS 94
12. Network Environment for Sample Session Management Exit Routine . . . 114
13. User Data Storage Area Format 118
14. Network Identifier (NETID) Registration Table 118
15. Sample Portion of PLU Resource Identification Control Vector 122
16. Gateway Path Selection List for Sample Session Management Exit Routine 129
17. Real and Alias Network-Qualified Names 131
18. Alias Terminal Name Pool - Three Digit Suffix 135
19. Example of a Language Statement 146
20. NCP Examples 146
21. VTAM Examples 146
22. Links and Path Controls 151

Tables

1. The VTAM V3R3 Library xii
2. Example of Session Management Exit Routine Functions 4
3. Summary of Parameter Lists for Session Management Exit Routine (Part 1) 8
4. Summary of Parameter Lists for Session Management Exit Routine (Part 2) 9
5. Begin-Function Parameter List 10
6. Initial-Authorization-Function Parameter List 11
7. Secondary-Authorization-Function Parameter List 12
8. Accounting-Function Parameter List 13
9. Gateway-Path-Selection-Function Parameter List 14
10. Session-Takeover-Accounting-Function Parameter List 15
11. SSCP-Selection-Function Parameter List 16
12. Alias-Selection-Function Parameter List 17
13. End-Function Parameter List 18
14. Environment Vectors 19
15. Function Code and Related Session Information 21
16. Session Management Exit Routine Options 24
17. PLU and SLU Resource Identifier Control Vector 26
18. PLU and SLU Hierarchy Control Vector 27
19. Format of OLU Gateway Information Vector 29
20. Format of Gateway Path Selection List 30
21. Format of SSCP Name List 31

22. Format of DLU Gateway Information Vector 32
23. Format of OLU Adjacent SSCP Vector 33
24. Format of DLU Adjacent SSCP Vector 33
25. Alias-Selection-Input Parameter List 34
26. Constants for the Alias-Selection-Input Parameter List 36
27. Alias-Selection-Output Parameter List 38
28. Parameter List for the Virtual Route Selection Exit Routine 51
29. Format of the Virtual Route Descriptor Block 52
30. VR-Pacing Window-Size-Calculation Exit Routine Parameter List 55
31. Explicit-Route Characteristics Table 58
32. Parameter List for the Authorization Exit Routine 62
33. TPRINT-Processing Parameter List 64
34. MVS and VM Libraries for VTAM Exit Routines 66
35. VTAM Module Names for Exit Routines 66
36. Default IBM Constants Module (ISTRACON) 70
37. Format of CNM Routing-Table Header 86
38. Format of CNM Routing Table 86
39. Parameter List for Keyboard/Printer and TWX-Device Data Manipulation Exit Routines 95
40. Parameter List for Display-Device Data Manipulation Exit Routines 96
41. Summary of the TSO/VTAM Exit Routines 101
42. Routine — Module Cross-Reference 107

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

About This Book

Customization is the process of tailoring VTAM* by enhancing or extending it to suit your needs. *VTAM Customization* is a reference book for programmers responsible for customizing VTAM.

VM VSCS, although it is shipped with VTAM, is an *application program*. Therefore, this book discusses VSCS separately from VTAM.

Note: VM/9370 VTAM can be installed only on the following IBM* Enterprise Systems/9370* models running under a VM/SP operating system:

- 9371
- 9373
- 9375
- 9377.

Who Should Use This Book

Your job title probably includes the word "system programmer" or "analyst." Use this book if your job includes customizing VTAM. Before using this book, you should be familiar with the information in the following publications:

- *Planning and Reference for NetView, NCP and VTAM*
- *VTAM Network Implementation Guide*
- *VTAM Resource Definition Reference.*

How to Use This Book

You can use this book as a reference for customizing VTAM to meet your own needs. It can also be used as background reading to help understand the uses and function of exit routines and VTAM constants. Definitions of technical terms are in the "Glossary" on page 141.

How This Book Is Organized

VTAM Customization consists of four chapters and two appendixes:

- Chapter 1, "Writing VTAM Exit Routines" on page 3, describes how to code and install VTAM exit routines, including:
  - Session management exit routine
  - Virtual route selection exit routine
  - VR pacing window size calculation exit routine
  - Session accounting exit routine
  - Session authorization exit routine
  - VSE TPRINT processing exit routine.

* Trademark of the IBM Corporation. See "Special Notices" on page iii for a list of all IBM trademarks used in this book.

- Chapter 2, "Defining User Modules and Tables" on page 69, describes how to create and modify the communication network management (CNM) table and VTAM modules. It also discusses installing tables and modules in VM, and logon-interpret routine requirements.
- Chapter 3, "Writing VSCS Exit Routines for VM" on page 93, describes how to code and install VSCS data manipulation exit routines.
- Chapter 4, "Writing TSO/VTAM Exit Routines for MVS" on page 101, provides information you will need to write and install exit routines for TSO/VTAM.
- Appendix A, "IBM-Supplied CNM Routing Table" on page 111, lists the IBM-supplied CNM table.
- Appendix B, "Sample Session Management Exit Routine" on page 113, illustrates how the initial session management exit routine environment is established, the type of information available in each parameter list for the specific sample environment, and some techniques used to examine the various parameter lists.

Symbols Used in This Book

The following symbols are used in this book to indicate information that pertains to a specific operating system:

MVS/XA Indicates information that applies to MVS/XA* only. It is not used in this manual, because this manual contains no information that is unique to MVS/XA.

MVS/ESA Indicates information that applies to MVS/ESA * only.

MVS Indicates information that applies to both MVS/XA and MVS/ESA, but not VM or VSE.

VM Indicates information that applies to VM only.

VM/ESA Indicates information that applies to VM/ESA only.

VSE Indicates information that applies to VSE only.

Note: Because VSE/ESA is one of the licensed programs in VSE/System Package, no distinction is made between the two.

ES/9370 Indicates information that applies only to the telecommunications subsystem controller, the token-ring subsystem controller, or the workstation subsystem controller on the Enterprise System/9370 processor supported by the VM or VSE operating system only.

These symbols precede unique information. If a piece of information applies to more than one operating system, but not all of them, more than one symbol can precede the information. For example:

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

The information in the example applies to MVS/XA, MVS/ESA, and VM, but not VSE. Information that applies to all operating systems is **not** denoted by any symbol.

* Trademark of the IBM Corporation. See "Special Notices" on page iii for a list of all IBM trademarks used in this book.

What Is New in This Book

Version 3 Release 3 for VM/ESA

VTAM Version 3 Release 3 for VM/ESA uses System 370/XA architecture to take advantage of 31-bit addressable storage. VTAM uses 31-bit addressable storage as much as possible to provide more space (beyond 24-bit addressable storage) for VTAM and customer application programs. This reduces the demand for 24-bit addressable storage and eases storage constraints on the members of a Group Control System (GCS) group. See "Installing VTAM Exit Routines" on page 66 and Chapter 2, "Defining User Modules and Tables" on page 69 for more information on VM/ESA.

Session Management Exit Enhancements

MVS VM Alias Selection Function

In V3R3, you can use the session management exit routine to translate session information. You can use this function to replace or complement NetView's alias application. See pages 17 and 33 for more information on this function.

Initial Authorization Function Enhancement

This function allows network providers to authorize new sessions that are requested by application programs passing sessions from one partner to another using the CLSDST OPTCD=PASS macroinstruction. See page 10 for more information on this function.

Dynamic Adjacent SSCP Tables

You no longer need to code adjacent SSCP tables to establish cross-domain or cross-network sessions. You can now use the DYNASSCP start option and VTAM will route session establishment requests dynamically. See page 15 for more information on this function.

Other Changes, Additions, and Deletions

Constants

MVS/ESA Three new constants are added to the constants module (ISTRACON):

RACSAWLM	SAW data space limit factor (page 84)
RACSAWPK	SAW data space packing factor (page 84)
RACPIULM	PIU data space limit factor (page 84).

VSE The constant RACCPS for X.21 CPS retry is removed from the constants module.

Sample Session Management Exit Routine

Appendix B, "Sample Session Management Exit Routine" on page 113, illustrates how the initial session management exit routine environment is established, the type of information available in each parameter list for the specific sample environment, and some techniques used to examine the various parameter lists.

Where to Find More Information

Table 1 shows the books in the VTAM V3R3 library, arranged according to related tasks. For a description of these manuals, see "Bibliography" on page 165. The bibliography also lists the titles and order numbers of manuals related to this manual or cited by name in this manual.

Table 1. The VTAM V3R3 Library

Planning	
SC31-6092	*Planning and Reference for NetView, NCP, and VTAM*
GC31-6429	*VTAM Directory of Migration Information*
SK2T-2010	*VTAM Storage Estimates* (diskette only)
Installation and Resource Definition	
SC31-6404	*VTAM Network Implementation Guide*
SC31-6412	*VTAM Resource Definition Reference*
Customization	
LY43-0046	*VTAM Customization*
Operation	
SC31-6408	*VTAM Operation*
SC31-6405	*VTAM Messages and Codes*
Writing Application Programs	
SC31-6409	*VTAM Programming*
SC31-6410	*VTAM Programming for LU 6.2*
Diagnosis	
LY43-0042	*VTAM Diagnosis*
LY43-0053 (V3R3)	*VTAM Data Areas for VSE*
LY43-0043 (V3R3)	*VTAM Data Areas for MVS*
LY43-0045 (V3R3)	*VTAM Data Areas for VM*
Reference	
LY43-0047	*VTAM Reference Summary*
GC31-6430	*Bibliography and Master Index for NetView, NCP, and VTAM*

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Writing VTAM Exit Routines

Session Management Exit Routine . . . 3
Register Contents . . . 7
Parameter List Structure . . . 8
Parameter Descriptions . . . 19
MVS VM Session Flow for Alias Selection Function . . . 39
Design Requirements . . . 47
Virtual Route Selection Exit Routine . . . 48
Initial Register and Parameter List Contents . . . 49
Changing the Virtual Route Selection List . . . 49
Design Requirements . . . 53
Final Register Contents . . . 54
VR Pacing Window Size Calculation Exit Routine . . . 54
Initial Register and Parameter List Contents . . . 55
Design Requirements . . . 56
Final Register Contents . . . 56
MVS Using the VR Window Size Calculation Exit Routine for IMS . . . 57
Session Accounting Exit Routine . . . 58
Initial Register Contents . . . 58
Design Requirements . . . 59
Final Register Contents . . . 59
Session Authorization Exit Routine . . . 60
Initial Register and Parameter List Contents . . . 61
Design Requirements . . . 61
Final Register Contents . . . 63
VSE TPRINT Processing Exit Routine . . . 63
Initial Register and Parameter List Contents . . . 63
Design Requirements . . . 64
Final Register Contents . . . 64
Example of a TPRINT Exit Routine . . . 65
Installing VTAM Exit Routines . . . 66

© Copyright IBM Corp. 1984, 1990

Chapter 1. Writing VTAM Exit Routines

This chapter describes user-coded exit routines that control session management, virtual route selection, virtual route pacing window size, trace print utility (TPRINT), session accounting, and session authorization processing.

Session Management Exit Routine

The session management exit routine enables you to combine session-related functions (authorization, accounting, gateway path selection, system services control point (SSCP) selection, and alias selection) into one exit routine.

The session management exit routine is called in the SSCPs of the origin logical unit (OLU) and destination logical unit (DLU), as well as in every SSCP along the path of the SSCP-SSCP session used to support a particular LU-LU session.

Note: The session management exit routine will not be called in an SSCP used as an intermediate routing node (IRN).

Although the session management exit routine is recommended especially for cross-network sessions, it is also recommended for single-network sessions. In the latter case, it can replace the authorization and accounting exit routines. It can also be called for the adjacent SSCP function and as a directory service for the alias selection function. It is not called for gateway path selection for single-network sessions, since there are no gateway paths to be selected.

When you use the session management exit routine for gateway path selection in cross-network sessions, you can modify the list of alternate gateway paths. The exit routine is called with the list of alternate gateway paths and the names and network identifiers of the LU-LU session partners. You can modify the order of the list or shorten it, thus changing the order in which the gateway paths are considered for use in this session. The first entry in the list is the GWPATH for the gateway NCP that was used for the SSCP-SSCP session, over which the LU session request was received. The gateway path selection function lets you distribute the LU-LU sessions across gateway NCPs according to your particular requirements.

Similarly, when you use the session management exit routine for adjacent SSCP selection, you can shorten or reorder the list from which the next SSCP used in the session setup is chosen. The session management exit routine receives as input either a default list or a list of adjacent SSCPs that was built during the previous LU-LU session setup for the resource.

MVS VM The alias selection function of the session management exit routine is an alternative method of translating names that can complement or replace the NetView* alias application. The alias selection function translates LU names, class-of-service (COS) names, and logmode names. It can also determine owning SSCP names. After processing the names passed to the exit, you can indicate that the NetView alias application should be called to translate any names not already trans-

* Trademark of the IBM Corporation. See "Special Notices" on page iii for a list of all IBM trademarks used in this book.

lated, the NetView alias application should not be called at all, or the session should be terminated.

Examples of when the session management exit routine functions are driven are shown in Table 2. A sample cross-network session between APPLA and APPLB, illustrating when the functions are driven, is shown in Figure 1 on page 5. A sample cross-network session for INIT OTHER CD processing is illustrated in Figure 2 on page 6. These sample sessions do not include the alias selection function. Examples of session flows for the alias selection function are shown in "MVS VM Session Flow for Alias Selection Function" on page 39.

Table 2. Example of Session Management Exit Routine Functions

Function	Driven
Begin	Once during VTAM initialization.
Adjacent SSCP selection	Once during session initialization when the adjacent SSCP table is being built.
Gateway path selection	During session initialization each time a new adjacent SSCP is selected and gateway path list is built (cross-network only).
Alias selection MVS VM	During session initialization each time a name needs to be translated and the NetView alias application might be called.
Initial authorization	Once during session initialization.
First time	After an adjacent SSCP is selected in the DLU direction. (For INIT OTHER CD, this occurs after the SSCP is selected in the secondary logical unit (SLU) direction.)
Second through *n*th time	Each time a new adjacent SSCP is selected in the DLU direction.
Secondary authorization	Once during session initialization when the DLU has been located and more information is known.
Initial accounting	Once during session initialization when all session started signals have arrived and the session is active.
Final accounting	Once during session termination.
End	Once during VTAM termination.

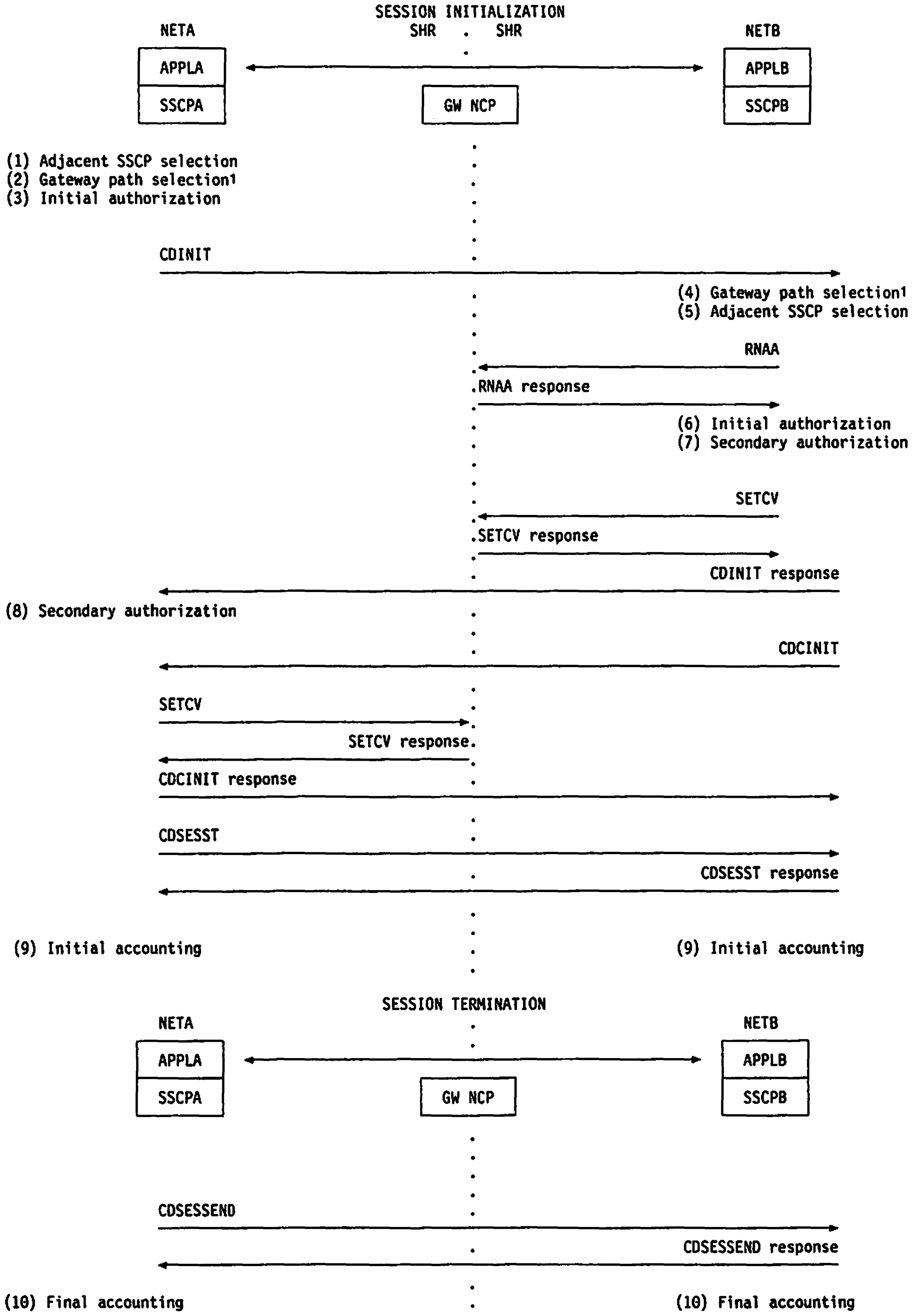


Figure 1. Sample Cross-Network Session

[1] This function applies to MVS and VM only (see figures 1 and 2).

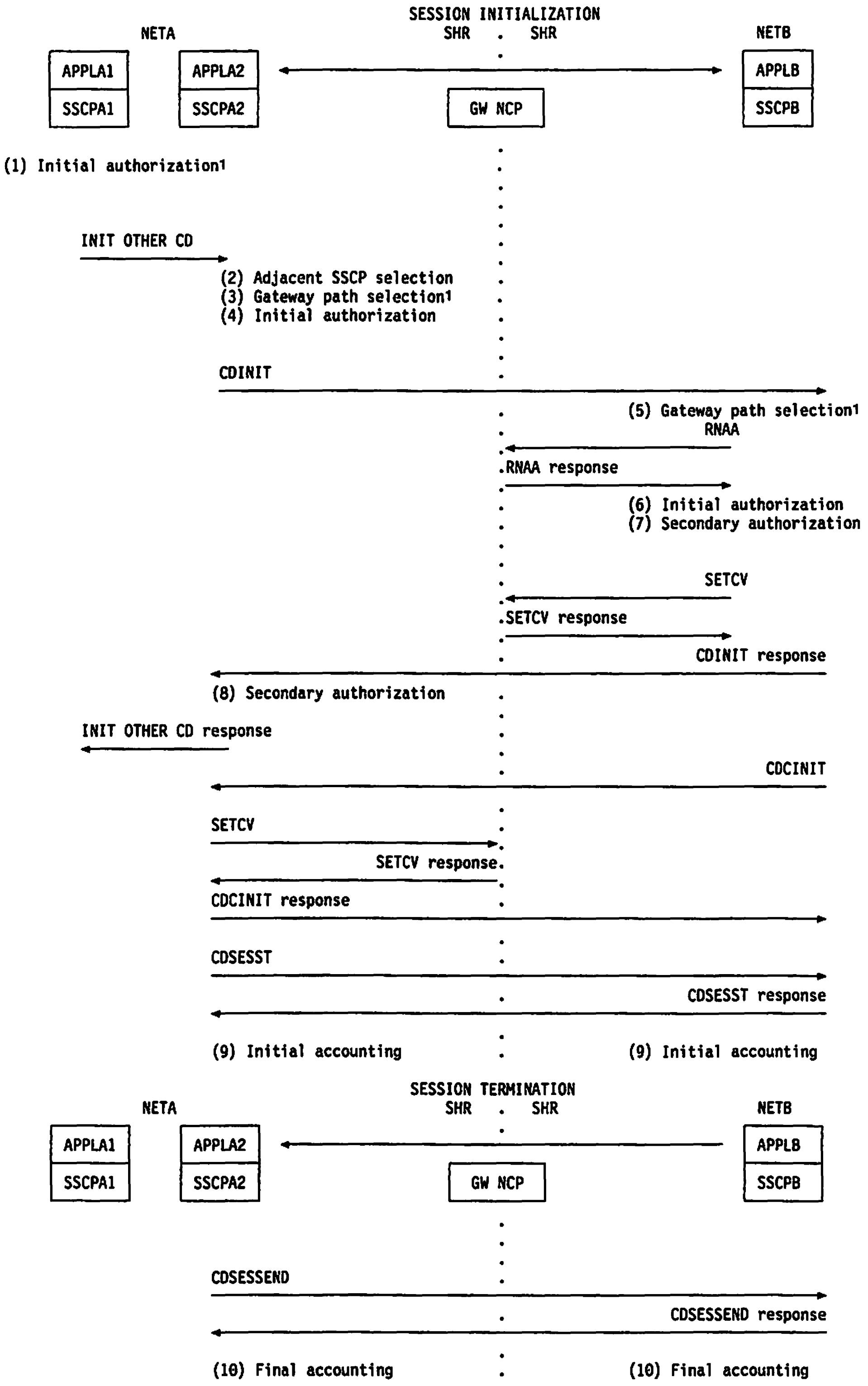


Figure 2. Sample Cross-Network Session for Init Other CD

The following sections contain information you need to write this routine.

Register Contents

When VTAM passes control to the session management exit routine, register contents are as follows:

Register 1: Address of a variable-length list of virtual storage addresses, which point to fields of parameter data. The end of the parameter list is indicated by a "1" in the high order bit of the last word.

Register 13: Address of an 18-fullword save area.

Register 14: Return address.

Register 15: Address of the entry point of the routine.

The final register contents vary depending on which function is being performed. They are described in the sections pertaining to the individual functions.

If any function other than the end function abends, VTAM posts return code 16 and sends a message to the operator. When the session management exit routine is disabled:

- VTAM authorizes all sessions and discards accounting data meant for the exit routine
- GWPATH definition statement determines the gateway path selection
- VTAM selects an adjacent SSCP from the original list and invokes the NetView alias application, if active.

Several functions are defined for the session management exit routine. All functions except the begin function are optional.

The parameters VTAM passes to the exit routine are described in the following section. Although the parameter lists vary depending on the use of the exit routine, the basic format is the same for all of the functions for which the exit routine is called. The input for each function appears at the same location in each parameter list.

The possible input parameter lists are:

- Begin
- Initial authorization
- Secondary authorization
- Initial accounting
- Final accounting
- MVS Session takeover accounting (XRF only)
- MVS VM Gateway path selection
- Adjacent SSCP selection
- MVS VM Alias selection
- End.

Table 3 on page 8 and Table 4 on page 9 summarize the parameters contained in each list.

The function code (described in "Function Code and Related Session Information" on page 20) indicates the function being processed.

If you do not provide a session management exit routine, a session accounting exit routine, or a session authorization exit routine, VTAM authorizes all sessions and discards accounting data meant for the exit routine. MVS VM When you provide none of these exit routines, VTAM:

- Selects a gateway path from the list determined by the GWPATH definition statements
- Selects an adjacent SSCP from the default list
- Sends all requests to a NetView alias application, if one is active.

Parameter List Structure

Table 3. Summary of Parameter Lists for Session Management Exit Routine (Part 1)

FUNCTION		BEGIN	INITIAL AUTHORIZATION	SECONDARY AUTHORIZATION	INITIAL ACCOUNTING	FINAL ACCOUNTING
Offset **dec**	**hex**	**Function Code X'FE'**	**Function Code X'00'**	**Function Code X'01'**	**Function Code X'02'**	**Function Code X'03'**
0	0	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14
4	4	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15
8	8	User Data Field Address	User Data Field Address	User Data Field Address	User Data Field Address	User Data Field Address
12	C	Exit Options Address See Table 16	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17
16	10	N/A	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17
20	14	N/A	Session ID Address	Session ID Address	Session ID Address	Session ID Address
24	18	N/A	Reserved	Reserved	Time of Day Address	Time of Day Address
28	1C	N/A	OLU Gateway Information Vector Address See Table 19	OLU Gateway Information Vector Address See Table 19	OLU Gateway Information Vector Address See Table 19	OLU Gateway Information Vector Address See Table 19
32	20	N/A	Reserved	DLU Gateway Information Vector Address See Table 22	DLU Gateway Information Vector Address See Table 22	DLU Gateway Information Vector Address See Table 22
36	24	N/A	OLU Adjacent SSCP Vector Address See Table 23	OLU Adjacent SSCP Vector Address See Table 23	OLU Adjacent SSCP Vector Address See Table 23	OLU Adjacent SSCP Vector Address See Table 23
40	28	N/A	For INIT OTHER CD, DLU Adjacent SSCP Vector Address See Table 24	DLU Adjacent SSCP Vector Address See Table 24	DLU Adjacent SSCP Vector Address See Table 24	DLU Adjacent SSCP Vector Address See Table 24
44	2C	N/A	N/A	N/A	N/A	N/A
48	30	N/A	N/A	N/A	N/A	N/A
Reg.15 Return Code		0	0,4,8,12	0,8	0	0

Table 4. Summary of Parameter Lists for Session Management Exit Routine (Part 2)

FUNCTION		MVS VM GATEWAY PATH SELECTION	MVS SESSION TAKEOVER ACCOUNTING	ADJACENT SSCP SELECTION	MVS VM ALIAS SELECTION	END
Offset		Function Code	Function Code	Function Code	Function Code	Function Code
dec	hex	X'04'	X'05'	X'06'	X'07'	X'FF'
0	0	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14	Environment Vectors Address See Table 14
4	4	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15	Function Code and Related Session Information Address See Table 15
8	8	User Data Field Address	User Data Field Address	User Data Field Address	User Data Field Address	User Data Field Address
12	C	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17	PLU Resource ID Control Vector Address See Table 17	N/A
16	10	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17	SLU Resource ID Control Vector Address See Table 17	N/A
20	14	Session ID Address	Session ID Address	Session ID Address	Session ID Address	N/A
24	18	Reserved	Time of Day Address	Reserved	Reserved	N/A
28	1C	Gateway Path Selection List Address See Table 20	OLU Gateway Information Vector Address See Table 19	SSCP Name List Address See Table 21	OLU Gateway Information Vector Address See Table 19	N/A
32	20	N/A	DLU Gateway Information Vector Address See Table 22	N/A	DLU Gateway Information Vector Address See Table 22	N/A
36	24	N/A	OLU Adjacent SSCP Vector Address See Table 23	N/A	OLU Adjacent SSCP Vector Address See Table 23	N/A
40	28	N/A	DLU Adjacent SSCP Vector Address See Table 24	N/A	DLU Adjacent SSCP Vector Address See Table 24	N/A
44	2C	N/A	N/A	N/A	Alias Selection Input Parameter List Address See Table 25	N/A
48	30	N/A	N/A	N/A	Alias Selection Output Parameter List Address Table 27	N/A
Reg.15 Return Code		0,4	0	0,4	0,4,8,12,16,20	0

Begin Function (Function Code X'FE')

The begin function of the session management exit routine is required; all other functions are optional. This function is processed only once during VTAM initialization, before any of the other functions are processed.

During begin function processing, the exit routine selects the other functions to process for all LU-LU sessions. If not selected at this time, the session management exit routine is not invoked for the other functions. Table 5 on page 10 shows the begin parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 5. Begin-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of function code and related session information
8 (8)	4	Address of user data field
12 (C)	4	Address of exit options

Final Register Contents: The routine must leave the register status as follows:

Registers 1–14:	Restore these registers.
Register 15:	0.

If the exit routine returns an invalid return code (that is, a nonzero return code) or invalid data as a result of processing the begin function, VTAM continues as if no exit routine exists. All sessions are authorized, accounting data is discarded, the list of gateway paths determined by the GWPATH statements are used for gateway path selection, an adjacent SSCP is selected from the original list, and the NetView alias application is used for translation if the application is active. VTAM issues message IST793E to inform the operator.

If the alias selection function is chosen during begin processing, the alias parameter list is obtained at this time. If the parameter list cannot be obtained, VTAM issues IST793E and the alias selection function is not invoked to perform translations.

Initial Authorization Function (Function Code X'00')

The session management exit routine invokes this function during session initiation as well as during SSCP takeover. Using this function, the session management exit routine may or may not allow a session. If, during the begin function, VTAM sets the proper bit in the exit options, indicating that authorization is allowed for INIT OTHER CD processing as well as normal initiation processing, the initial authorization function is also invoked for CLSDST PASS processing. The exit routine also invokes both initial and secondary authorization exits during LU takeover processing if the sessions being taken over were established using extended BIND protocols.

For normal initiate processing, the initial authorization function may also defer the decision of allowing a session to the secondary authorization function when more information is available, for example:

- The DLU's real name
- The DLU's network identifier
- The DLU's owning SSCP.

For normal initiate processing, the initial authorization function does not know the DLU gateway information vector (GIV) during initial authorization time. It can, however, be determined by looking at the parameter list obtained when the secondary authorization is invoked.

If the initial authorization function is not selected during begin function processing, VTAM defers authorization to the secondary authorization function. If the exit routine does not process secondary authorization either, VTAM automatically authorizes the session.

CLSDST PASS (INIT OTHER CD) processing is slightly different for initial authorization. The secondary authorization function is not driven for CLSDST PASS processing when neither the primary logical unit (PLU) nor the SLU are in the initiating LU's host; therefore, session authorization cannot be deferred. Furthermore, information is available through both the OLU and the DLU gateway information vectors for this processing. (The vectors contain information about the initiating logical unit (ILU) and SLU rather than the OLU and DLU.)

Table 6 shows the initial-authorization function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 6. Initial-Authorization-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of function code and related session information
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Reserved
28 (1C)	4	Address of OLU gateway information vector
32 (20)	4	Reserved
36 (24)	4	Address of OLU adjacent SSCP vector. (For INIT OTHER CD, address of the ILU adjacent SSCP vector)
40 (28)	4	Address of SLU adjacent SSCP vector. (Present only for INIT OTHER CD)

Final Register Contents: The routine must leave the register status as follows:

Registers 1–14: Restore these registers.

Register 15: 0, session setup authorized; do not schedule the secondary authorization function.

4, session setup authorized; secondary authorization is required if an INIT OTHER CD request is not being processed. This return code is valid only if the exit routine processes the secondary authorization function. Otherwise, session setup is not authorized, and the session setup fails. For INIT OTHER CD processing, this return code is equivalent to a zero return code.

8, session setup not authorized and session setup fails. Do not reroute.

12, session setup not authorized. Allow sending SSCP to reroute.

Note: If this return code is set in the host of the origin LU or if this return code is set during INIT OTHER CD processing, it is handled as return code 8.

Note: An invalid return code (that is, a return code not defined above) always means that session setup is not authorized and the session setup fails. VTAM issues message IST793E to inform the operator.

Secondary Authorization Function (Function Code X'01')

The secondary authorization function may receive authorization requests that were deferred by the initial authorization function or all session authorization requests, if the exit routine does not process the initial authorization function. This function is processed for initiation or LU takeover if the sessions being taken over were established using extended BIND protocols. A bit setting in the related session information indicates whether the invocation is because of session setup or failure. The secondary authorization function may receive more information about a session from VTAM than does the initial authorization function. That is, if the DLU's real name, network identifier, and owning SSCP were not known at initial authorization, they are known at secondary authorization since the DLU gateway information vector is provided.

If the exit routine processes neither initial nor secondary authorization functions (because they were not selected during begin function processing), all sessions are authorized.

Table 7 shows the secondary authorization function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 7. Secondary-Authorization-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of exit routine function code
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Reserved
28 (1C)	4	Address of OLU gateway information vector
32 (20)	4	Address of DLU gateway information vector
36 (24)	4	Address of OLU adjacent SSCP vector
40 (28)	4	Address of DLU adjacent SSCP vector

This function is processed when the CDINIT response is received.

Final Register Contents: The routine must leave the register status as follows:

Registers 0—14:	Restore these registers.
Register 15:	0, session authorized.
	8, session setup not authorized, and session setup fails.

Note: An invalid return code always means that session setup is not authorized and the session setup has failed. VTAM issues message IST793E to inform the operator.

Initial and Final Accounting Functions (Function Codes X'02' and X'03')

The initial accounting function is processed at session initiation time. The final accounting function is processed at session termination time. These functions pass accounting information only. No results are returned to VTAM.

If the initial accounting function is selected for processing, the final accounting function is also selected for processing.

If the session management exit routine does not process the accounting functions (because they were not selected during begin function processing), VTAM discards data meant for the exit routine.

Table 8 shows the accounting-function parameter list (for both initial and final accounting function) pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 8. Accounting-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of exit routine function code
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Address of time of day field
28 (1C)	4	Address of OLU gateway information vector
32 (20)	4	Address of DLU gateway information vector
36 (24)	4	Address of OLU adjacent SSCP vector
40 (28)	4	Address of DLU adjacent SSCP vector

Final Register Contents: The routine must leave the register status as follows:

Registers 0–14:	Restore these registers.
Register 15:	0.

Note: An invalid return code (that is, a nonzero return code) is ignored. VTAM issues message IST793E to inform the operator.

MVS VM Gateway Path Selection Function (Function Code X'04')

The gateway path selection function enables you to shorten or reorder the list of gateway paths determined by the GWPATH definition statements. (See *VTAM Resource Definition Reference* for an explanation of the GWPATH statement.) If this function is not selected during begin function processing, VTAM uses the original gateway path list determined by the GWPATH statements.

Table 9 on page 14 shows the gateway path selection function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 9. Gateway-Path-Selection-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of exit routine function code
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Reserved
28 (1C)	4	Address of gateway path selection list

Final Register Contents: The routine must leave the register status as follows:

Registers 0–14: Restore these registers.

Register 15: 0, proceed with session setup; the gateway path list returned by the exit routine is used. Modification of the gateway path list by the exit routine is optional.

4, proceed with session setup; the original gateway path list (determined by the GWPATH definition statements) is used.

Note: If an invalid return code is received, the original gateway path list determined by the GWPATH statements is used for gateway path selection. The original gateway path list is also used if the returned list is empty or if it is larger than the list that was passed to the routine. VTAM issues message IST793E to inform the operator.

MVS Session Takeover Accounting Function for XRF (Function Code X'05')

This primary function code indicates that the session management exit routine has been called to process a session takeover by the alternate XRF application. The exit routine is driven for this function when the alternate application has completed its takeover of a session. The session takeover accounting function notifies the exit routine of a change of session status.

If this function is not selected during begin function processing, data intended for it is discarded.

A bit setting in the related session information (shown in Table 15 on page 21) indicates whether a session is a backup XRF session. This setting applies to primary function codes hex 00–07. It does not apply to the begin and end functions.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Table 10 shows the session takeover accounting function parameter list that is pointed to by register 1. These parameters are described in "Parameter Descriptions" on page 19.

Table 10. Session-Takeover-Accounting-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of function code and related session information
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Address of time of day
28 (1C)	4	Address of OLU gateway information vector
32 (20)	4	Address of DLU gateway information vector
36 (24)	4	Address of OLU adjacent SSCP vector
40 (28)	4	Address of DLU adjacent SSCP vector

Final Register Contents: The routine must leave the register status as follows:

Registers 0—14:	Restore these registers.
Register 15:	0.

Note: An invalid return code (that is, a nonzero return code) in register 15 is ignored. VTAM issues message IST793E to inform the operator.

Adjacent SSCP Selection Function (Function Code X'06')

An SSCP invokes the adjacent SSCP selection function during the processing of an LU-LU session request to select the SSCP to which the request is routed. This function enables you to shorten or reorder the list of SSCPs from which the next SSCP used in session setup is chosen during CDINIT or DSRLST routing. This function is invoked for direct search list (DSRLST) routing only if it is enabled during the begin function. If no adjacent SSCPs exist, this function is not invoked.

The session management exit routine receives as input the user-defined list of adjacent SSCPs and the list of adjacent SSCPs that were built during the previous LU-LU session setup for the same resource (if any).

By using the dynamic adjacent SSCP table function in conjunction with the adjacent SSCP selection function, you can maintain control of session routing without having to define adjacent SSCP tables. The SSCP selection function is provided with a list of all active adjacent SSCPs, which it alters as necessary, to control session routing. To use the dynamic adjacent SSCP table function, you must specify the function in the start options.

Both the user-defined tables and the dynamic tables are searched in the following manner:

1. If the NETID and the CDRM names are known, a search is done for a table coinciding with the known information.
2. If that table is not found or if only NETID is known, a search is done for a table defined with just the NETID given.
3. If an attempt to find that table fails or if only the CDRM name is known, the indicated CDRM adjacent SSCP table for all networks is searched.
4. If an attempt to find that table fails, the default adjacent SSCP table for all networks is used.
5. If an attempt to find that table fails and the SSCP is using the dynamic adjacent SSCP function, the dynamic adjacent SSCP table is used.
6. If no tables exist or if a usable table could not be located, routing is attempted to any dynamic adjacent SSCP entries. This includes entries from the CDRM-CDRM sessions for this host.

   Note: If the owner is known, it is provided in the PLU or the SLU resource identification vector, or both.

Table 11 shows the SSCP-selection-function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 11. SSCP-Selection-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of exit routine function code
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Reserved
28 (1C)	4	Address of SSCP name list

Final Register Contents: The routine must leave the register status as follows:

Registers 0–14: Restore these registers.

Register 15: 0, proceed with session setup; use SSCP name list returned by this function. Changing the SSCP name list is optional.

4, proceed with session setup; standard VTAM routing is used.

If an invalid return code is returned or if the exit routine terminates abnormally, the session initiation request continues to be processed using the original list of adjacent SSCPs received by the exit routine. VTAM also uses the original adjacent SSCP list if the returned list is invalid, for instance, if it is empty or if it is larger than the list that was passed to the routine. VTAM issues error message IST793E to inform the operator.

MVS VM Alias Selection Function (Function Code X'07')

This function is processed along the session setup path during session initiation. The alias selection function can be called during:

- INIT OTHER CD processing to translate the PLU and the SLU names, SLU logmode name, PLU- and SLU-owning SSCP names, and SLU COS name
- CDINIT request routing processing to determine the DLU-owning SSCP name and to translate the DLU real names, OLU alias name, DLU logmode and COS names, and associated LU alias names
- CDINIT response time to translate the COS names, associated LU alias names, and USERVAR names
- DSRLST routing time to translate the DLU real names and the USERVAR names and to determine the DLU-owning SSCP names.

Table 12 shows the alias-selection-function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 12. Alias-Selection-Function Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of environment vectors
4 (4)	4	Address of exit routine function code
8 (8)	4	Address of user data field
12 (C)	4	Address of PLU resource identifier control vector
16 (10)	4	Address of SLU resource identifier control vector
20 (14)	4	Address of session ID
24 (18)	4	Reserved
28 (1C)	4	Address of OLU gateway information vector
32 (20)	4	Address of DLU gateway information vector
36 (24)	4	Address of OLU-adjacent SSCP vector
40 (28)	4	Address of DLU-adjacent SSCP vector
44 (2C)	4	Address of alias input parameter list
48 (30)	4	Address of alias output parameter list

Final Register Contents: The routine must leave the register status as follows:

Registers 1 – 14: Restore these registers.

Register 15: 0, no information has been translated; do not invoke the NetView alias application.

4, no information has been translated; invoke the NetView alias application to translate all data.

8, some information has been translated; do not invoke the NetView alias application.

12, some information has been translated; invoke the NetView alias application to translate the remaining information.

16, session initiation cannot continue through this SSCP; allow the sending SSCP to reroute the request.

20, session initiation cannot continue through this SSCP; do not allow the sending SSCP to reroute the request. The session will fail to set up.

Note: Only CDINIT and DSRLST requests can be rerouted. All other RU types fail.

End Function (Function Code X'FF')

This function of the session management exit routine is processed during VTAM termination (HALT or HALT QUICK). This function should perform any cleanup required and return processing to VTAM. Note that HALT CANCEL causes VTAM to abend. In this case, the end function is not processed.

Table 13 shows the end-function parameter list pointed to by register 1. The parameters are described in "Parameter Descriptions" on page 19.

Table 13. End-Function Parameter List

Offset	Size (Bytes)	Description
0	4	Address of environment vectors
4	4	Address of exit routine function code
8	4	Address of user data field

Final Register Contents: The routine must leave the register status as follows:

Registers 0 – 14: Restore these registers.
Register 15: 0, continue with termination.

Note: An invalid return code (that is, a nonzero return code) causes VTAM to con-. tinue termination. VTAM issues message IST793E to inform the operator.

VTAM termination does not complete if the exit routine does not complete. In this case, the operator must issue one of the following to terminate VTAM:

- MVS VM a HALT CANCEL command
- VSE a CANCEL Fx (the VTAM partition).

Parameter Descriptions

The descriptions of each of the parameters that follow refer to Table 5 on page 10 through Table 13 on page 18. These figures are summarized in Table 3 on page 8.

Environment Vector List

The address of the environment vector list is the first word of the parameter list for all function codes. These vectors, which are preceded by a 2-byte length field, make it possible for your installation to use a single exit routine, installed in each host, to collect session authorization and accounting data. Although all information collected may be combined for analysis, it is possible to identify the host at which the data was collected.

The environment vector list has the same format as the resource identification vector list described in *VTAM Programming*. (Do not confuse this with the resource identifier control vector described later in this section.) The parameter list points to the environment vector list in Table 14.

Table 14 (Page 1 of 2). Environment Vectors

Comments	Offset	Size (Bytes)	Description
A header precedes the list of vectors:			
	0	2	Total length of the parameter list including this length field (m)
This header is followed by several vectors, which may be in any order. The network identification vector:			
	0	1	Length of the vector including this field (n)
	1	1	ID of the vector (X'06')
	2	n−2	ID of the network in which this exit routine is operating (from the NETID start option)
The SSCP-name vector:			
	0	1	Length of the vector including this field (n)
	1	1	ID of the vector (X'07')
	2	n−2	SSCP name of the host in which this exit is operating. The start option SSCPNAME is required.
The host-PU-name vector:			
	0	1	Length of the vector including this field (n)
	1	1	ID of the vector (X'08')
	2	n−2	Name of the host in which this exit is operating (from the HOSTPU start option)
The host-PU-network-address vector:			
	0	1	Length of the vector including this field
	1	1	ID of the vector (X'09')

Table 14 (Page 2 of 2). Environment Vectors

Comments	Offset	Size (Bytes)	Description
	2	6	Network address of the host PU
The adjacent OLU-network-name vector is used for the following functions:			
Initial authorization Secondary authorization Initial accounting Final accounting Gateway path selection Adjacent SSCP selection Alias selection	0	1	Length of the vector including this field If this information is not applicable, the length field is represented by X'00' and the actual length of the vector is X'01'.
	1	1	ID of the vector (X'0A')
	2	n–2	Network name of the adjacent network in the OLU direction
The adjacent DLU-network-name vector is used for the following functions:			
Secondary authorization Initial accounting Final accounting Alias selection	0	1	Length of the vector including this field This field is zero if this information is not applicable.
	1	1	ID of the vector (X'0B')
	2	n–2	Network name of the adjacent network in the DLU direction

Function Code and Related Session Information

The address of the function code and the related session information is always passed to the session management exit routine as the second parameter in the function parameter list. The function code indicates which of the functions of the session management exit routine will be processed. The first byte (byte 0) is the function code. The parameters passed to the exit routine depend on the function code.

The second and third bytes (bytes 1 and 2, respectively) are the related session information. The second byte contains information about the initiator of the session, which type of request (CDINIT or DSRLST) is being processed, whether the session management exit routine is being driven for takeover, and whether a session is a backup extended recovery facility (XRF) session.

The third byte contains information about the SSCP that invoked the session management exit routine and indicates whether the RU was a request or a response, and whether an INIT OTHER CD or some other RU was being processed. (The other RUs would be CDINIT or DSRLST. You can determine the RU by referring to the related session information field.)

See Table 15 for function code and related session information.

Table 15 (Page 1 of 3). Function Code and Related Session Information

Byte	Description	
0	Function Code:	
	X'00'	The exit routine has been called for initial session authorization; only partial information may be available. This function is processed for initiation or LU takeover if the sessions being taken over were established using extended BIND protocols.
	X'01'	The exit routine has been called for secondary session authorization; complete session initiation information is available. This function is processed for initiation or LU takeover if the sessions being taken over were established using extended BIND protocols. A bit setting in the related session information indicates whether the invocation is due to session setup or failure.
	X'02'	The exit routine has been called for session accounting at session setup completion. The session is in an active state.
	X'03'	The exit routine has been called for session accounting at the completion of session termination.
	X'04'	The exit routine has been called for gateway path selection. The related session information defines whether the function is invoked for the DLU or OLU direction.
	X'05'	The exit routine has been called for XRF session switch processing.
	X'06'	The exit routine has been called for SSCP selection.
	X'07'	The exit routine has been called for the alias selection function. CDINIT, DSRLST, or INIT OTHER CD is being processed by this SSCP.
	X'FE'	The exit routine has been called to select the functions it will process; this begin function is processed only once.
	X'FF'	The exit routine has been called to perform required cleanup during VTAM termination. This end function is processed only once during VTAM termination.
1	Related Session Information The following bit settings indicate whether a session is a backup XRF session. (The settings are defined for primary function codes hex 00—07.)	
	B'0... ....'	The session is not a backup XRF session. If the primary function code is hex 05, the session is now a primary session and was previously a backup session. For all other valid primary function codes, the session is either a primary session or not enabled for XRF.
	B'1... ....'	The session is a backup XRF session. If the primary function code is hex 05, the session was previously a primary XRF session.

Table 15 (Page 2 of 3). Function Code and Related Session Information

Byte	Description	
	For the session authorization or alias selection functions (primary function codes hex 00, 01, or 07), the following flags define the session initiator:	
	`B'..00 ....'`	Autologon session (VARY LOGON, LOGAPPL)
	`B'..01 ....'`	PLU requested the session (SIMLOGON, OPNDST ACQUIRE). (PLU is the OLU, SLU is the DLU.)
	`B'..10 ....'`	SLU requested the session (USS logon, INIT SELF). (SLU is the OLU, PLU is the DLU.)
	`B'..11 ....'`	Some other LU requested the session: third party initiation (CLSDST PASS).
	For the session authorization or alias selection functions (primary function codes hex 00, 01 or 07), the following flag defines whether the DLU real network ID is assumed:	
	`B'.... 0...'`	DLU real network ID is not assumed
	`B'.... 1...'`	DLU real network ID is assumed and is being processed.
	For the SSCP selection or alias selection function (primary function codes hex 06 or hex 07), the following flag defines whether the function is invoked for CDINIT or DSRLST routing:	
	`B'.... .0..'`	SSCP or alias selection invoked for CDINIT routing
	`B'.... .1..'`	SSCP or alias selection invoked for DSRLST routing.
	For the gateway path selection function (primary function code hex 04), the following flag defines whether the function is invoked for DLU or OLU direction:	
	`B'.... ..0.'`	Gateway path selection invoked for OLU direction
	`B'.... ..1.'`	Gateway path selection invoked for DLU direction.
	For primary function codes hex 00 – 02, the following flag indicates whether session management exit routine functions are being driven during takeover:	
	`B'.... ...0'`	Session management exit function not driven for takeover
	`B'.... ...1'`	Session management exit function driven for takeover.

Table 15 (Page 3 of 3). Function Code and Related Session Information

Byte	Description
2	Related Session Information: For the initial authorization function (primary function code hex 00) and the alias selection function (primary function code hex 07), the following flag defines whether the function is invoked for INIT OTHER CD routing, or for some other type of RU. The initial authorization function is invoked only for INIT OTHER CD and CDINIT routing. `B'1... ....'` Alias selection or initial authorization invoked for INIT OTHER CD routing `B'0... ....'` Alias selection or initial authorization invoked for other routing (CDINIT or DSRLST routing, see related description under Byte 1 above to determine which type of routing). For the alias selection function (primary function code hex 07), the following flag defines which SSCP requested translation: `B'.00. ....'` The SSCP of the ILU requested translation `B'.01. ....'` The SSCP of the OLU requested translation `B'.10. ....'` The SSCP of the DLU requested translation `B'.11. ....'` An intermediate SSCP (neither the OLU nor the DLU) requested translation. For the alias selection function (primary function code hex 07), the flag defines whether the function is invoked because of a request or response: `B'...0 ....'` Alias selection invoked because of a request `B'...1 ....'` Alias selection invoked because of a response. For the secondary authorization function (primary function code hex 01), the following flag indicates whether session initiation has succeeded: `B'.... 0...'` Session setup has succeeded; a positive initiate response is being processed. `B'.... 1...'` Session setup has failed; a negative response to session initiation or a session setup failure request is being processed. The remaining bits are reserved.

User Data Field

The address of the user data field is always passed to the exit routine as the third parameter in the function parameter list. This word points to a 4-byte user data field, originally initialized to 0, which the exit routine can use for any purpose (for example, to store the address of a dynamically obtained storage area). VTAM saves the contents of this user field after every successful invocation of the session management exit routine, so that it is available to the exit routine the next time it is called.

Notes:

1. Do not confuse this field with VTAM's user data field. This field is used only by the session management exit routine.
2. VTAM returns the address of this field on subsequent calls, not the data itself.

Exit Options

The address of the exit options field is passed to the exit routine as the fourth parameter of the parameter list only for the begin function. This word points to a 2-byte field that indicates the functions for which the exit routine will be called. This field is modified by the exit routine during begin function processing. The bit definitions are described below:

Table 16 (Page 1 of 2). Session Management Exit Routine Options

Byte	Bit	Description
0	`B'1... ....'`	The exit routine will process the initial authorization function.
	`B'0... ....'`	The exit routine will **not** process the initial authorization function. (The default is 0.)
	`B'.1.. ....'`	The exit routine will process the secondary authorization function.
	`B'.0.. ....'`	The exit routine will **not** process the secondary authorization function. (The default is 0.)
	`B'..1. ....'`	The exit routine will process the initial and final accounting functions.
	`B'..0. ....'`	The exit routine will **not** process the accounting functions. (The default is 0.)
	`B'...1 ....'`	The exit routine will process the gateway path selection function.
	`B'...0 ....'`	The exit routine will **not** process the gateway path selection function. (The default is 0.)
	`B'.... 1...'`	The exit routine will process the end function.
	`B'.... 0...'`	The exit routine will **not** process the end function. (The default is 0.)
	`B'.... .1..'`	The exit routine will process the session takeover accounting function.
	`B'.... .0..'`	The exit routine will not process the session takeover accounting function (primary function code hex 05). (The default is 0.)
	`B'.... ..1.'`	The exit routine will process the SSCP selection function.
	`B'.... ..0.'`	The exit routine will **not** process the SSCP selection function. (The default is zero.)
	`B'.... ...1'`	SSCP selection is allowed for DSRLST.
	`B'.... ...0'`	SSCP selection is not allowed for DSRLST. (The default is 0.)
1	`B'1... ....'`	Session management exit functions are allowed for takeover of LU sessions.
	`B'0... ....'`	Session management exit functions are not allowed for takeover of LU sessions. (The default is 0.)
	`B'.1.. ....'`	Initial authorization is allowed for INIT OTHER CD.
	`B'.0.. ....'`	Initial authorization is not allowed for INIT OTHER CD. (The default is 0.)

Table 16 (Page 2 of 2). Session Management Exit Routine Options

Byte	Bit	Description
	B'..1. ....'	The exit routine will process the alias selection function.
	B'..0. ....'	The exit routine will not process the alias selection function. (The default is 0.)
	The remaining bits are reserved.	

SLU and PLU Resource Identifier Control Vector

The address of the PLU or the SLU resource identifier control vector is passed to the exit routine as the fourth or fifth parameter, respectively, of the function parameter list for primary function codes hex 00—07.

These words point to resource identifier control vectors for the PLU and the SLU in the session. If the primary function code is hex 01, 02, 03, or 05, all the information described later in this section (that is, the resource identifier control vector) is available. If the primary function code is hex 00, 04, 06 or 07, only a portion of the information may be available. The real name of the destination LU and the alias name of the origin LU may be unknown as well as the name of the destination LU's owning SSCP. If unknown, its field length is set to zero and the length field of the next vector immediately follows.

The PLU and the SLU hierarchy control vector (Table 18 on page 27) follows the PLU resource identifier control (RIC) vector in the function parameter lists. The hierarchy control vector is not part of the RIC.

The format of the PLU resource identifier control vector is the same as for the SLU (see Table 17 on page 26).

Table 17 (Page 1 of 2). PLU and SLU Resource Identifier Control Vector

Comments	Offset	Size (Bytes)	Description
	0	1	Vector key (X'19')
	1	1	Vector length (number of bytes of vector, not including the vector key and length of this length field) **Note:** If any of the name fields do not exist (cannot be determined at the time the exit is called), the length field will be zeros and the length field of the next name in the RIC vector immediately follows. The vector data length may include the LU-address vectors when sending data for session management exits.
	2	1	SSCP rerouting count (number of SSCPs remaining in the count for this session setup request). This byte is reserved if the function code and related session information is X'03'.
	3	1	Usage indicators. The flags are defined as follows: `B'1... ....'` This name has been translated `B'0... ....'` This name has not been translated `B'.1.. ....'` The resource is the target resource **Note:** The OLU is **never** the target resource. `B'.0.. ....'` The resource is not the target resource **Note:** The DLU is **always** the target resource.
This is followed by the SSCP name:			
	0	1	Length of SSCP name (m)
	1	m	Symbolic name of SSCP that controls the LU
This is followed by the network ID:			
	0	1	Length of network ID (n)
	1	n	Network ID of the network containing the LU
This is followed by the real LU name:			
	0	1	Length of the LU name (p)
	1	p	Network name of the LU (real name)
This is followed by another network ID:			
	0	1	Length of network ID (q)
	1	q	Network ID of the network in which the following alias LU is known

Table 17 (Page 2 of 2). PLU and SLU Resource Identifier Control Vector

Comments	Offset	Size (Bytes)	Description
This is followed by the alias LU name:			
	0	1	Length of alias LU name (r)
	1	r	Alias LU name
This is followed by three LU-address vectors. (The LU-address vectors are not part of the RIC vector.)			
	0	1	ID of the vector (X'1A')
	1	1	Vector length (number of bytes of vector, not including the vector key and length of this length field)
	2	6	PLU or SLU address in this network
	8	1	This byte is not filled in when passed to the exit.
	9	1	This byte is not filled in when passed to the exit.
	0	1	ID of the vector (X'1A')
	1	1	Vector length (number of bytes of vector, not including the vector key and length of this length field)
	2	6	PLU or SLU address in the adjacent network in the OLU direction
	8	1	This byte is not filled in when passed to the exit.
	9	1	This byte is not filled in when passed to the exit.
	0	1	ID of the vector (X'1A')
	1	1	Vector length (number of bytes of vector, not including the vector key and length of this length field)
	2	6	PLU or SLU address in the adjacent network in the DLU direction
	8	1	This byte is not filled in when passed to the exit.
	9	1	This byte is not filled in when passed to the exit.

Table 18 (Page 1 of 2). PLU and SLU Hierarchy Control Vector

Comments	Offset	Size (Bytes)	Description
The PLU and the SLU resource identifier control vectors are each followed by a hierarchy resource vector. (The hierarchy vector is not part of the RIC vector.)			
	0	1	ID of the vector (X'FF')
	1	1	Vector length (number of bytes of vector, not including the vector key and length of this length field)
	2	1	Number of hierarchy resource entries (n)

Table 18 (Page 2 of 2). PLU and SLU Hierarchy Control Vector

Comments	Offset	Size (Bytes)	Description
This header is followed by a variable number (n) of hierarchy resource entries.			
	0	8	Resource name
	8	1	Resource type 26, add dec 27-29. Dec (Hex) 1 (1) Communication controller 2 (2) APPL major node 3 (3) Local non-SNA major node 4 (4) Switched major node 5 (5) Local SNA major node 6 (6) CDRM major node 7 (7) CDRSC major node 8 (8) CA major node 9 (9) Reserved 10 (A) CDRM 11 (B) Reserved 12 (C) GROUP 13 (D) Reserved 14 (E) LINE 15 (F) Direct attachment node 16 (10) APPL 17 (11) Reserved 18 (12) PU 19 (13) Reserved 20 (14) Reserved 21 (15) Reserved 22 (16) LU 23 (17) Link station 24 (18) CDRSC 25 (19) Reserved 26 (1A) Reserved 27 (1B) Reserved 28 (1C) LAN major node 29 (1D) Packet major node **Note:** If this field indicates the resource is an LU and the resource information indicates that the resource is not within this network, the session is being established using non-native connection.
	9	3	Reserved

Session Identifier

The session ID is passed to the exit routine for all primary function codes **except** hex FE and FF.

This word of the parameter list points to an 8-byte session identifier. It is a unique identifier for this session within the SSCP in which the session management exit routine is scheduled.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Time-of-Day Field

This word of the parameter list points to an 8-byte field containing the time of day that the session was started (for primary function code hex 02), ended (for primary function code hex 03), or taken over (for primary function code hex 05).

The time-of-day field is in system format time.

MVS VM OLU Gateway Information Vector

This word of the parameter list points to the OLU gateway information vector.

This word is nonzero only if there is an SSCP in the OLU direction that is a gateway SSCP and the session is cross-network. If the exit routine is called within the network of the origin LU, there is no such gateway NCP and the field is set to zero.

The OLU gateway information vector provides the gateway NCP name and COS names for the gateway in the direction of the origin LU.

The format of the OLU gateway information vector is shown in Table 19. Its format is the same as that of the DLU gateway information vector.

Table 19. Format of OLU Gateway Information Vector

Comments	Offset	Size (Bytes)	Description
	0	1	Length of vector data (excluding this field) **Note:** If one of the name fields does not exist (cannot be determined at the time the exit is called), the length field will be zeros and the length field of the next name in the GIV vector immediately follows.
This is followed by the gateway node name:			
	0	1	Length of gateway NCP name (m)
	1	m	Name of the gateway NCP in the direction of the origin LU
This is followed by a COS name:			
	0	1	Length of COS name (n)
	1	n	COS name for the network on the origin LU side of the gateway NCP identified above
This is followed by another COS name:			
	0	1	Length of COS name (p)
	1	p	COS name for the network on the destination LU side of the gateway NCP identified above

MVS VM Gateway Path Selection List

This word of the parameter list points to the list of alternate gateway paths available for this LU-LU session. Table 20 on page 30 shows the format of this list.

Gateway path entries are provided for all the alternate gateway paths defined by the GWPATH definition statements for each gateway NCP for which the gateway SSCP is performing gateway path selection. The gateway path entries are passed to the session management exit routine in the order in which they are defined to VTAM (by the GWPATH definition statements). However, the gateway NCP used for an

SSCP-SSCP session is always placed first in the list if this SSCP is in session with the gateway NCP.

Note that the gateway path selection function might be processed in one SSCP to select a gateway for an adjacent SSCP.

Table 20. Format of Gateway Path Selection List

Comments	Dec (Hex) Offset	Size (Bytes)	Description
	0 (0)	2	Total length of the gateway NCP path list (excluding this field). Do not use this field to determine the last gateway path entry.
	2 (2)	8	NETID of the adjacent SSCP.
	10 (A)	2	Number of gateway path selection entries (n). Use this number to determine the last gateway path entry.
This header is followed by a variable number (n) of gateway path selection entries.			
	0 (0)	1	Length of the following gateway path information (excluding this field). Use this field to determine the end of this path specification and the beginning of the next one.
	1 (1)	8	Name of the gateway NCP connecting this network to the adjacent network (obtained from the GWN operand on the GWPATH definition statement or determined from the SUBAREA operand). This field will be zeros if the GWN operand was not coded on GWPATH and this SSCP is not in session with the gateway NCP.
	9 (9)	4	Subarea address of the gateway NCP in this network (obtained from the SUBAREA operand on the GWPATH definition statement or determined from the GWN operand). This field will be 0 if this SSCP is not in session with the gateway node and the SUBAREA operand was not coded on GWPATH. **Note:** If both GWN and SUBAREA are coded on the GWPATH definition statement, you cannot tell if this SSCP is in session with the gateway NCP.
	13 (D)	8	NETID of the adjacent network accessed by the gateway NCP identified above (from ADJNET on the GWPATH statement).

You can modify the selection list in the following ways:

- You can reduce the number of gateway path entries in the list by decreasing the number of gateway path entries (bytes 10 and 11) in the gateway path selection list. If the exit routine returns a smaller number of alternate gateway nodes than that presented to it at entry (in the gateway path selection list), VTAM uses a list of only those entries. In its attempt to find a gateway node to use for the LU-LU session, VTAM searches that list of entries in the order they were returned to VTAM by the exit routine.

- You can change the order of the gateway path entries in the list. The order of the entries returned to VTAM by the exit routine is the same order in which the gateway path entries are used to attempt LU-LU session setup.

For a list to be correct, the following requirements must be met:

- Each returned gateway path entry must match one of the entries originally received. You cannot add new entries to the list.
- The number of entries returned cannot be larger than the number of entries received.
- The adjacent NETID field of the adjacent SSCP must remain unchanged.
- The returned gateway path list must contain at least one valid entry.

If the list returned by the exit routine is invalid for any of the above reasons, the original list that was sent to the exit routine is used.

The address of the gateway path selection list should remain unchanged. The modified gateway path list should use the same storage that was passed by VTAM to the exit routine.

SSCP Name List

This word of the parameter list points to the list of adjacent SSCPs available for this LU-LU session.

Table 21 shows the format of the SSCP name list.

Table 21. Format of SSCP Name List

Comments	Offset	Size (Bytes)	Description
	0	2	Total length of the SSCP name list (excluding this field)
	2	2	Number of SSCP names in this list (n)
This header is followed by a variable number (n) of SSCP names:			
	0	8	Adjacent SSCP name

You can modify the SSCP name list in the following ways:

- You can reduce the number of SSCP name entries in the list by decreasing the number of SSCP name entries (bytes 2 and 3) in the SSCP name list. If the exit routine returns a smaller number of SSCP names than that presented to it at entry (in the SSCP name list), VTAM uses a list of only those entries. In its attempt to find an SSCP name to use, VTAM searches that list of entries in the order they were returned to VTAM by the exit routine.
- You can change the order of the SSCP name entries in the list. The order of the entries returned to VTAM by the exit routine is the same order in which the SSCP name entries are used.

For a list to be correct, the following requirements must be met:

- Each SSCP name entry returned must match one of the entries originally received. You cannot add new entries to the list.
- The number of entries returned cannot be larger than the number of entries received.
- The returned SSCP name list must contain at least one valid entry.

If the list returned by the exit routine is invalid for any of the above reasons, the original list that was sent to the exit routine is used.

The address of the SSCP name list should remain unchanged. The modified SSCP name list should use the same storage that was passed by VTAM to the exit routine.

MVS VM DLU Gateway Information Vector

The DLU gateway information vector provides the gateway NCP name and COS names for the gateway NCP in the direction of the destination LU. This parameter is set to zero if the session management exit routine is scheduled in the destination LU's network, since there is no such gateway NCP.

The format of the DLU gateway information vector shown in Table 22 is the same as that of the OLU.

Table 22. Format of DLU Gateway Information Vector

Comments	Offset	Size (Bytes)	Description
	0	1	Length of vector data (excluding this field)
This is followed by the gateway node name:			
	0	1	Length of gateway NCP name (m)
	1	m	Name of the gateway NCP in the direction of the OLU
This is followed by a COS name:			
	0	1	Length of COS name (n)
	1	n	COS name for the network on the OLU side of the gateway NCP identified above
This is followed by another COS name:			
	0	1	Length of the following COS name (p)
	1	p	COS name for the network on the DLU side of the gateway NCP identified above

OLU Adjacent SSCP Vector

The OLU adjacent SSCP vector provides the name of the adjacent SSCP in the direction of the OLU. For INIT OTHER CD processing, this provides the name of the adjacent SSCP in the direction of the ILU.

Table 23 shows the format of the OLU adjacent SSCP vector.

Table 23. Format of OLU Adjacent SSCP Vector

Comments	Dec (Hex) Offset	Size (Bytes)	Description
	0 (0)	1	Length of vector data (excluding this field)
This is followed by the adjacent SSCP name:			
	0 (0)	1	Length of the adjacent name (q)
	1 (1)	q	Name of the adjacent SSCP in the direction of the OLU

This word of the parameter list is set to zero if the session management exit routine is scheduled in the network of the OLU, since there is no OLU adjacent SSCP.

DLU Adjacent SSCP Vector

The DLU adjacent SSCP vector provides the name of the adjacent SSCP in the direction of the DLU. For INIT OTHER CD processing, this provides the name of the adjacent SSCP in the direction of the SLU.

Table 24 shows the format of the DLU adjacent SSCP vector.

Table 24. Format of DLU Adjacent SSCP Vector

Comments	Offset	Size (Bytes)	Description
	0	1	Length of vector data (excluding this field)
This is followed by the adjacent SSCP name:			
	0	1	Length of the adjacent name (q)
	1	q	Name of the adjacent SSCP in the direction of the DLU

Note: This word of the session management exit parameter list is set with the high order bit on and the remaining 31 bits off, if the session management exit routine is scheduled in the network of the DLU, since there is no DLU adjacent SSCP.

MVS VM Alias-Selection-Input Parameter List

This word of the parameter list points to the alias-selection-input parameter list for LU-LU sessions.

The alias-selection-input parameter list contains information necessary for translation and is a fixed length. Its storage is obtained during begin-function processing. If storage cannot be obtained for the alias parameter list during the begin function, message IST793E is issued indicating the error and the alias selection function is not invoked. Depending on which information needs translation, different fields are filled in.

This parameter list exists only if the SSCP is gateway-capable. To be gateway-capable, the start option GWSSCP=YES must be specified.

Table 25 on page 34 shows the format of the alias-selection-input parameter.

Table 25 (Page 1 of 3). Alias-Selection-Input Parameter List

Dec (Hex) Offset	Size (Bytes)	Comment	Description
0 (0)	1		Flags for NetView alias application
		x...	0 = NetView alias application is not active
			1 = NetView alias application is currently active
		.x..	0 = NetView alias application was never active
			1 = NetView alias application used to be active. This bit is only meaningful if the previous bit is off.
		..xx xxxx	Reserved
1 (1)	7		Reserved
		Start of NAME1 Information	
8 (8)	1		Indicates the role of the LU in the session
9 (9)	1		Indicates the primary and secondary roles of the OLU and DLU
		xxxx	Indicates whether the LU is the OLU, DLU or unknown
		 xxxx	Indicates whether the LU is the PLU, SLU, or unknown
10 (A)	1		Information pertaining to the LU name relevant for translation
		x...	1 = NAME1 is the real name 0 = NAME1 is the alias name
		.xxx xxxx	Reserved
11 (B)	5		Reserved
16 (10)	8	NAME1	LU name to be translated
24 (18)	8	NETFROM1	Network in which NAME1 is known
32 (20)	8	NETTO1	Network that NAME1 is to be translated to. Blank if exit needs to determine the network.
		Start of NAME2 Information	
40 (28)	1		Indicates the role of the LU in the session
41 (29)	1		Indicates the primary and secondary roles of the OLU and DLU
		xxxx	Indicates whether the LU is the OLU, DLU, or unknown
		 xxxx	Indicates whether the LU is the PLU, SLU, or unknown
42 (2A)	1		Information pertaining to the LU name relevant for translation
		x...	1 = NAME2 is the real name 0 = NAME2 is the alias name
		.xxx xxxx	Reserved

Table 25 (Page 2 of 3). Alias-Selection-Input Parameter List

Dec			
(Hex)	**Size**		
Offset	**(Bytes)**	**Comment**	**Description**
43 (2B)	5		Reserved
48 (30)	8	NAME2	LU name to be translated
56 (38)	8	NETFROM2	Network in which NAME2 is known
64 (40)	8	NETTO2	Network that NAME2 is to be translated to. Blank if exit needs to determine the network.
Start of COS/LOG information			
72 (48)	8	COSNAME	COS name
80 (50)	8	LOGNAME	logmode name
88 (58)	8	NETFROM	Network ID in which the name(s) are known
96 (60)	8	NETTO	Network ID to which the name(s) are to be translated
Start of NAME3 information			
104 (68)	1		Indicates the role of the LU in the session
105 (69)	1		Indicates the primary and secondary roles of the OLU and DLU
		xxxx	Indicates whether the LU is the OLU, DLU, or unknown
		 xxxx	Indicates whether the LU is the PLU, SLU, or unknown
106 (6A)	1		Information pertaining to the LU name relevant for translation
		x...	1 = NAME3 is the real name 0 = NAME3 is the alias name
		.xxx xxxx	Reserved
107 (6B)	5		Reserved
112 (70)	8	NAME3	LU name to be translated
120 (78)	8	NETFROM3	Network in which NAME3 is known
128 (80)	8	NETTO3	Network that NAME3 is to be translated to. Blank if exit needs to determine the network.
Start of NAME4 information			
136 (88)	1		LU type. Indicates the role of the LU in the session.
137 (89)	1		Indicates the primary and secondary roles of the OLU and DLU
		xxxx	Indicates whether the LU is the OLU, DLU, or unknown
		 xxxx	Indicates whether the LU is the PLU, SLU, or unknown
138 (8A)	1		Information pertaining to the LU name relevant for translation

Table 25 (Page 3 of 3). Alias-Selection-Input Parameter List

Dec (Hex) Offset	Size (Bytes)	Comment	Description
		x...	1 = NAME4 is the real name 0 = NAME4 is the alias name
		.xxx xxxx	Reserved
139 (8B)	5		Reserved
144 (90)	8	NAME4	LU name to be translated
152 (98)	8	NETFROM4	Network in which NAME4 is known
160 (A0)	8	NETTO4	Network that NAME4 is to be translated to. Blank if exit needs to determine the network.

Table 26 shows the constants for the alias-selection-input parameter list.

Table 26. Constants for the Alias-Selection-Input Parameter List

Type	Value	Description
		Settings for real name or alias name
Bit	0	The name sent for translation is the alias name. The real name needs to be determined.
Bit	1	The name sent for translation is the real name. The alias name needs to be determined.
		Settings for OLU or DLU
Bit	0000	It cannot be determined whether the LU is the OLU or DLU
Bit	0001	The LU is the origin logical unit for this session
Bit	0010	The LU is the destination logical unit for this session
		Settings for PLU or SLU
Bit	0000	It cannot be determined whether the LU is the PLU or SLU
Bit	0001	The LU is the primary logical unit for this session
Bit	0010	The LU is the secondary logical unit for this session
		Settings for LU role
Hex	00	The LU is a session partner
Hex	01	The LU is the primary printer
Hex	02	The LU is the alternate printer

The flags explained in offset 0 (0) (Table 25 on page 34) contain information that is not included in the translate inquiry. One flag indicates whether the NetView alias application is currently active. The second flag indicates whether the NetView alias application was ever active. Use these bits to maintain system integrity.

The parameter list contains sections labeled NAME1, NAME2, NAME3 and NAME4. Each of these sections include information relating to the LU names, for example:

- Which LU name is requesting translation
- Role of the LU
- Role of the PLU or SLU and OLU or DLU
- Whether a real name or alias name is being sent for translation

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

- Whether the network ID that the name is to be translated into is known.

In general, if a name is included in the input list, the exit may translate the name. This rule holds for all names except the DLU name, where additional guidelines apply.

- If the DLU name is an alias name, the exit may translate the DLU name to its real name and provide the owning SSCP name.
- If the DLU name is a real name and VTAM invokes the alias selection function for a request, the exit should not translate the LU name but may return the owning SSCP name. (The function code and related session information indicate whether the exit was driven for a request or response.)
- If the DLU name is a real name and VTAM invokes the alias selection function for a response, then the exit may translate the LU name to the alias name, but should not return an owning SSCP name.

NAME1 is the first name to be translated. NETFROM1 is the network in which the name to be translated (NAME1) is known. NETTO1 is the network to which NAME1 is to be translated. For example:

- NAME1 is APPLB5
- NETFROM1 is NETB
- NETTO1 is NETA.

The above values indicate that APPLB5 is the name known in NETB. APPLB5 in NETB must be translated to the LU name in NETA. If these values are set for the first name, the alias selection function may evaluate the values, and return the translated name in the output parameter list in field RTNNAME1.

COSNAME is the class-of-service name to be translated. LOGNAME is the logon mode name to be translated. NETFROM is the network ID in which either the logmode name, the COS name, or both names are known. NETTO is the network ID to which either the logmode name, the COS name, or both names are to be translated. For example:

- LOGNAME is INTERACT
- NETFROM is NETA
- NETTO is NETB.

INTERACT in NETA is to be translated into the logmode name as known in NETB. The translated logmode name is returned in the output parameter list in RTNLOG.

Note: If any NETTO is unknown, VTAM requests that the network ID be provided. If the network ID is not returned, the translated information is ignored.

The input constants in Table 26 on page 36 are used in the alias-selection-input parameter list. These bit settings indicate:

- Whether the name to be translated is a real name or an alias name
- Whether the LU is the OLU or DLU
- Whether the LU is the PLU or SLU
- The role of the LU.

MVS VM Alias-Selection-Output Parameter List

The output parameter list is where data is returned to VTAM. A name could be requested to be translated, but the network ID into which the name needed to be translated was not known. In this case, the network ID must first be determined. The network ID which was determined by the alias selection function is then used to determine the desired names. For CDINIT, DSRLST and INIT OTHER CD processing for SLU information, the network ID is returned in RTNNET1. For INIT OTHER CD processing for PLU information, the network ID is returned in RTNNET2. If any names were translated, but the network ID was not known, the session fails to setup.

Table 27 shows the alias-selection-output parameter list.

Table 27. Alias-Selection-Output Parameter List

Dec(Hex) Offset	Size (Bytes)	Comment	Description
0 (0)	8	RTNNAME1	The name returned by the user corresponding to NAME1
8 (8)	8	RTNNAME2	The name returned by the user corresponding to NAME2
16 (10)	8	RTNCOS	The translated COS name
24 (18)	8	RTNLOG	The translated logmode name
32 (20)	8	RTNSSCP1	The SSCP name determined by the exit. For CDINIT and DSRLST processing, this field contains the DLU owning SSCP. For INIT OTHER CD processing, this field contains the SLU owning SSCP name.
40 (28)	8	RTNSSCP2	The SSCP name determined by the exit. For CDINIT and DSRLST processing, this field is not filled in. For INIT OTHER CD processing, this field contains the PLU-owning SSCP name.
48 (30)	8	RTNNET1	The network ID if the exit was to determine it. See the explanation above this table.
56 (38)	8	RTNNET2	The network ID if the exit was to determine it. See the explanation above this table.
64 (40)	8	RTNNAME3	The name returned by the user corresponding to NAME3
72 (48)	8	RTNNAME4	The name returned by the user corresponding to NAME4

The returned information must meet the following requirements:

- The real network ID (the network to translate the information into) **must** be returned if it was not sent.
- If a real network ID is returned, and the network to translate the names into was sent, the networks must be the same. The network ID cannot be altered.
- If you indicate that information was translated, but no information was returned, processing continues without updating any information. The return code determines whether the NetView alias application will be invoked.
- The input parameter list cannot be altered.
- Blank values are valid only for the COS and the logmode names.

- LU, NETID, and the owning SSCP names must be padded on the right with blanks.
- If information was needed, and a network ID was returned, this network ID is used for the real name (if it was not known). This network ID is sent to the NetView alias application to determine the rest of the unknown information, if desired.
- All names must contain 1 – 8 characters in the following format:
  - First character: alphabetic (A – Z) or the national characters @, #, or $
  - Second through eighth characters: alphanumeric (A – Z or 0 – 9) or the national characters @, #, or $.

  Note: Alphabetic characters must be in uppercase. The results of their being in lowercase are unpredictable.

If the list returned by the exit routine fails to meet any of these requirements, the request fails, unless otherwise indicated.

The address of the alias parameter list should remain unchanged.

MVS VM Session Flow for Alias Selection Function

VTAM invokes the alias selection function from a variety of places throughout session initiation and DSRLST processing. The following flows indicate when the alias selection function is invoked and for what reason. Keep in mind the following information while reading the flows:

- Unless otherwise noted, the flows assume that the alias selection function does not return any translated names and that no host is running the NetView alias application.
- If the exit returns a translated name, subsequent hosts do not request translation for that name.
- Associated LU names do not appear if the SLU is not using an associated LU table.
- All flows assume that the gateway control is SHARE/SHARE. If the gateway control is ONLY, alias invocations and RU flows vary slightly.

Note: The session flows are numbered. Refer to the respective numbered items following the session flows for an explanation of the flows.

Figure 3 on page 40 shows the SLU-initiated session flow.

SLU-Initiated Session

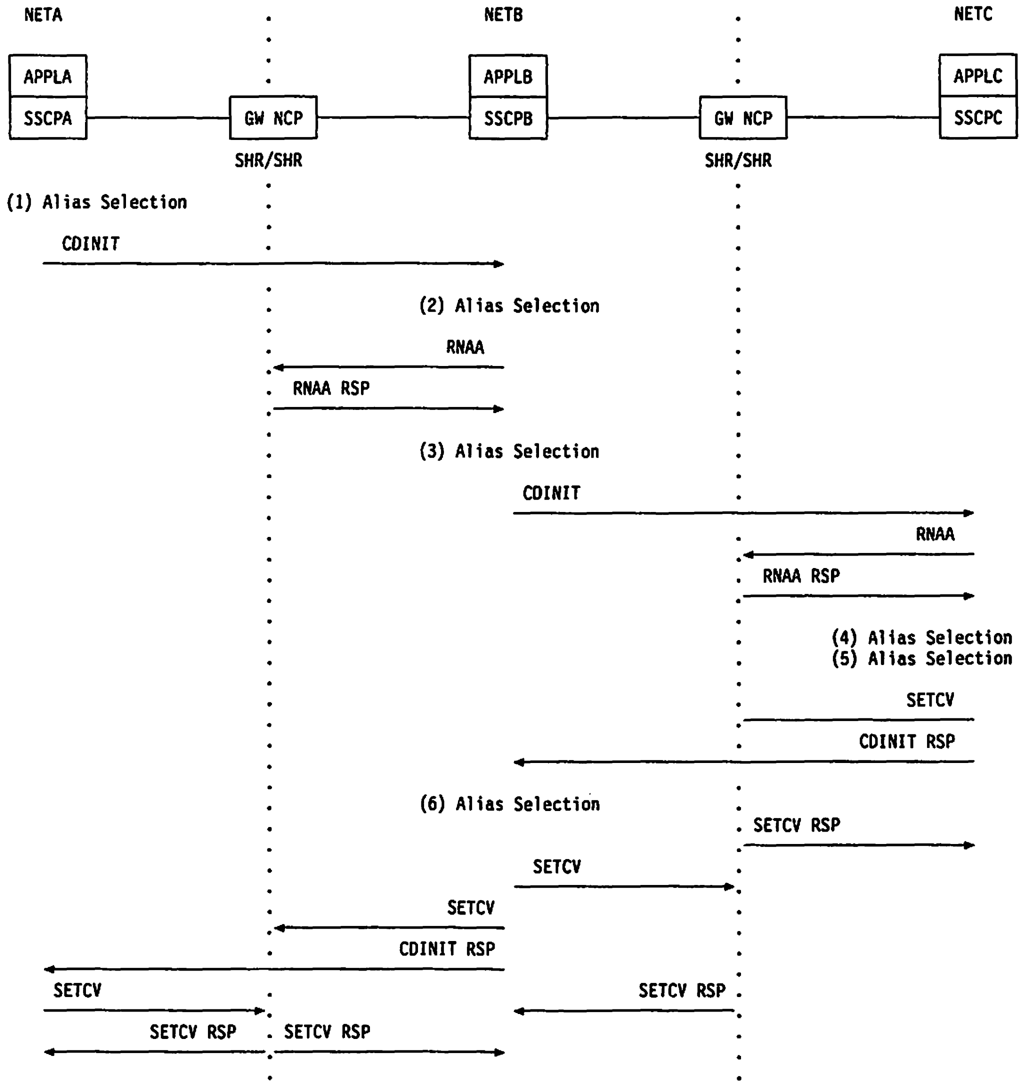


Figure 3. Session Flow for SLU-Initiated Session

1. VTAM passes the DLU alias name, the OLU real name, the associated LU real names, and the LOGMODE and COS names as known in the OLU's network. The alias selection function has the option of returning the DLU real name, the network ID of the DLU real name, the owning SSCP of the DLU, the OLU and associated LU alias names as known in the DLU network, and the LOGMODE and COS names as known in the DLU network.

2. See number 1.

3. Since the session management exit routine has not returned any names, VTAM assumes the DLU network is NETC and the name portion of the DLU real name is the same as the DLU alias name. VTAM requests translations from NETA to NETC for the OLU alias name, the associated LU alias names, the LOGMODE name, and the COS name. VTAM also requests the owning SSCP for the DLU.

4. This invocation is identical to number 3 except that no request is made for the owning SSCP.

5. VTAM drives the session management exit routine to translate the COS name from NETC to NETB.
6. See number 5.

Figure 4 shows the PLU-initiated session flow.

PLU-Initiated Session

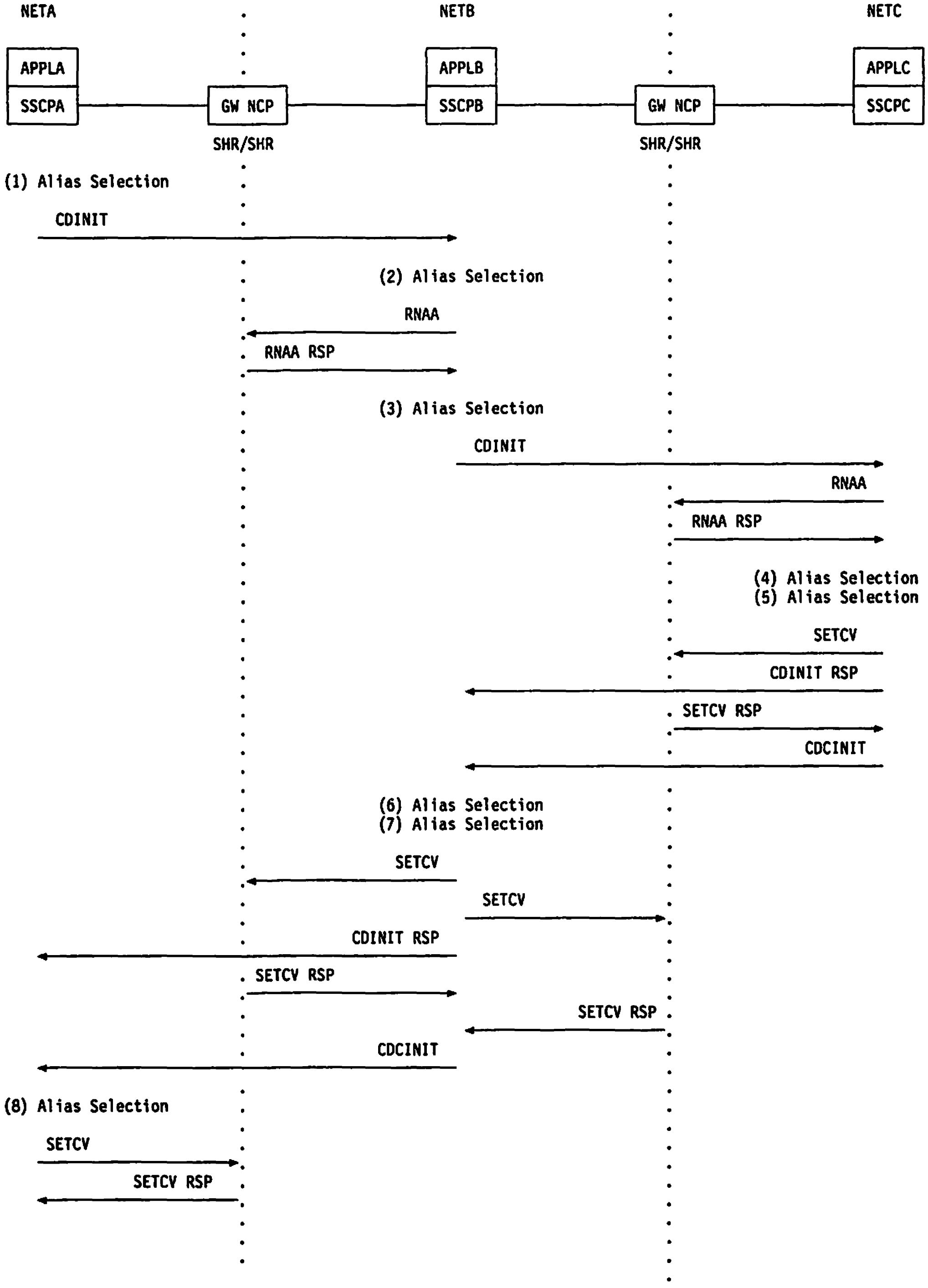


Figure 4. Session Flow for PLU-Initiated Session

1. VTAM passes the DLU alias name, the OLU real name, and the LOGMODE name as known in the OLU's network. The alias selection function has the option of returning the DLU real name, the network ID of the DLU real name, the owning SSCP of the DLU, the OLU alias name, and the LOGMODE name as known in the DLU network.
2. See number 1.
3. Since the session management exit routine has not returned any names, VTAM assumes the DLU network is NETC and the name portion of the DLU real name is the same as the DLU alias name. VTAM requests translations from NETA to NETC for the OLU alias name and the LOGMODE name. VTAM also requests the owning SSCP for the DLU.
4. This invocation is identical to number 3 except that no request is made for the owning SSCP.
5. VTAM drives the session management exit routine to translate the COS name from NETC to NETB, and to translate the associated LU names from NETC to NETA.
6. See number 5.
7. VTAM drives the exit routine to translate the COS name from NETB to NETA. No attempt is made to translate the associated LU names since this host has already attempted to translate them.
8. VTAM drives the exit routine to translate the COS name from NETB to NETA, and to translate the associated LU names from NETC to NETA.

© Copyright IBM Corp. 1984, 1990

Third Party-Initiated Sessions

Figure 5 shows the third-party-initiated session flow.

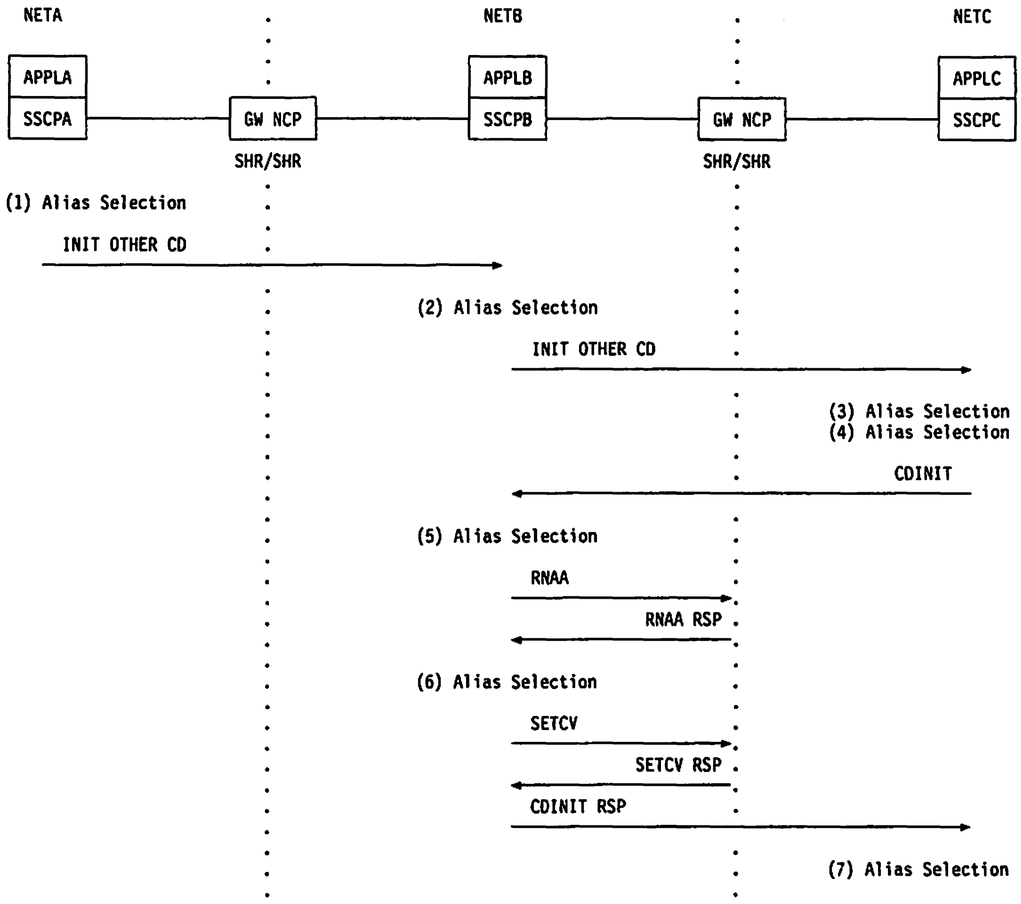


Figure 5. Session Flow for ILU-Initiated Session

1. VTAM passes the PLU alias name as known in NETA. The alias selection function has the option of returning the real name of the PLU, the network ID of the PLU, and the owning SSCP of the PLU. The exit routine may also return the LOGMODE name as known in the SLU's real network.

2. See number 1.

3. See number 1.

4. VTAM passes the DLU alias name, the OLU real name, the associated LU real names, and the LOGMODE and COS names as known in NETC. The alias function has the option of returning the DLU real name, the network ID of the DLU real name, the owning SSCP of the DLU, the OLU and associated LU alias names as known in the DLU network, and the LOGMODE and COS names as known in the DLU network.

5. See number 4.

6. SSCPB assumes the DLU network ID is NETB and finds that it owns the PLU. VTAM requests translations from NETC to NETB for the OLU alias name, the associated LU alias names, the LOGMODE name, and the COS name.

7. VTAM drives the session management exit routine to translate the OLU alias name, the associated LU alias names, the LOGMODE name, and the COS name from NETC to NETB.

Figure 6 shows the ILU-initiated session flow.

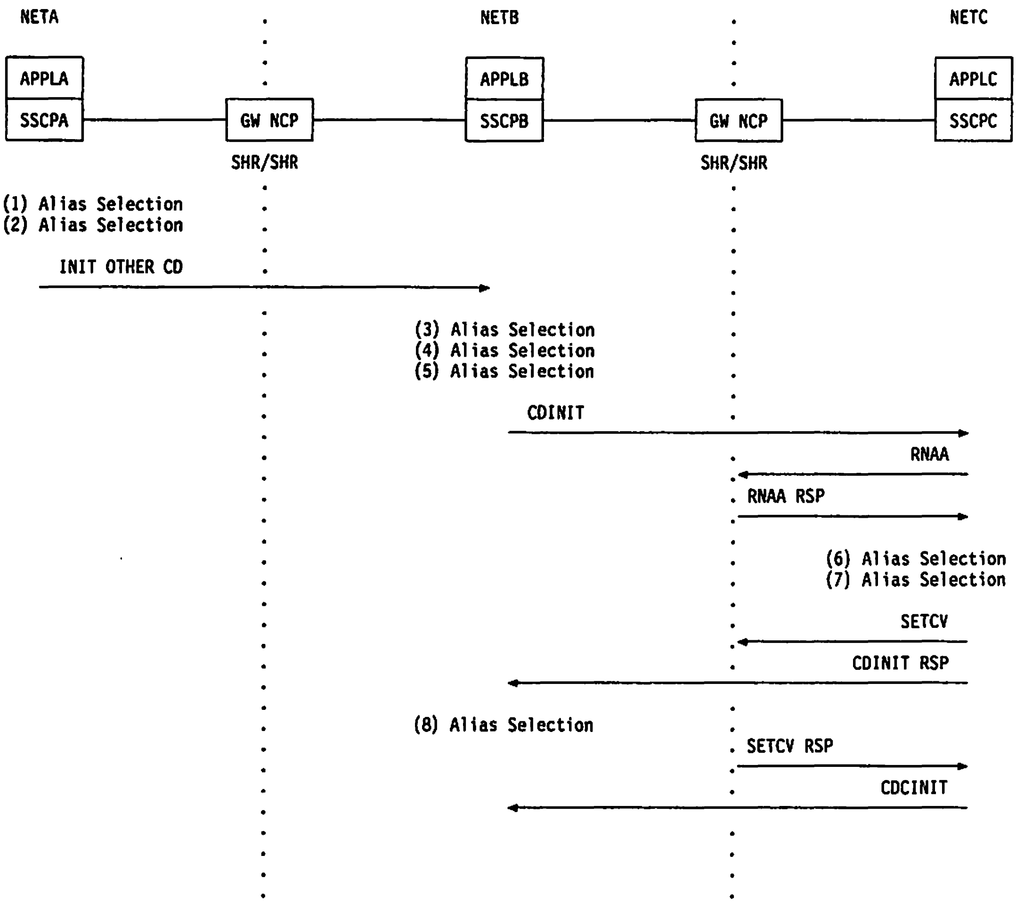


Figure 6. Session Flow for ILU-Initiated Session

1. VTAM passes the PLU alias name as known in NETA. The alias selection function has the option of returning the real name of the PLU, the network ID of the PLU, and the owning SSCP of the PLU. The exit may also return the LOGMODE name as known in the SLU's network. *In this case, assume the alias selection function has returned the PLU real name and network ID.*
2. Since VTAM has learned the PLU real name and already knows the SLU real name, VTAM drives the session management exit routine again asking for the PLU and SLU alias names.
3. VTAM drives the session management exit routine for the LOGMODE name as known in the SLU's network.
4. See number 2.
5. VTAM drives the exit routine to translate the OLU alias and the LOGMODE name from NETB to NETC.
6. See number 5.
7. VTAM drives the exit routine to translate the COS name and associated LU names from NETC to NETA.

8. See number 7.

SLU-Initiated Session with USERVAR Name

Figure 7 shows the SLU-initiated session with USERVAR name flow.

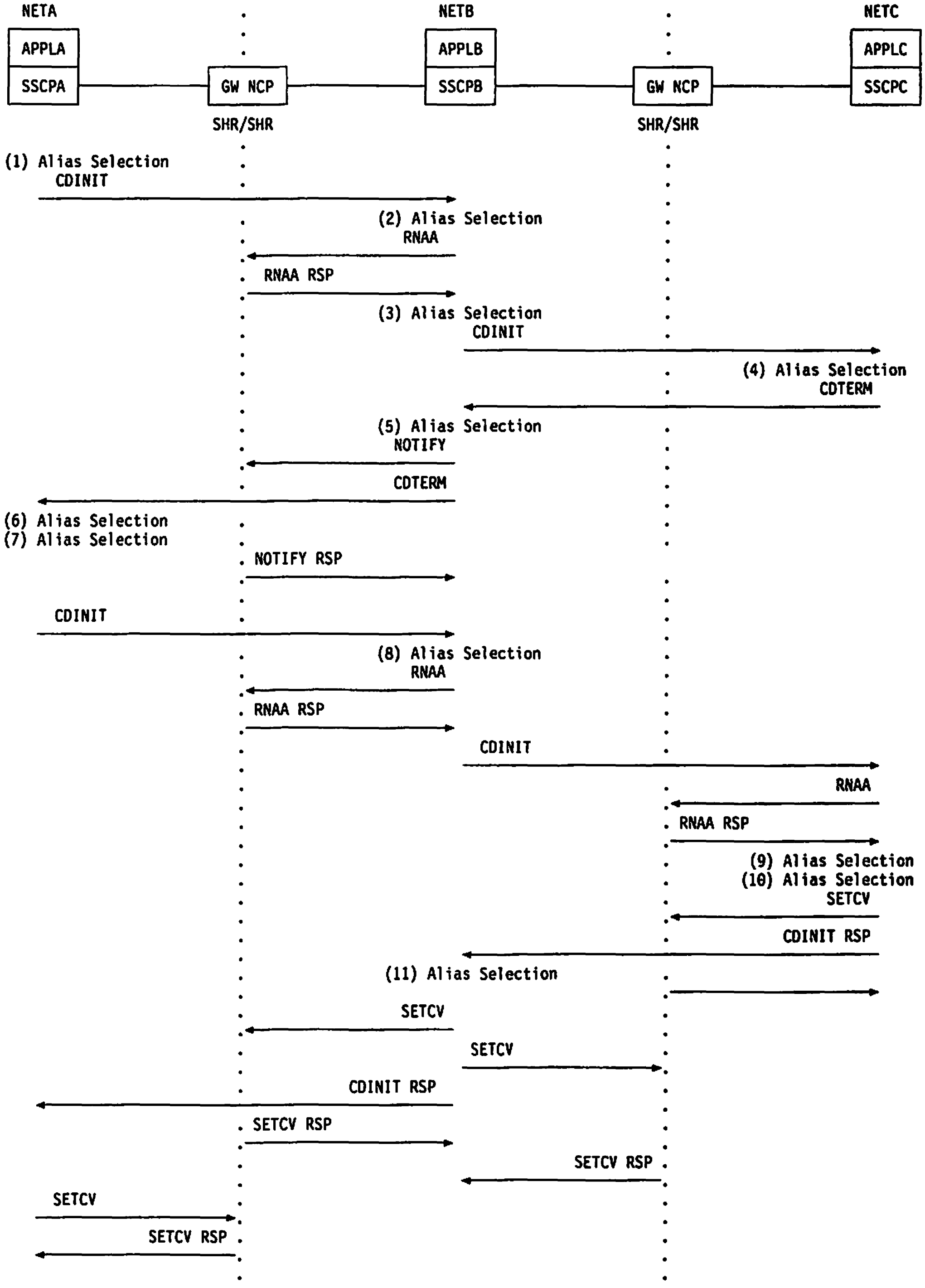


Figure 7. Session Flow for SLU-Initiated Session with USERVAR name

In this example, the APPLA initiates a session with an application named UVNAME. SSCPC knows that UVNAME is a USERVAR for the application named APPLC.

1. VTAM passes the DLU alias name, the OLU real name, and the LOGMODE, COS, and associated LU names as known in the OLU's network. The alias selection function has the option of returning the DLU real name, the network ID of the DLU real name, the owning SSCP of the DLU, the OLU and associated LU alias names as known in the DLU net, and the LOGMODE and COS names as known in the DLU network.
2. See number 1.
3. Since the session management exit routine has not returned any names, VTAM assumes the DLU network is NETC and the DLU real name is the same as the DLU alias name. VTAM requests translations from NETA to NETC for the OLU alias name, the associated LU alias names, the LOGMODE name, and the COS name. VTAM also requests the owning SSCP for the DLU.
4. SSCPC recognizes that UVNAME is a USERVAR for APPLC. SSCPC then drives the exit routine to translate the DLU real name to the DLU alias name.
5. VTAM drives the exit routine to translate the DLU real name to the DLU alias name.
6. See number 5.
7. SSCPA restarts the session with the new application name and drives the session management exit routine to translate the OLU alias name, associated LU alias names, the LOGMODE name, and the COS name from NETA to NETC.
8. VTAM drives the exit routine to translate the OLU alias name, the associated LU alias names, the LOGMODE name, and the COS name from NETA to NETC.
9. See number 8.
10. VTAM drives the exit routine to translate the COS name from NETC to NETB.
11. See number 10.

DSRLST Session Flow

Figure 8 on page 47 shows the session flow for DSRLST.

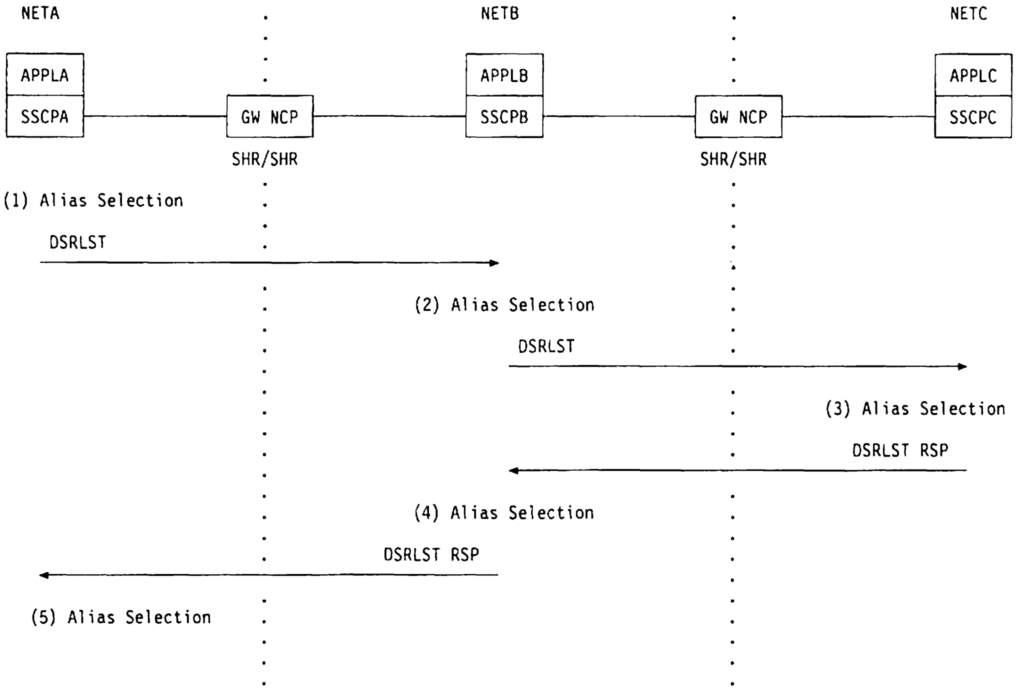


Figure 8. Session Flow for DSRLST

In this example, APPLA has issued an INQUIRE OPTCD=APPSTAT for an application by the name of UVNAME. UVNAME is a USERVAR known in NETC as APPLC.

1. VTAM passes the DLU alias name. The alias selection function has the option of returning the DLU real name, the network ID of the DLU real name, and the owning SSCP of the DLU.
2. See number 1.
3. SSCPC recognizes that UVNAME is a USERVAR for APPLC, returns the real name on the DSRLST RSP, and drives the session management exit routine to translate the DLU real name to the DLU alias name.
4. VTAM drives the exit routine to translate the DLU real name to the DLU alias name.
5. See number 4.

Design Requirements

Follow these procedures when writing this routine:

- Use standard linkage.
- Save registers 0–14.

Likewise, keep in mind the following restrictions when writing this routine:

- The name of the session management exit routine must be ISTEXCAA.
- You should code only conditional GETMAINS in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- MVS All data is addressable only in 24-bit mode.

- The session management exit routine must be reentrant. It must be link-edited into the appropriate library. See "Installing VTAM Exit Routines" on page 66 for more information on installing the correct VTAM exit routine.
- The routine operates as an internal VTAM subroutine. It operates enabled in pageable storage. The routine gets control in supervisor state with a VTAM storage key. Errors in the routine could cause damage to VTAM or system control blocks and modules. VTAM performance is degraded if the routine requires lengthy processing time.
- If your installation permits parallel LU-LU sessions, the session management exit routine must be capable of processing more than one request for the same LU-LU pair. You can distinguish between these sessions using the session ID pointed to by word 6 of the input parameter list.
- Do not invoke any function that causes a system wait to occur, including implied waits for I/O operations. System waits may cause VTAM failure in some timing-dependent situations.
- Do not use VTAM macroinstructions in the routine.
- MVS This exit routine may be above the 16M line. Data is always presented below the line.
- MVS VM All functions, except the end function, run under abend protection. If an error occurs, VTAM issues message IST793E and continues as though the exit routine had not been coded.
- VSE An abend of the session management exit routine causes VTAM to abend. It is best not to execute the abend function in this exit routine. VTAM does not issue message IST793E.

Suggestions: When writing your exit routine, keep in mind that any number of vectors may exist and the vector lengths may be different in different releases of VTAM. Therefore, use the variable-length parameter list indicator to determine the end of the list. In addition, use the length byte to manipulate each vector in the list. If any bytes are added to the vector, or any vectors to the list, you do not have to recode your exit routine.

Virtual Route Selection Exit Routine

A virtual route selection exit routine enables you to modify the ordered list of virtual routes as specified in the COS entry. VTAM then uses your new list of virtual routes to select a route for the session.

You can write a virtual route selection routine, named ISTEXCVR, to be invoked before VTAM establishes any sessions between logical units and before VTAM is terminated (to enable it to cleanup any necessary resources).

VTAM calls the virtual route selection exit routine when a session between a primary logical unit in the VTAM subarea and a logical unit in **another** subarea is about to be established. The virtual route selection exit routine is not called if both logical units reside in the same VTAM subarea. The exit is not called for independent PLUs.

The exit routine is passed only the list of routes within the local network and is scheduled only within the subarea of the primary logical unit.

Initial Register and Parameter List Contents

When this user-written routine gains control, register contents are as follows:

Register 1:	Address of a variable-length parameter list (described in Table 28 on page 51).
Register 13:	Address of the standard 18-fullword save area.
Register 14:	Return address.
Register 15:	Address of the entry point of this routine.

Changing the Virtual Route Selection List

The parameter list that VTAM sends to the virtual route selection exit routine includes pointers to a series of data blocks called *virtual route descriptor blocks.* Each virtual route descriptor block contains the virtual route number and transmission priority for each virtual route that is defined and operative between the origin and destination subareas and within the class of service requested for the session.

The exit routine can modify VTAM's route selection process by changing the descriptor blocks it receives in the following manner:

Note: In the following examples, "(*)Address" means that the high-order bit indicates this is the last entry in the list.

- Reordering the original list

  Example: Virtual route exit routine receives:

  Address of VR1,TP1 descriptor block
  Address of VR2,TP2 descriptor block
  Address of VR3,TP1 descriptor block
  (*)Address of VR4,TP0 descriptor block

  Virtual route exit routine reorders the list and returns to VTAM:

  Address of VR3,TP1 descriptor block
  Address of VR4,TP0 descriptor block
  Address of VR1,TP1 descriptor block
  (*)Address of VR2,TP2 descriptor block

  Now VTAM will attempt to assign the pending session to VR3, then VR4, and so forth.

- Deleting entries from original list

  Example: Virtual route exit routine receives:

  Address of VR1,TP2 descriptor block
  Address of VR2,TP0 descriptor block
  Address of VR3,TP2 descriptor block
  (*)Address of VR4,TP1 descriptor block

  VR exit routine deletes VR2,TP0 from the list and returns to VTAM:

  Address of VR1,TP2 descriptor block
  00 00 00 00 (4 bytes of zeros)
  Address of VR3,TP2 descriptor block
  (*)Address of VR4,TP1 descriptor block

  Now VTAM will attempt to use VR3,TP2 if VR1,TP2 cannot be activated.

- Using only a portion of the list

  Example: Virtual route exit routine receives:

  Address of VR1,TP1 descriptor block
  Address of VR2,TP1 descriptor block
  Address of VR3,TP0 descriptor block
  (*)Address of VR4,TP2 descriptor block

  VR exit routine sets the high-order address bit on in VR2,TP1 and returns to VTAM:

  Address of VR1,TP1 descriptor block
  (*)Address of VR2,TP1 descriptor block
  Address of VR3,TP0 descriptor block
  Address of VR4,TP2 descriptor block

  VTAM uses only VR1,TP1 and then VR2,TP1 to establish the pending session. If neither route can be activated, the session-establishment request is terminated because no paths are available.

- Replacing a virtual route in the list

  Example: Virtual route exit routine receives:

Addresses of Descriptor Blocks		Descriptor Block Contents
VR1,TP2 descriptor block	------>	VR1,TP2.....
VR2,TP1 descriptor block	------>	VR2,TP1.....
VR3,TP0 descriptor block	------>	VR3,TP0.....
(*)VR4,TP0 descriptor block	------>	VR4,TP0.....

To replace the first entry in the virtual route list, for instance, you must overwrite the contents of the appropriate **descriptor block**:

Addresses of Descriptor Blocks		Descriptor Block Contents
(Former) VR1,TP2 descriptor block	------>	VR6,TP1.....
VR2,TP1 descriptor block	------>	VR2,TP1.....
VR3,TP0 descriptor block	------>	VR3,TP0.....
(*)VR4,TP0 descriptor block	------>	VR4,TP0.....

After the replacement, the first address in the list points to the same descriptor block. However, this descriptor block will have been overwritten, and now describes VR6,TP1. VTAM attempts to use VR6,TP1 before using the other virtual routes in the list.

Note that if the virtual route is invalid (defined as being outside the range 0–7 or as having a transmission priority outside the range 0–2) or is not defined between the origin and destination subareas, VTAM continues through the set of virtual routes, attempting to activate the next one in the set (VR2,TP1). Note that you cannot return more virtual route blocks than presented on input. The list can be shortened, but not lengthened.

The virtual route exit routine is **not** scheduled for the following events:

- An SSCP session is established.
- An LU-LU session is established that is confined entirely to the host's subarea. For example, two application programs communicating within the same host, or an application program communicating with a channel-attached terminal.
- A session setup is attempted, and no defined routes, which are operational, exist between the origin and destination subareas. In this case, the request to establish a session is rejected before the exit would have been scheduled.

Table 28 shows the parameter list for the virtual route selection exit routine.

Table 28 (Page 1 of 2). Parameter List for the Virtual Route Selection Exit Routine

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	Address of a 1-byte reason code indicating why the exit routine was scheduled: Hex Code — Reason for Scheduling 00 — To establish an LU-LU session (the first time) 01 — To establish an LU-LU session (not the first time) 02 — MVS To establish an LU-LU session (after an abend to the exit subtask) 03 — Because VTAM is terminating due to HALT (last time) 04 — MVS Because VTAM is terminating
4 (4)	4	Address of a 4-byte user field[1]
8 (8)	4	Address of a 1-byte session information field: Bit — Description 1... .... — Pending session requires a VR that maps to an explicit route with an ER number of 0 (originating in the SLU subarea and terminating in the PLU subarea) .xxx xxxx — Reserved
12 (C)	4	Address of an 8-byte field containing the COS name associated with this pending session establishment request[2]
16 (10)	4	Address of a 12-byte field containing the origin subarea information and LU name[3]
20 (14)	4	Address of a 12-byte field containing the destination subarea information and LU name[3]

Table 28 (Page 2 of 2). Parameter List for the Virtual Route Selection Exit Routine

Dec (Hex) Offset	Size (Bytes)	Description
24 (18)	4	This word and all subsequent words will each contain the address of a virtual route descriptor block[4]

Notes:

1. The user field is initially set to zero. The exit routine may use this field (for example, to store the address of a dynamically obtained storage area). VTAM preserves the contents of this user field and resupplies it to the exit routine the next time it is driven.
2. This field is padded on the right with blanks as necessary to make a total of 8 characters.
3. This field is 12 bytes long and contains a 4-byte subarea number in binary, followed by an 8-byte LU name in EBCDIC.
4. The last word of the parameter list is indicated by the setting of the high-order bit (byte 0, bit 0) to 1.

Table 29 shows the format of the virtual route descriptor block.

Table 29. Format of the Virtual Route Descriptor Block

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	1	VR number (0–15)
1 (1)	1	Transmission priority (0–2)
2 (2)	1	VR status Hex 01 — VR not active Hex 02 — VR is active
3 (3)	1	Reserved
4 (4)	2	Current[1] number of LU-LU sessions between the origin and destination subareas using the given VR number and transmission priority
6 (6)	2	Current[1] number of LU-LU sessions between the origin and destination subareas using the given VR number regardless of priority
8 (8)	2	Current[1] number of all sessions between the origin and destination subareas using the given VR number and transmission priority
10 (A)	2	Current[1] number of all sessions between the origin and destination subareas using the given VR number regardless of priority.

Note: [1]Current as of the queuing of the virtual route selection request to the virtual route selection exit routine subtask.

Design Requirements

Follow these procedures when writing the virtual route selection exit routine:

- Use standard linkage.
- Save registers 0–14.

Likewise, consider the following restrictions when writing the exit routine:

- You should code only conditional GETMAINs in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- MVS All data is addressable only in 24-bit mode.
- The exit routine must be reentrant.
- MVS VM This routine runs under a VTAM subtask to permit the exit routine to perform any necessary I/O or other processing without affecting the VTAM main task.
- VSE The routine runs under the VTAM main task.

The exit routine will execute in the problem state with VTAM's storage key.

MVS VM The exit routine can set up its own ESTAE (extended specify task abnormal exit) environment for recovery. See *MVS/Extended Architecture Supervisor Services and Macro Instructions* or *VM/SP Group Control System Macro Reference* for more information. A SYNC interface will be used between VTAM and the exit routine, so that the VTAM module invoking the exit routine will be isolated from any ESTAE environment established by the exit routine.

VSE An abend of the virtual route selection exit routine causes VTAM to abend.

MVS VM Abend Processing

If scheduled as a result of an abend of the subtask under which the exit routine executes, the exit routine is responsible for any cleanup or reopening of data sets that may be required.

If a dynamic storage address was stored in the user field (second word of the parameter list [see Table 28 on page 51]), the address is not valid after the abend is processed.

An abend indication may occur either for a normal route selection request or for the last time the exit routine is driven during VTAM termination. Also, if the subtask under which the exit routine runs suffers an abend during "first time" route selection, and the exit routine has not yet returned to VTAM, the next time the exit routine is driven will still be considered the "first time" (reason code hex 00). This is not true if the next time the exit routine is driven is for VTAM termination, in which case reason code hex 04 is indicated. In any case, modifications made by the exit routine to the user field prior to an abend are preserved for use by the exit routine in any cleanup or recovery actions it might perform the next time it is driven.

To summarize the implications of the above reason code specifications:

Hex 00:	Driven for route selection (first time)
Hex 01:	Driven for route selection (nth time – normal)
Hex 02:	Driven for route selection (previous time resulted in abend)
Hex 03:	Driven for VTAM termination (last time)

Hex 04: Driven for VTAM termination (after abend).

If the exit routine subtask abnormally terminates, VTAM reattaches the subtask unless the subtask has exceeded an abend threshold. If the abend threshold is exceeded, VTAM uses the default list originally supplied to the exit routine. The session setup undergoing route selection at the time of an abend fails. Other queued selection requests are presented to the reattached exit.

Final Register Contents

The routine must leave the register status as follows:

Registers 0 – 14:	Restore these registers.
Register 15:	Must be set to zero. If a nonzero value is returned, VTAM no longer schedules the exit routine.

VR Pacing Window Size Calculation Exit Routine

This exit routine specifies the bounds for virtual route pacing windows. A virtual route pacing window represents the quantity of path information units (PIUs) that can be transmitted on a virtual route before a virtual route pacing response is received. This response indicates that the virtual route receiver is ready to receive more PIUs on the route. The exit routine is called when a virtual route is activated. It returns the minimum and maximum values for the window of the virtual route.

The IBM-supplied algorithm for window size calculation works with the route pacing algorithm used in the network. It is appropriate for most installations and configurations. VTAM calculates the minimum and maximum sizes of virtual route pacing windows based on the link protocol and the explicit route length (that is, the number of transmission groups in the explicit route used by the virtual route). While the virtual route is being used to transmit data, subarea nodes along the route request that the endpoints adjust the window sizes within the minimum and maximum limits according to traffic conditions along the route.

However, after tuning VTAM and analyzing traffic patterns and resource capabilities, you may want to choose your own bounds or code a replacement routine to set the window sizes to different values than the ones supplied by IBM. This exit routine is appropriate for systems where the number of resources could vary considerably from one day to the next. For example, you may find one or more resources whose capacities are not consistently used. This may warrant increasing the window sizes. Decreasing the window sizes is less likely to be useful, since network-flow control protocols are designed to prevent congestion, and setting window sizes too small could reduce traffic flow considerably.

The logic for calculating the maximum and minimum window sizes for virtual route pacing is contained in the VTAM exit routine named ISTPUCWC.

Note: You can **avoid coding** the VR pacing window size calculation exit routine by specifying default minimum and maximum window sizes using the VRPWS*nn* operand on the PATH definition statement. For more information on specifying virtual route pacing window sizes on the PATH definition statement, see *VTAM Resource Definition Reference.*

For most routes, VTAM sets the minimum window size to the explicit route length and the maximum size to 3 times the explicit route length. However, if the virtual route ends in a subarea that is adjacent to VTAM, the maximum window size is set to the greater of 15 or 255—16*n*, where *n* is the number of explicit routes (defined or operative) originating in the host and passing through and not ending in the adjacent subarea. This increases the maximum window size for a route to a channel-attached NCP that has only a few explicit routes passing through it.

The exit routine is called when VTAM is preparing to activate a virtual route. VTAM passes to it the following information:

- The particular virtual route number and its transmission priority.
- The number and length of the explicit route that is associated with the particular virtual route.
- The subarea addresses of the destination and adjacent subareas.
- The number of defined and operative explicit routes that pass through, but do not end in, the adjacent subarea.
- The address of the explicit route characteristics table, which describes the transmission group traversed by the explicit route associated with a virtual route. This table determines the link protocol being used between the host and the adjacent node on the explicit route. (See Table 31 on page 58 for the format of this table.)

The following sections provide information you need to replace or modify this routine. See *VTAM Network Implementation Guide* for details.

Initial Register and Parameter List Contents

When the routine gains control, register contents are as follows:

Register 1: Address of parameter list (described in Table 30).

Register 13: Address of a standard 18-fullword save area for use by this routine.

Register 14: Return address.

Register 15: Address of the entry point of this routine.

Table 30 (Page 1 of 2). VR-Pacing Window-Size-Calculation Exit Routine Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
0 (0)	4	VR identifier: Byte 0 = Hex 00 Byte 1 = Hex 00 Byte 2 = VR number Byte 3 = Transmission priority
4 (4)	4	Explicit route number
8 (8)	4	Destination subarea number
12 (C)	4	Adjacent subarea number
16 (10)	4	Explicit route length (this length is equal to the total number of transmission groups in the explicit route)
20 (14)	4	Total number of defined or operative explicit routes that pass through, but do not end in, the given adjacent subarea

Table 30 (Page 2 of 2). VR-Pacing Window-Size-Calculation Exit Routine Parameter List

Dec (Hex) Offset	Size (Bytes)	Description
24 (18)	4	Address of explicit route characteristics table

Design Requirements

Follow these procedures when writing the routine:

- Use standard linkage.
- Save and restore registers 2–14.

Likewise, take into account the following restrictions when writing the routine:

- The routine operates as an internal VTAM subroutine. Therefore, VTAM performance may be degraded if the routine requires lengthy processing time. While this routine is being executed, no new VTAM operator requests or requests to initiate or terminate sessions are processed by VTAM for any application program. System waits, including implied waits for I/O operations, should be avoided. Also, be aware that a program check causes VTAM to abend.
- You should code only conditional GETMAINs in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- MVS All data is addressable only in 24-bit mode.
- This routine operates enabled in pageable storage. The routine gains control in supervisor state and with a supervisor storage key, so errors within the routine could cause damage to VTAM or system control blocks and modules.

Final Register Contents

The routine must leave the register status as follows on return:

Register 0: Minimum VR window size (if register 15=0).

Note: This value must be greater than zero and less than or equal to the maximum value specified in register 1.

Register 1: Maximum VR window size (if register 15=0).

Note: This value must be greater than or equal to the minimum value specified in register 0 and less than or equal to 255.

Registers 2–14: Restore these registers.

Register 15: Return code of zero if the minimum and maximum window size values are to be used by VTAM. Any nonzero return code (or invalid values specified in register 0 or 1) causes VTAM to ignore the values indicated by registers 0 and 1. Instead it uses a minimum window size equal to the ER length and a maximum window size of three times the ER length or the minimum and maximum window sizes that you defined on the PATH definition statement, if any.

For more information on specifying virtual route pacing window sizes all the PATH definition statement, see *VTAM Resource Definition Reference.*

MVS Using the VR Window Size Calculation Exit Routine for IMS

Because of the specialized use of channel-to-channel virtual routes by the IMS resource lock manager (RLM), you may find it useful to modify the VTAM virtual route window size calculation exit routine (ISTPUCWC). The RLM in one IMS system sends one message at a time to a correspondent IMS RLM, and waits for a message in response. With this level of message traffic on a virtual route, the virtual route window size algorithm does not indicate that a demand exists to increase the window size. Thus, in the absence of other message traffic flowing on the virtual route, the window size tends to stay at its minimum value. While this situation does not impede throughput on the virtual route, it tends to increase the number of machine instructions executed by VTAM for each IMS transaction (because VTAM processes a virtual route pacing response for every, or nearly every, IMS RLM message).

Accordingly, if a channel-to-channel virtual route between two IMS systems is not also used for non-IMS message traffic, the following changes may be made to the module to reduce the number of virtual route pacing responses sent and processed by VTAM in each of the IMS hosts:

1. Insert code to recognize the destination subarea address, virtual route number, and transmission priority that you have defined for use between the two IMS systems.
2. Then, insert code so that when the module recognizes such a virtual route, it sets the minimum window size to a value of 6.
3. The maximum window size may be computed in the normal manner, but the output of that calculation must be checked against this new minimum and must not be allowed to fall below it. That is, if the maximum window size is calculated to be less than 6, the maximum should then be set to 6.

Notes:

1. The source code for ISTPUCWC is not provided with the VTAM program. You must obtain it from microfiche.
2. You can **avoid coding** the VR pacing window size calculation exit routine for IMS by allocating IMS-to-IMS communications to a specific virtual route and transmission priority with a unique class of service. This enables you to use the VRPWSxx operand.

There are two possible ways to compute the maximum window size in the IBM-supplied version of the module. One is used when the destination subarea is the same as the adjacent subarea, and the other is used in all other cases. Both parts of this logic should be copied into any new section of code added to compute window sizes for IMS RLM routes. Again, the maximum window size values computed must be adjusted, if necessary, to be at least equal to the minimum window size selected.

These changes to the module are **not** recommended if the IMS RLM virtual route is shared with other virtual route users. The message traffic of the other users should keep the operating window size values at acceptable levels. Table 31 on page 58 shows the characteristics of the explicit route.

Table 31. Explicit-Route Characteristics Table

Comments	Offset	Size (Bytes)	Description
	0	1	Control block identifier (X'4C')
	1	1	Reserved
	2	2	Length of this table
The transmission group entry consists of the following data:			
	0	1	Control block identifier (X'4D')
	1	1	Length of entry
	2	1	Not used
	3	1	DLC protocol: X'01' SDLC X'02' channel-to-communication controller X'03' channel-to-channel X'04' LAN attachment

Session Accounting Exit Routine

You should use the session management exit routine rather than an accounting exit routine to gather accounting information, since the session management routine allows you to combine session-related functions into one exit routine. The session management exit routine also fully supports cross-network and takeover processing. For example, information passed to the accounting exit routine for cross-network sessions may be misunderstood since LU names are not necessarily unique. The session management exit routine's accounting function avoids this problem since additional LU information is provided to the exit in the PLU and the SLU resource information control vectors. (See "Session Management Exit Routine" on page 3 for more information on its capabilities.)

You can write a session accounting exit routine to collect statistics on the number of times that sessions start and end, so that the users can be charged accordingly.

The session accounting exit routine is scheduled only in SSCPs that are in the same domain as one of the LUs. If both the accounting and session management exit routines are provided within the LU's domain, VTAM schedules both exits. (See "Session Management Exit Routine" on page 3 for more information on its capabilities.)

IBM does not supply a skeleton session accounting exit routine. If you want this routine, you must code it in its entirety. You can use any previously written accounting exit routine without changes for VTAM Version 3. If you do not code this routine, all session accounting information is discarded.

Initial Register Contents

When this user-written routine gains control, register contents are as follows:

Register 0: Hex 00000000 if a session has been established; hex FFFFFFFF if a session has been terminated.

Register 7: Address of the doubleword containing the name of the PLU.

Register 11: Address of the doubleword containing the name of the SLU.

Register 13: Address of the save area for use by this routine.

Register 14: Return address.

Register 15: Address of the entry point of this routine.

Design Requirements

Follow these procedures when writing this routine:

- Use standard linkage.
- Save registers 0 – 14.

Likewise, consider the following restrictions when writing this routine:

- The name of the module must be ISTAUCAG.
- You should code only conditional GETMAINs in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- MVS All data is addressable only in 24-bit mode.
- The accounting exit provides the two LU names involved in the session. You can write this information to System Management Facility (or an equivalent facility), along with the time of day. From these records, you can determine the session connection time. This exit does not have access to the logon data, nor to any VTAM or TSO control blocks.
- The routine operates as an internal VTAM subroutine. Therefore, performance is degraded if the routine requires lengthy processing time. While this routine is being executed, no requests to establish or terminate a session are processed by VTAM. System waits, including implied waits for I/O operations, should be avoided.
- This routine operates enabled in pageable storage. Because the routine operates at VTAM's main task dispatching priority, there is a possibility of lockout if a wait requires another task action. The routine gets control in supervisor state with a VTAM storage key, so errors within the routine could cause damage to VTAM or system control blocks and modules.
- Do not use VTAM macroinstructions in the routine.
- The routine is notified as part of an LU-LU session setup and takedown. Design the routine to process requests involving only LU-LU sessions.
- If an application program is capable of establishing parallel sessions, the exit routine must be capable of processing more than one request from the same LU-LU pair.

Final Register Contents

All general-purpose registers, except register 15, must be restored. No return code is expected by VTAM.

Session Authorization Exit Routine

You should code the session management exit routine rather than an authorization exit routine to authorize sessions, since the session management routine handles both same- and cross-network sessions and allows you to combine session-related functions into one exit routine. (See "Session Management Exit Routine" on page 3 for more information on its capabilities.)

The session authorization exit routine checks on or restricts the use of an application program or other LU.

For same-network sessions, VTAM calls the session authorization routine whenever it receives a request to establish a session between two LUs. For example, it may be called as the result of a logon from a terminal, an automatic logon, or a VTAM operator logon. It is called for both the initial and all subsequent logons to a controlling application (that is, one to which the terminal is logged on automatically).

For cross-domain sessions, the authorization exit routine in each domain (the domain of the PLU and of the SLU) is scheduled whenever a session is in the process of being established.

The session authorization exit routine is scheduled only in SSCPs that are in the same domain as one of the LUs. For cross-network sessions, this means that any information provided to the authorization exit routine could be misunderstood, since an LU name is not necessarily unique. You can avoid this problem by using the initial and secondary authorization functions of the session management exit routine instead of the session authorization exit routine.

If both the session authorization and session management exit routines exist within the LU's domain, the authorization exit routine is scheduled first. If it authorizes the session, the session management exit routine is scheduled; otherwise, the session management exit routine is not scheduled and session setup is rejected.

You might code the routine to contain a table of valid sessions against which the session-establishment request can be compared. For example, you can design an application program to establish a session with any LU, using the OPNDST OPTCD=ACCEPT macroinstruction in its LOGON exit routine. The authorization exit routine could compare the identity of any LU that attempts to establish a session with the application program to entries in such a table to determine whether authorization can be granted for that LU. For example, a particular LU could be authorized only at a particular time of day.

IBM does not supply a skeleton session authorization routine. If you want this routine, you must code it in its entirety. You can use any previously written authorization routine without changes for VTAM Version 3. If you do not write an authorization exit routine, all sessions are authorized. The following sections explain how to code this routine.

© Copyright IBM Corp. 1984, 1990

Initial Register and Parameter List Contents

When VTAM passes control to this routine, register contents are as follows:

Register 1:	Address of a parameter list
Register 13:	Address of the save area for use by this routine
Register 14:	Return address
Register 15:	Address of the entry point of this routine.

Table 32 on page 62 lists the contents of the parameter list that VTAM passes to the routine. The macroinstructions and options referred to are described in *VTAM Programming*. The commands are described in *VTAM Operation*.

Design Requirements

Follow these procedures when writing this routine:

- Use standard linkage.
- Save registers 0 – 14.

Likewise, consider the following restrictions when writing this routine:

- The name of the module must be ISTAUCAT.
- You should code only conditional GETMAINs in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- MVS All data is addressable only in 24-bit mode.
- The routine operates as an internal VTAM subroutine. Therefore, performance is degraded if the routine requires lengthy processing time. While this routine is being executed, no new session-establishment, session-termination, or activation requests are processed by VTAM. System waits, including implied waits for I/O operations, should be avoided.
- This routine operates enabled in pageable storage. Because the routine operates at VTAM's main task dispatching priority, there is a possibility of lockout if a wait requires other task action. The routine gets control in supervisor state and with a VTAM storage key, so errors within the routine could cause damage to VTAM or system control blocks and modules.
- Do not use VTAM macroinstructions in the routine.
- This routine must not modify the parameter list pointed to by register 1 (described in Table 32 on page 62). Neither can any field pointed to from the parameter list be modified.
- The routine must supply a return code to VTAM in register 15. A return code of zero authorizes the session to be established. Any nonzero return code means that the request is not authorized. If the request is not authorized, VTAM informs the session initiator.

Table 32. Parameter List for the Authorization Exit Routine

Dec (Hex)	Size	
Offset	(Bytes)	Description
0 (0)	1	Reserved
1 (1)	1	Contains a decimal number 1–6 that identifies the request: **No.** **Request** 1 Primary LU initiated session (SIMLOGON or OPNDST macroinstruction with ACQUIRE option) 2 Reserved 3 Reserved 4 Operator initiated session (Logon request initiated by the network operator by issuing a VARY ACT command or VARY LOGON command) 5 Third party initiated session (CLSDST macroinstruction with PASS option). (See offset 8 for pointer to name of application program for which logon is requested.) 6 Secondary LU initiated session (REQSESS macroinstruction)
2 (2)	2	Reserved
4 (4)	4	0 or address of doubleword with name of the LU that issued the request. (Not applicable if offset 1 is 4 or 5.)
8 (8)	4	0 or address of doubleword with name of the application program acting as the primary end of the session.
12 (C)	4	Identifies the SLU associated with request. This field contains the address of the doubleword containing the LU name.
16 (10)	2	X'0001'

Failure of the authorization exit routine to honor requests of the type discussed below may cause VTAM to authorize sessions that should not be authorized.

VTAM invokes the exit routine after the initiate request for the session is received by the SSCP.

If an application program is capable of establishing parallel sessions, the exit routine must be capable of processing more than one request from the same LU-LU pair.

For cross-domain session-initiation requests, the authorization exit routines in each host (the host owning the PLU and the host owning the SLU) are called during processing of either the initiate or cross-domain initiate requests.

Final Register Contents

The routine must leave the register status as follows:

Registers 0 – 14:	Restore these registers.
Register 15:	Return code of zero if the request is authorized. Any nonzero return code if the request is not authorized.

VSE TPRINT Processing Exit Routine

You can write an exit routine to be invoked whenever the trace print utility (TPRINT) has selected a record to be formatted. This routine can be used for:

- Formatting trace records (other than the formatting provided by TPRINT).
- Discarding unnecessary trace records.
- Discarding unnecessary information in trace records.
- Performing data analysis or problem determination. You can analyze application program data flows as a whole or on particular sessions; time stamping of TRFILE records provides a resolution of 1/1000 second.

There is no restriction on I/O functions you might want to perform in this exit routine. You can, for example, communicate with the system operator (SYSLOG) to obtain processing options. Or you might open a VTAM access method control block (ACB) and communicate with a terminal user for entry of options and printing of formatted trace data.

Note: The TPRINT exit routine can operate as a subtask of VTAM or as part of TPRINT when TPRINT is operating as a separate job step. When it is a subtask of VTAM, it can degrade VTAM performance if it requires lengthy processing time caused, for example, by excessive looping. This potential problem can be reduced by executing TPRINT as a separate job in a lower-priority partition.

The following sections provide information you need to write this routine.

Initial Register and Parameter List Contents

When this user-written routine gains control, register contents are as follows:

Register 0:	0, indicates first entry after record selection. 1, indicates subsequent entries for more output after return code of 8.
Register 1:	Address of a parameter list (described in Table 33).
Register 13:	Address of the save area for use by this routine.
Register 14:	Return address.
Register 15:	Address of the entry point of this routine.

The parameter list is shown in Table 33 on page 64.

Table 33. TPRINT-Processing Parameter List

Offset	Size (Bytes)	Description
0	4	Address of trace record (ISTTRAB)
4	4	Address of 121-byte output buffer
8	4	Address of formatted date and time stamp for this record: *yy.ddd/hh:mmth*

Design Requirements

Follow these procedures when writing this routine:

- Use standard linkage.
- Save registers 0 – 14.

Likewise, consider the following restrictions when writing this routine:

- The name of the module must be ISTRAEUE.
- You should code only conditional GETMAINs in this routine. You can reduce the possibility of VTAM abending during a storage shortage by coding conditional GETMAINs.
- This routine operates enabled in pageable storage. It gains control in problem state with the user's protection key. The routine is executed under the invoking program's task, which can be the VTAM task.
- When the TPRINT exit routine operates as a subtask of VTAM, it can degrade VTAM performance if the routine requires lengthy processing time, for example, if it includes excessive looping. This potential problem can be reduced by executing TPRINT as a separate job in a lower-priority partition.
- This routine must not modify the parameter list pointed to it by register 1 (the 12-byte area described in Table 33). Neither can the fields pointed to by word 1 (ISTTRAB) or word 3 (time stamp) be modified.
- The routine must supply a return code to the calling routine in register 15. See "Final Register Contents" for a detailed description of these return codes. In addition, if the return code is 4 or 8, the output buffer is assumed to contain a SYSLST record with a valid ASA control character in the first byte.
- Do not code any VSE macroinstructions that are invalid for a subtask (for example STXIT). Do not use STXIT AB because TPRINT establishes its own AB exit which must not be prevented from executing.

Final Register Contents

The routine must leave the register status as follows:

Registers 0 – 14: Restore these registers.

Register 15: Valid return codes:

0, indicates skip this record, get the next record.

4, indicates print output, get the next record.

8, indicates print output, return the same record for more output.

12, indicates terminate TPRINT (no SYSLST record).

Any other return codes indicate that standard editing is to be performed.

Example of a TPRINT Exit Routine

The flowchart in Figure 9 shows the logic for a simple TPRINT exit routine. The function of this routine is to format internal trace (VIT) records only and to permit TPRINT to perform standard formatting on all other trace records. Options initialized at label SPECIFY on the flowchart might be used, for example, to select (at SELECT) those types of VIT records to be formatted by this routine.

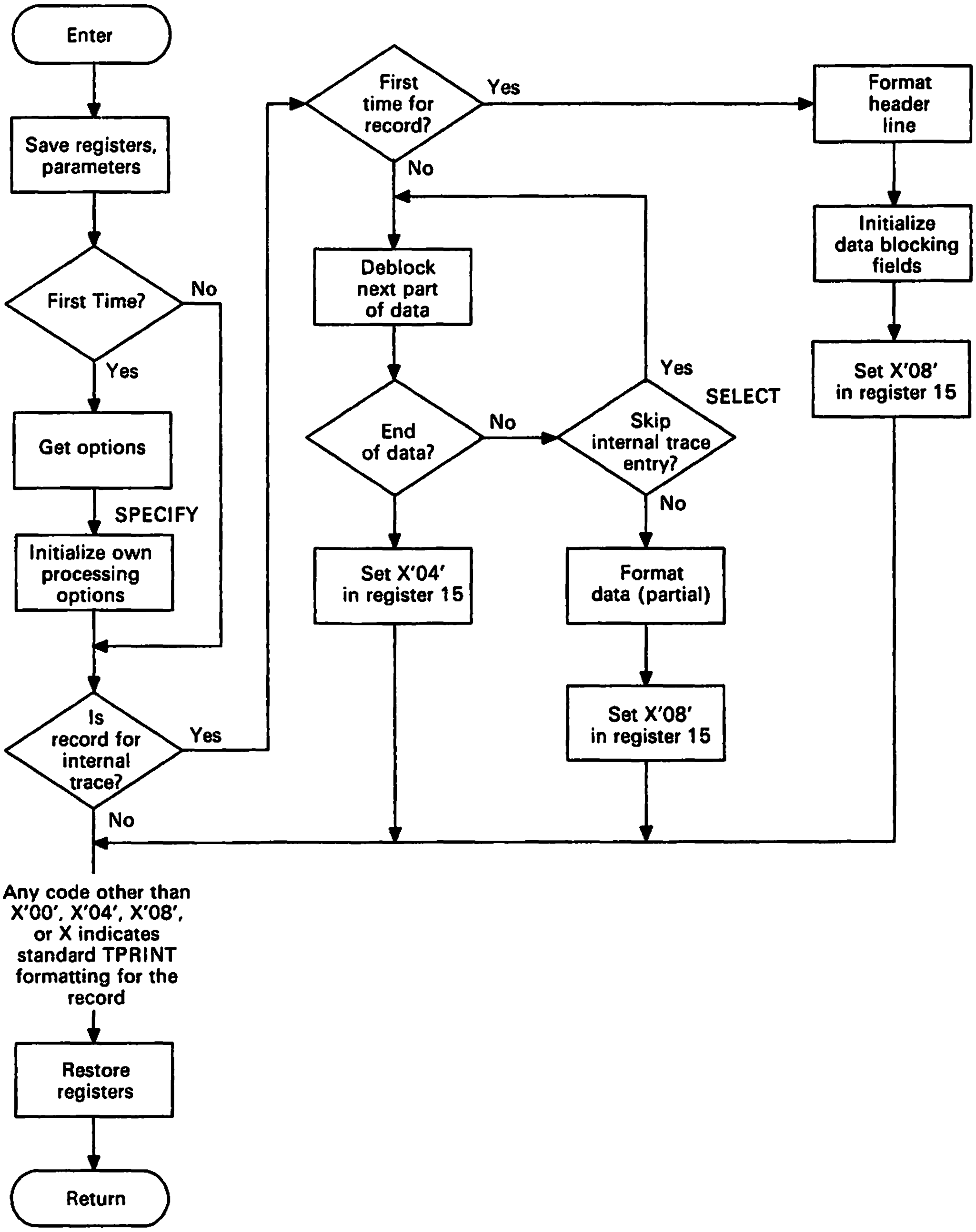

Enter
Save registers, parameters
First Time?
No
Yes
Get options
SPECIFY
Initialize own processing options
Is record for internal trace?
Yes
No
First time for record?
Yes
No
Format header line
Initialize data blocking fields
Set X'08' in register 15
Deblock next part of data
End of data?
No
Yes
SELECT
Skip internal trace entry?
No
Set X'04' in register 15
Format data (partial)
Set X'08' in register 15
Any code other than X'00', X'04', X'08', or X indicates standard TPRINT formatting for the record
Restore registers
Return

Figure 9. Example of Logic for a TPRINT Exit Routine

Installing VTAM Exit Routines

Except for the TPRINT processing exit routine, follow these steps to install the appropriate VTAM exit routines:

1. Assemble the routines.
2. Link-edit the routines to the appropriate VTAM library:
   MVS VM Use the libraries indicated in Table 34.
   VSE Use a private definition library.

Table 34. MVS and VM Libraries for VTAM Exit Routines

Exit Routine	MVS Library	VM Library
Session management exit routine	SYS1.VTAMLIB	VTAMUSER LOADLIB[1]
Virtual route selection exit routine	SYS1.VTAMLIB	VTAMUSER LOADLIB[1]
Virtual route pacing window size calculation exit routine	SYS1.VTAMLIB	VTAMUSER LOADLIB[1]
Accounting exit routine	SYS1.LPALIB[2]	DCSS[3]
Authorization exit routine	SYS1.LPALIB[2]	DCSS[3]

Notes:

1. VM See "Installing Tables and Modules in VM" on page 89 for information on using VMFLKED to link-edit these routines.

2. If you place your user-written exit routine in SYS1.LPALIB, re-IPL to reformat the link-pack area. For details, see *MVS/XA System Programming Library: Initialization and Tuning*.

3. VM If your user-coded exit routine is in the discontiguous shared segment (DCSS), replace the TEXT deck for the exit routine (on the VTAM BASE disk) with the new copy and rebuild VTAM.

VSE To install the TPRINT processing exit routine, catalog the routine (ISTRAEUE) and link-edit it with the TPRINT routine into the library where VTAM is installed.

VM/ESA VTAM exit routines can be written to run in AMODE 31 or AMODE 24 for migration. VTAM calls the routine in the proper AMODE based on the link-edit attributes of the exit routine load module.

The link-editing must be done before VTAM is started. The names of the user-written routines must be those shown in Table 35.

Table 35. VTAM Module Names for Exit Routines

Exit Routine	VTAM Module Name
Session management exit routine	ISTEXCAA
Virtual route selection exit routine	ISTEXCVR
Virtual route pacing window size calculation exit routine	ISTPUCWC
Accounting exit routine	ISTAUCAG
Authorization exit routine	ISTAUCAT

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Defining User Modules and Tables

Module for VTAM Constants 69
  Modifying the Constants Module 72
  Description of Constants 72
CNM Routing Table 85
  Installing the CNM Routing Table 85
  Structure of the CNM Routing Table 86
Logon-Interpret Routine Requirements 88
Installing Tables and Modules in VM 89

Chapter 2. Defining User Modules and Tables

This chapter discusses how to create or modify the CNM table and VTAM modules. It also discusses installing tables and modules in VM and logon-interpret routine requirements.

Module for VTAM Constants

VTAM provides a module, ISTRACON, that contains constants used to control functions that do not usually warrant operator modification.

You can modify a number of VTAM functions using ISTRACON. These functions are discussed under the following headings:

- "Buffer-Use (SMS) Trace Snapshot Value" on page 72
- "Maximum RU Size for Sessions" on page 72
- "BSC 3270 Timeout (Inoperative) Value" on page 73
- "VR Selection Subtask Interval Analysis Block" on page 73
- "Adjacent SSCP Message Control" on page 74
- "Alias Name Translation Control" on page 74
- "Switched Subarea Disconnect Timeout" on page 75
- "Host IRN Slowdown Storage" on page 75
- "NetView Trace Buffers" on page 75
- "Maximum SSCP Rerouting Count" on page 75
- "Directory Size of Symbol Resolution Table for the Host Network" on page 76
- "MVS VM Directory Size of Symbol Resolution Tables for Other Networks" on page 78
- "MVS Process Scheduling Services Trace Control" on page 79
- "Size of the Host Node Table (HNT)" on page 80
- "Size of the CID Index Table" on page 80
- "EAS Value for VTAM's Application" on page 80
- "Maximum Subarea Number" on page 81
- "Size of Data Used by VTAM in Dumping or Loading a Remote NCP" on page 81
- "Session Awareness (SAW) Buffer Limit" on page 82
- "Missing Interrupt Handling for a CTC" on page 83
- ' VM VSE ES/9370 Token-Ring Adapter Buffer Size" on page 83
- "VM VSE ES/9370 Token-Ring Adapter Transmit Buffer Maximum" on page 83
- "VM VSE ES/9370 Token-Ring Adapter Transmit Buffer Minimum" on page 83
- "Option to Receive Dumps for INOPs" on page 83
- "Option to Prevent MDR Recording" on page 84

- "Individual IST530I/IST532I Message Pair Limit" on page 84.
- "MVS/ESA SAW Data Space Limit Factor" on page 84
- "MVS/ESA SAW Data Space Packing Factor" on page 84
- "MVS/ESA PIU Data Space Limit Factor" on page 84

Table 36 summarizes **all** of these constants, showing their defaults as well as their offsets within the module.

Table 36 (Page 1 of 2). Default IBM Constants Module (ISTRACON)

Dec Offset	Hex Offset	Field Name	Default Value	Description
2–3	2–3	RACBSNAP	1000	Number of requests for VTAM buffers between snapshot dumps
4–5	4–5	RACMCPBF	4096	Maximum RU size for all SSCP sessions
6–7	6–7	RACMLUBF	6144	Maximum RU size for LU-LU sessions
8–11	8–B	RACINOPT	X'0000011E'	Timeout value for a BSC 3270
12–21	0C–15			Reserved
22–23	16–17	RACABCNT	X'0004'	Number of times subtask can abend within the time interval
24–27	18–1B	RACABINT	X'00001AD2'	Time interval during which maximum number of abends can occur
28	1C	RACSSMSG	X'01'	Defines how adjacent SSCP table related messages are issued
29	1D	RACALIAS	X'FF'	Used to control alias name translations
30–31	1E–1F	RACSSDTO	30 seconds	Switched subarea disconnect timeout
32–35	20–23	RACINNBL	X'00000000'	Defines maximum size of virtual storage area used to store host IRN transmissions
36	24	RACPDBFS	X'02'	Maximum buffers to build for a NetView request
37	25	RACVCNT	10	Maximum SSCP rerouting count
38–39	26–27	RACHSRT	1499	Number of entries in host SRT directory
40–41	28–29	RACONSRT	43	MVS VM Number of entries in SRT directories in other networks VSE Reserved
42	2A	RACTRFLG	X'00'	MVS Enables the generation of trace entries which show the interaction between the MVS dispatcher and VTAM process scheduling services
43	2B			Reserved
44–47	2C–2F	RACHNTSZ	4080 MVS VM 2024 VSE	Size of the host node table (HNT)

Table 36 (Page 2 of 2). Default IBM Constants Module (ISTRACON)

Dec Offset	Hex Offset	Field Name	Default Value	Description
48–51	30–33	RACCITSZ	4080 MVS VM 2024 VSE	Size of the CID index table and of the CONVT table (both)
52–55	34–37	RACEAS	3000'MVS VM/ESA 50 VM VSE	Default EAS value for VTAM's application
56–59	38–3B	RACSASUP	511	Maximum subareas supported in the network
60–77	3C–4D			Reserved
78–79	4E–4F	RACBUFSZ	512	Size of request unit used by VTAM in taking a static dump or loading a remote NCP
80–83	50–53	RACMXBUF	500 VM VSE 2000 VM/ESA 10000 MVS	Maximum number of session awareness (SAW) buffers allowed
84–87	54–57	RACMIHTM	1800	INOP the CTC at reply-timeout plus this many tenths of a second
88–89	58–59	RACLBFSZ	112	VM VSE ES/9370 Token-ring adapter buffer size
90	5A	RACLTXMX	28	VM VSE ES/9370 Token-ring adapter transmit buffer maximum count
91	5B	RACLTXMN	14	VM VSE ES/9370 Token-ring adapter transmit buffer minimum count
92	5C	RACINOP	0	Indicates whether or not a dump should be generated whenever an INOP occurs
93	5D	RACSMDR	0	Indicates whether VTAM should collect SDLC statistical MDRs in LOGREC
94–95	5E–5F			Reserved
96–99	60–63	RACNTWRE	X'7FFFFFFF'	Individual IST530I/IST532I message pair limit
100–103	64–67	RACHXSRT	0	MVS VM/ESA Number of entries in host SRT directory
104–107	68–6B	RACXNSRT	0	MVS VM/ESA Number of entries in SRT directories in other networks
108–111	6C–6F	RACSAWLM	100	MVS/ESA SAW data space limit factor
112–115	70–73	RACSAWPK	0	MVS/ESA SAW data space packing factor
116–119	74–77	RACPIULM	200	MVS/ESA PIU data space limit factor

Modifying the Constants Module

The VTAM constants module may be modified by replacing values within the executable module. To do this, "zap" ISTRACON's executable module within the appropriate VTAM library:

MVS SYS1.VTAMLIB
VM The VTAMUSER LOADLIB
VSE A private definition library referenced in the VTAM start procedure,

using the appropriate system service aid:

MVS The AMASPZAP program
VM The ZAP command
VSE The MSHP patch function.

Whenever ISTRACON's values are modified, VTAM must be restarted before the new values are used.

Description of Constants

The following sections describe each of the VTAM constants in detail, indicating which operating systems they are found in.

Buffer-Use (SMS) Trace Snapshot Value

ISTRACON field name: RACBSNAP

If the SMS trace is active, VTAM takes a snapshot dump after a specified number of requests occur for VTAM buffers. While more frequent dumps may enable you to more accurately determine buffer use, frequent dumps may degrade system performance.

The IBM-supplied default number causes a snapshot dump to be taken after every 1000 (hex 3E8) requests. You can modify this number by changing the RACBSNAP field in ISTRACON.

Specify a number from 0 to 32767 (hex 7FFF). If zero is specified, a snapshot dump is taken after every request (that is, it is equivalent to specifying 1).

Maximum RU Size for Sessions

ISTRACON field names: RACMCPBF and RACMLUBF

You can set a maximum length for RUs in LU-LU sessions and in SSCP (SSCP-PU, SSCP-LU, and SSCP-SSCP) sessions. For LU-LU sessions, the maximum RU lengths can be changed using parameters in the BIND request and can be specified through the appropriate logon mode table entry. In the absence of such specification, VTAM uses a default value. VTAM uses another default value for SSCP sessions. It is these default values that you can change in the constants module. The defaults are 4096 for an SSCP session and 6144 for an LU session.

The default values should be adequate in most cases. However, if they are too small, devices that send RUs larger than the default value are deactivated. The default values should be increased if your installation has logon mode table entries for devices that do not specify the maximum RU size and the devices may send larger RUs than the default.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

BSC 3270 Timeout (Inoperative) Value

ISTRACON field name: RACINOPT

VTAM keeps track of the number of general poll failures that occur for a BSC 3270 over a specific time interval. If two failures occur within the time interval, VTAM treats this as an unrecoverable failure of the terminal and deactivates it. If the second failure occurs outside the time interval, VTAM starts the poll again, and an operator message is sent to report the failure, even though recovery is automatically initiated.

The pattern of responses to general polls in your installation could require you to change the time interval. For example, if a small number of operator messages report polling failures, this might suggest that the terminal responds frequently enough to general polls so that it would not be efficient to deactivate the terminal. If two of these failures occurred within the specified time interval, however, the device would be deactivated unnecessarily. In this case, you might want to decrease the time interval, thus decreasing the chance that the terminal is unnecessarily deactivated.

Conversely, if general poll failures outside the time limit are frequently reported, you might want to increase the time interval, so that the device is deactivated earlier and corrective action can be initiated.

The IBM-supplied default for the time interval is approximately 300 seconds. Thus, if two failures occur within 300 seconds, the 3270 is deactivated. By changing RACINOPT, you can specify a different time interval.

The value in RACINOPT is specified in units of time. Each unit of time is 1.048576 seconds. The default is hex 11E units of time, or approximately 300 seconds. The value can range from 0 to 2147483647 (hex 84E1FFFF). If zero is specified, the device is never deactivated as a result of a general poll failure.

VR Selection Subtask Interval Analysis Block

ISTRACON field names: RACABCNT and RACABINT

The first field used (at offset hex 16) indicates the number of times the subtask can abend within the time interval. This value may be specified in the range of 0–32767 (hex 7FFF). If zero is specified, there is no limit; that is, VTAM continues to reattach the subtask as often as it abends.

The field at offset hex 18 is the time interval in which the maximum number of abends can occur. This interval is specified in units of 0.262144 seconds. The default is approximately 30 minutes.

The time interval may be specified in the range of 0–2147483647 (hex 84E1FFFF). If 0 is specified, VTAM does not consider the time interval in deciding whether or not to reattach the subtask. For example, if the count is 10 and the time interval is 0, VTAM stops reattaching the subtask after 10 abends.

Ordinarily, VTAM reattaches the subtask each time it abends. If the number of abends exceeds the maximum within the time interval, however, VTAM no longer reattaches the subtask.

If these constants are not appropriate to your needs, you may want to change them. However, you should use low enough values to avoid a possible abend loop, where an abend is caused by a programming error within your routine.

Adjacent SSCP Message Control

ISTRACON field name: RACSSMSG

This constant controls the display of messages that are generated whenever session initiation requests to or from adjacent SSCPs fail. It may not be desirable to have adjacent SSCP-related messages issued in every SSCP. These messages are not "duplicates" and, consequently, cannot be suppressed by the use of VTAM's duplicate message ID table.

IST894I, IST895I and IST896I may be controlled using the RACSSMSG constant.

These messages are issued when a session initiation fails because trial and error routing using an adjacent SSCP table has exhausted the table. All adjacent SSCPs were tried but the destination LU was not located by any of the SSCPs.

The following values may be specified for the RACSSMSG constant:

Hex Value	Description
00	Do not issue adjacent SSCP messages in any SSCP
01	Issue adjacent SSCP messages in SSCP (OLU) only
02	Issue adjacent SSCP messages in all SSCPs.

The default value is hex 01.

Any value of the RACSSMSG constant specified other than hex 00, hex 01, or hex 02 results in an **error** default value of hex 00 being used.

The specified value is only meaningful if the host's role is OLU.

Alias Name Translation Control

ISTRACON field name: RACALIAS

This constant controls alias name translation.

The following bit values may be specified for the RACALIAS constant:

Bit Value	Description
x...	If ON, determine alias names
.x..	If ON, determine real names
..x.	If ON, determine owning SSCP name
...x	If ON, determine COS names
.... x...	If ON, determine logmodes
.... .x..	If ON, determine associated LU alias names
.... ..11	Reserved.

The default value is hex FF.

Valid values are hex 03–FF. Any invalid value of the RACALIAS constant specified results in the default value of hex FF being used.

Switched Subarea Disconnect Timeout

ISTRACON field name: RACSSDTO

This parameter indicates the number of seconds that can pass, without a new LU-LU session beginning, before VTAM automatically disconnects the switched PU. The default value is 30 seconds.

For more on automatic disconnection, see *VTAM Network Implementation Guide*.

Host IRN Slowdown Storage

ISTRACON field name: RACINNBL

This constant defines the maximum size, in bytes, of the virtual area in VTAM storage that can save host intermediate routing node (IRN) transmissions. This pageable data area is used only when an adjacent subarea is not accepting traffic (for example, an NCP that has gone into slowdown mode and is not receiving data). Any host IRN traffic for that subarea is moved from fixed I/O buffers to the VTAM pageable buffers.

If the IBM-supplied default of zero is used, the amount of storage is unlimited.

NetView Trace Buffers

ISTRACON field name: RACPDBFS

This constant specifies the number of buffers to build for a session monitor request. The valid range is 2–255. The default value is 2.

Maximum SSCP Rerouting Count

ISTRACON field name: RACVCNT

The maximum SSCP rerouting count is the maximum number of SSCPs on a specific path that is searched for the destination SSCP before alternate paths are tried. If all paths are tried without finding the destination SSCP, the RU is not rerouted and VTAM sends a negative response. This constant affects the rerouting of both session initiations (CDINIT) and direct search routing (INQUIRE APPSTAT).

For example, assume that this constant has been changed to 3. In Figure 10 on page 76, a CDINIT is sent from SSCP1 to SSCP2. SSCP2 decrements the rerouting count to 2 and passes the request to SSCP3, which decrements the count to 1 and passes the request to SSCP4. At SSCP4 the count is decremented to zero, and the request can go no further on that particular path; it cannot be rerouted to SSCP6. SSCP4 then sends a negative response to the CDINIT, and SSCP3 increments the rerouting count by one, allowing the request to be routed to SSCP5. As was the case at SSCP4, the rerouting count at SSCP5 is decremented to zero, and a negative response is returned. The rerouting count is again incremented to 1 at SSCP3. SSCP3 can continue to reroute the CDINIT if other potential paths exist.

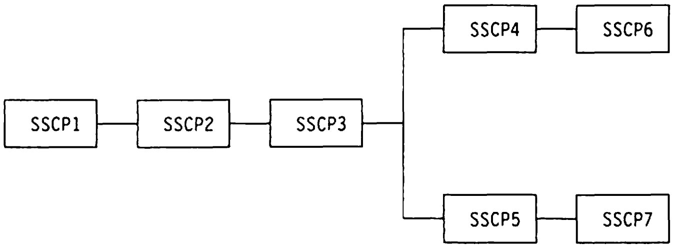


Figure 10. Example of SSCP-Rerouting-Count Use

Thus, you would reduce this count primarily to reduce potential logon time and storage use due to unnecessary adjacent SSCP rerouting. You should review the level of CDINIT and INQUIRE APPSTAT rerouting required for each network environment carefully, and change the SSCP rerouting count accordingly.

The IBM-supplied default value is 10. You can change the rerouting count by specifying a number from 0 to 255 (hex FF). Specifying 0 is equivalent to specifying 1. A maximum SSCP rerouting count of 1 prevents the gateway-capable[2] SSCP that receives the cross-network RU from rerouting that RU to another SSCP.

For session initiation, a maximum of one rerouting can occur within a given network. At least every other rerouting must cross a network boundary.

For other methods of resolving the problem of excessive rerouting, see the discussion of the adjacent SSCP selection function of the session management exit routine on page 15.

Directory Size of Symbol Resolution Table for the Host Network

ISTRACON field name: RACHSRT (MVS VM/ESA RACHXSRT)

The symbol resolution table (SRT) consists of an internal VTAM table used primarily to find information about control blocks. For a VTAM started as gateway-capable and interconnected with other networks, the SRT is several tables: one for the host network and one for each network known by VTAM. Each symbolic name has two parts. The first is a network ID (one assigned to each network) that finds an SRT directory. After the SRT directory is found, the second part of the symbolic name points to the address of a queue of SRT entries.

This field specifies the number of queue pointers in the SRT directory for the network containing the VTAM host node. The IBM-supplied default value is 1499. You can change the number of SRT entry queues by specifying a number from 0 to 32767 (hex 7FFF). If you specify 0, VTAM uses the default value of 1499.

Note: For networks with a large number of LUs, increasing this number shortens the length of the queues, thereby decreasing the logon time.

[2] To be gateway-capable, an SSCP must be started with both NETID and SSCPNAME specified. GWSSCP=YES must also be specified.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

MVS VM/ESA RACHXSRT allows a 31-bit value for the number of SRT entry queues. If it contains a nonzero value, it is used instead of RACHSRT.

There are several facts you should consider before changing the default directory size:

1. Using a prime number of queue pointers results in a fairly even distribution of SRT entries to the queues. It is recommended that you choose a prime number.
2. Besides the SRT entries for the names of network resources, add SRT entries to the host network's SRT for the following:
   - Network ID
     - Add one SRT entry for each network known to this VTAM.
   - Class-of-service table
     - Add one SRT entry for each COS table associated with this host's network by COSTAB operands in active gateway NCPs. That is, add one to your count of SRT entries for every COSTAB operand specified on the BUILD and NETWORK macroinstructions with this host's NETID.
     - Add one SRT entry for this host's COS table.
   - Adjacent SSCP table
     - Add one SRT entry for each destination SSCP in this host's network that is specified in an active adjacent SSCP table.
     - Add one SRT entry for this network's SSCP list.
     - Add one SRT entry for the default SSCP list.
   - Alias name
     - Add one SRT entry for each alias LU name that this VTAM host recognizes.
   - Autologon
     - Add one SRT entry for each SSCP that controls at least one LU that is designated as the PLU for a PLU-SLU automatic logon.
   - Network address
     - Add one SRT entry for each network address assigned to nodes in this VTAM domain.
     - Add one SRT entry for the real network address of each cross-domain resource that establishes a session through this VTAM SSCP.
     - Add one SRT entry for the alias network address of each cross-network resource that establishes a session through this VTAM SSCP.
     - Add one SRT entry for each alternate gateway path to other-network SSCPs (if the SUBAREA operand is specified on the GWPATH statement).

- NetView trace requests
  - Add one SRT entry for each resource for which there may be pending NetView trace requests outstanding at any one time.

    This SRT entry represents a NetView trace request that is pending for a resource that is not yet defined to VTAM. An example of this kind of undefined resource is a dynamically defined cross-domain resource for which no session is active at the time of the trace request. To determine how many entries may be needed for these pending traces, you should know how the NetView program is used in your installation, and be aware of the kinds of resources in your configuration that may be undefined to VTAM at the time of a NetView trace request.

    Note: This value should represent the maximum number of resources for which there may be pending NetView trace requests outstanding at any one time, not the total number of resources for which a NetView trace may be requested.
- PCID
  - Add one SRT entry for each LU-LU session.
- Physical unit services control block

  For each active connection to a physical unit channel-attached to this VTAM host:

  - Add one SRT entry for the link.
  - Add one SRT entry for the station.

  In addition, if the PU is a subarea node (for example, a communication controller or a channel-to-channel attachment to a host),

  - Add one SRT entry for that subarea node.

MVS VM Directory Size of Symbol Resolution Tables for Other Networks

ISTRACON field name: RACONSRT (MVS VM/ESA RACXNSRT)

This field specifies the number of queue pointers in every SRT directory for networks other than this VTAM network (that is, all networks with a NETID other than the NETID of this VTAM specified on the BUILD and NETWORK macroinstructions which are included in the NCP deck). The IBM-supplied default value is 43. You can change the number of SRT entry queues in each directory by specifying a number from 0 to 32767 (hex 7FFF). If you specify 0, VTAM uses the default value of 43. It is recommended that you use a prime number of queue pointers.

Note: For networks with a large number of LUs, increasing this number shortens the length of the queues, thereby decreasing the logon time.

MVS VM/ESA RACXNSRT allows a 31-bit value for the number of queue pointers in every SRT directory for networks other than this VTAM network. If it contains a nonzero value, it is used instead of RACONSRT.

The following are the SRT entry types in the SRT tables for each network:

- Class-of-service table
  - Add one SRT entry for each COS table associated with another network by COSTAB operands in active gateway NCPs. That is, add one to your count of SRT entries for every COSTAB operand specified on the BUILD and NETWORK macroinstructions with NETIDs other than that of this host.
- Adjacent SSCP table
  - Add one SRT entry for each destination SSCP in another network specified in an active adjacent SSCP table.
  - Add one SRT entry for each default adjacent SSCP list for the other networks.
- Network address
  - Add one SRT entry for the real network address of each cross-network resource that establishes a session through this VTAM SSCP.
  - Add one SRT entry for the alias network address of each host-network resource that establishes a session through this VTAM SSCP.
  - Add one SRT entry for each alternate gateway path to other-network SSCPs (if the ADJNETSA operand is specified on the GWPATH statement).
- SSCPID
  - Add one SRT entry for each other-network SSCP that has a session with this VTAM SSCP.
- PCID
  - Add one SRT entry for each LU-LU session.
- Nodes
  - Add one SRT entry for each active CDRSC in another network.

MVS Process Scheduling Services Trace Control

ISTRACON field name: RACTRFLG

The trace entries that show the interaction between the MVS dispatcher and the process scheduling services (PSS) component of VTAM are normally not generated when using the VTAM internal trace (VIT) with OPTION=PSS or OPTION=ALL specified. The zappable constant RACTRFLG enables all or some of these entries to be optionally generated during VIT processing. The VTAM internal trace must have OPTION=PSS or OPTION=ALL active to allow the optional entries to be generated.

The following values may be specified for the RACTRFLG constant:

Value	Description
X'00'	Do not include IRB or SRB entries in VTAM internal trace
X'40'	Include IRB entries in VTAM internal trace
X'80'	Include SRB entries in VTAM internal trace
X'C0'	Include IRB and SRB entries in VTAM internal trace.

The default value is hex 00.

Any value of the RACTRFLG constant specified other than hex 00, hex 40, hex 80, or hex C0 results in the default value of hex 00 being used.

Size of the Host Node Table (HNT)

ISTRACON field name: RACHNTSZ

This field determines the maximum size of the host node table used to find element addresses. Each entry in this table represents an element address of the host. When all entries in a table are filled, a chain is created to another block of entries. The larger the table size, the less chaining there is and the better the performance. However, storage may be used less efficiently if larger tables are used. The minimum for correct VTAM activation is 136 bytes. If a lower value is specified, 136 is used. The default value is MVS VM 4080 and VSE 2024.

Size of the CID Index Table

ISTRACON field name: RACCITSZ

This field determines the maximum size of the following index tables, used to find control blocks as indicated:

ISTCIT Function management control blocks (FMCBs). Each entry in this table represents the location of an FMCB that represents the session identified by the CID value. The minimum for correct VTAM activation is 28 bytes. If a lower value is specified, 28 is used.

ISTCONVT Conversation control blocks (RABs). Each entry in this table represents the location of an RAB that represents the conversation identified by the CID value.

When all entries in one of these tables are allocated, a chain is created to another block of entries. The larger the table size, the less chaining there is and the better the performance. However, storage may be used less efficiently if larger tables are used.

The default value is MVS VM 4080 and VSE 2024.

EAS Value for VTAM's Application

ISTRACON field name: RACEAS

This field determines the maximum number of SNA network addressable units and channel-attached non-SNA terminals that may be active at the same time. If the actual total number of SNA network addressable units and channel-attached non-SNA terminals in the VTAM system is greater than the number specified here, or if the actual number is greater than 8080, the VTAM path length is increased.

The valid range is 0—32767. The default value is MVS 3000 and VM VSE 50.

For more information on EAS values, see the *VTAM Network Implementation Guide*.

Maximum Subarea Number

ISTRACON field name: RACSASUP

This field determines the maximum subarea number supported by a network to which a host running VTAM is interconnected.

The valid values for this field are:

- 255
- 511
- 1023
- 2047
- 4095
- 8191
- 16383
- 32767
- 65535.

If the value coded is not one of the values listed above, VTAM:

- Assumes 65535 if the value is greater than 65535
- Assumes 255 if the value is less than 255
- Rounds up to the next higher supported value if the invalid value is between 255 and 65535.

Size of Data Used by VTAM in Dumping or Loading a Remote NCP

ISTRACON field name: RACBUFSZ

This field determines the request unit size (I/O *bufsize* minus the size of the SNA header) used by VTAM when taking a static dump of a remote NCP or when loading one. The default size is 512 bytes, but VTAM can also handle sizes of 1024 (hex 400) or 2048 (hex 800) bytes.

The range of allowable values for this field is determined by calculating the result of:

```
(MAXBFRU X UNITSZ) - 34 bytes
```

for **each** of the NCPs on the path between the host and the remote NCP being dumped or loaded. The value you choose for this field must be less than the lowest number in this series of results.

Notes:

1. This constant does not apply to a MODIFY LOAD. The length of the IPLTEXT RU is always 2048 plus the length of the SNA header.
2. This constant applies only to static dumps or loads of 3720, 3725, and 3745 Communication Controllers. VTAM ignores this constant when dumping or loading a 3705 and uses a default data size of 512 bytes.
3. This constant is equivalent to the start option NCPBUFSZ.

Session Awareness (SAW) Buffer Limit

ISTRACON field name: RACMXBUF

This field sets the maximum number of buffers used for session awareness (SAW) buffers. To determine whether you need to change this constant, use the following method:

1. First, note the following information:
   - The size of the buffer header (*buffer-header-size*) is 48.
   - The buffer size (*buffer-size*), as you defined it through the NetView program.
   - The number of physical units (*number-of-PUs*), as defined on the NCP.
   - The number of dependent logical units (*number-of-LUs*), as defined on the NCP.
   - The maximum SAW record size (*SAW-record-size*):
     - For SSCP-PU or SSCP-LU sessions is 133
     - Within a single network is 235
     - For cross-network is 370.
2. Then, compute the average SAW record size:
   a. The number of SSCP SAW records is obtained by adding the number of physical units to the number of logical units:

   ```
   SSCP-SAW-records = Number-of-PUs + Number-of-dependent-LUs
   ```

   b. The number of LU-LU SAW records is equal to the number of LUs:

   ```
   LU-LU-SAW-records = Number-of-LUs
   ```

   c. The total number of SAW records is obtained by adding the number of SSCP SAW records to the number of LU-LU SAW records from the last two calculations:

   ```
   Total-SAW-records = SSCP-SAW-records + LU-LU-SAW-records
   ```

   d. The average SAW record size is derived from the following formula:

   ```
   Avg-SAW-record-size =
        ((SSCP-SAW-records × 133) +
        (same-network LU-LU-SAW records × 235) +
        (cross-network LU-LU-SAW records × 370)) ÷ Total-SAW-records)
   ```

3. Compute the number of SAW records in a SAW buffer. To do this, subtract the size of the buffer header from the size of the buffer, then divide this difference by the average SAW record size; round down the answer obtained from this division:

   ```
   Records-per-buffer =
      Round-down((Buffer-size - Buffer-header-size) ÷ Avg-SAW-record-size)
   ```

4. Finally, determine the maximum number of buffers used when taking down an NCP by using the formula:

   ```
   Maximum-number-of-SAW-buffers =
        (1.25 × Total-SAW-records) ÷ Records-per-buffer
   ```

If the number produced by this algorithm is significantly different from the default value (MVS 10,000 buffers, VM VSE 500 buffers), then you may want to alter this constant.

Missing Interrupt Handling for a CTC

ISTRACON field name: RACMIHTM

This value is related to missing interrupt handling by VTAM for a CTC (channel-to-channel). It is a time value coded in tenths of seconds.

After CTC I/O has been started, if no reply is received in REPLYTO seconds, VTAM waits (RACMIHTM x .10) seconds. If no reply is received by then, the I/O is halted.

Note: REPLYTO is a system definition parameter.

The default value is 3 minutes.

VM VSE ES/9370 Token-Ring Adapter Buffer Size

ISTRACON field name: RACLBFSZ

This parameter sets the buffer size used in the I/O adapter for token-ring connections in an IBM 9370. This buffer size may be set to zero. If this parameter is set to zero, the adapter uses its default value. If the buffer size is not set to zero, it must be greater than or equal to 96 (hex 60) and must be a multiple of 8. The default value is 112.

VM VSE ES/9370 Token-Ring Adapter Transmit Buffer Maximum

ISTRACON field name: RACLTXMX

This parameter indicates the maximum number of buffers that are used as transmit buffers in the I/O adapter for token-ring connections in an IBM 9370. The adapter always keeps a minimum of two buffers as receive buffers. If this parameter is set to zero, the adapter uses its default value.

Note: The transmit buffer count, with the buffer size, is used to determine the maximum size frame that the adapter can transmit. The default value for the buffer count is 28.

VM VSE ES/9370 Token-Ring Adapter Transmit Buffer Minimum

ISTRACON field name: RACLTXMN

This parameter indicates the number of buffers that are to be reserved as transmit buffers in the I/O adapter for token-ring connections in an IBM 9370. These buffers are reserved as transmit buffers and are never used for receive buffers. If zero is specified, no buffers are reserved for transmit. The minimum transmit buffer count must be less than or equal to the transmit maximum buffer count. The default value is 14.

Option to Receive Dumps for INOPs

ISTRACON field name: RACINOP

This parameter indicates whether VTAM dumps should be generated whenever VTAM detects certain INOPs. Any nonzero value indicates that dumps are generated for INOPs. The default value for RACINOP is zero.

Note: Once this field has been changed, VTAM dumps are taken for certain CTC, local-SNA, and channel-attached NCP INOPs.

Option to Prevent MDR Recording

ISTRACON field name: RACSMDR

This parameter indicates whether the SDLC statistical MDRs should be recorded to LOGREC. The default value is zero, which indicates recording will occur. To prevent recording, code a nonzero RACSMDR value.

Individual IST530I/IST532I Message Pair Limit

ISTRACON field name: RACNTWRE

This parameter limits the number of IST530I and IST532I messages that are written for individual subareas by the IOPD facility.

RACNTWRE can be set to any non-negative, signed, binary, fixed-point value. If the number of pending I/O operations for a subarea is greater than the value of RACNTWRE, then the IOPD facility writes just one message pair for each **type** of pending I/O operation for that subarea, rather than one pair for **each** operation.

The default value for RACNTWRE is hex 7FFFFFFF, which ensures that the IST530I and IST532I messages report each individual pending I/O operation.

MVS/ESA SAW Data Space Limit Factor

ISTRACON field name: RACSAWLM

This parameter is used to calculate the maximum buffer limit for SAW data space. Maximum number of SAW data space buffers equals RACSAWLM times the number of SAW buffers in VTAM private storage. The default value is 100.

MVS/ESA SAW Data Space Packing Factor

ISTRACON field name: RACSAWPK

This parameter is used to calculate the number of SAW data space buffers to queue before packing SAW buffers. The packing threshold equals RACSAWPK times the number of SAW buffers in VTAM private storage. The default value is 0.

MVS/ESA PIU Data Space Limit Factor

ISTRACON field name: RACPIULM

This parameter is used to calculate the maximum buffer limit for PIU data space. The maximum number of PIU data space buffers equals RACPIULM times the number of PIU buffers in VTAM private storage. The default value is 200.

CNM Routing Table

VTAM refers to a communication network management (CNM) routing table to determine which CNM application program is to receive an unsolicited network-services request unit that requires further processing. An application program can embed its own procedure-related identifier (PRID) in each request sent to VTAM. When a reply to the request is returned, VTAM uses the PRID to route the reply to the application program. Unsolicited RUs are not responses to previous requests; they contain network information, but they have no PRIDs.

IBM supplies default routing information for the following CNM application programs:

- The NetView program
- Alias name translation facility (supplied by IBM as part of the NetView program)
- Downstream load utility.

This table is named ISTMGC01. The listing beginning on page 111 shows the current IBM-supplied CNM routing table for all systems.

For any user-written application program to use the CNM interface to receive unsolicited request units, you must write a supplemental table with an entry for each RU. This table should be named ISTMGC00.

If the application to receive this unsolicited RU has opened its ACB, this user-supplied table (ISTMGC00) overrides any entries within the default CNM routing table (ISTMGC01).

Installing the CNM Routing Table

Follow these steps to install the CNM routing table:

1. Assemble the module.
2. Link-edit it to the appropriate VTAM library:
   - MVS SYS1.VTAMLIB or SYS1.LPALIB
     If you place your user-written module in SYS1.LPALIB, re-IPL to reformat the link-pack area. For details, see *MVS/XA System Programming Library: Initialization and Tuning.*
   - VM The VTAMUSER LOADLIB
     See "Installing Tables and Modules in VM" on page 89 for information on using VMFLKED to link-edit this module.
   - VSE A private definition library referenced in the VTAM start procedure.

The link-editing must be done before VTAM is started. The name of the replacement module must be the same as the IBM-supplied module it replaces.

Structure of the CNM Routing Table

A CNM routing table consists of a 12-byte header entry and routing-table entries. The 12-byte header entry contains the size and number of routing table entries that follow it. Each routing-table entry contains the network services RU (NSRU) type to be routed, followed by the application program name to which the NSRU is to be routed. The header format is shown in Table 37.

Table 37. Format of CNM Routing-Table Header

Dec(Hex) Offset	Size (Bytes)	Description
0(0)	2	Number of entries
2(2)	2	Entry length (X'000C')
4(4)	8	Reserved

A CNM routing-table entry consists of 12 bytes defined as shown in Table 38.

Table 38. Format of CNM Routing Table

Dec(Hex) Offset	Size (Bytes)	Description	
0(0)	1	Flag byte:	
		Bit 0	0 = Do not send to VTAM operator 1 = Send to VTAM operator also
		Bit 1	0 = Embed in DELIVER RU 1 = Do not embed in DELIVER RU
		Bits 2–7	Reserved
1(1)	3	Network services RU type	
4(4)	8	Application program name in EBCDIC	

Bit 0 of the flag byte specifies whether the RU is to be sent to the VTAM operator designated to receive unsolicited messages as well as to the CNM application program named in the table entry. This flag bit is supported only for requests for which VTAM has operator message support (that is, ROUTE-INOP).

Bit 1 of the flag byte enables you to specify that an RU is to be sent to a user-written CNM application program without being embedded in a DELIVER RU. If this flag bit is set to 1, VTAM sends the request unit to the application program without embedding it in a DELIVER RU. If the flag bit is set to 0, the request unit is embedded in the DELIVER RU.

The following types of requests can be received by an application program that uses the CNM interface. You should provide an entry in the table for each type of unsolicited NSRU. The network services header value listed in the following text must be coded in the respective entry for that NSRU type (bytes 1–3).

Request	Header Value
RECMS	Hex 010381
RECFMS	Hex 410384
INIT-LOAD	Hex 3F0233
TR-INQ	Hex 3F0814
ROUTE-INOP	Hex 410289
ER-TESTED	Hex 410386
CNM	Hex 810814
NMVT	Hex 41038D

You can code more than one entry associating a single type of RU with more than one application program; however, no more than one program associated with that type of RU can be active at the same time. For example, if an application program associated with the RECMS RU is already active, another application program associated with the RECMS RU is unable to open its ACB.

For a CNM routing table that permits either of two CNM application programs (CNMAPPL1 or CNMALT) to receive unsolicited requests, the following might be coded:

```
         CSECT
HDR      DC    X'0004'         NUMBER OF ENTRIES
         DC    X'000C'         ENTRY LENGTH
         DC    8X'00'          RESERVED

ENT1     DC    X'00'           RESERVED
         DC    X'010381'       RECMS
         DC    CL8'CNMAPPL1'   APPLNAME

ENT2     DC    X'00'           RESERVED
         DC    X'410384'       RECFMS
         DC    CL8'CNMAPPL1'   APPLNAME

ENT3     DC    X'00'           RESERVED
         DC    X'010381'       RECMS
         DC    CL8'CNMALT  '   APPLNAME

ENT4     DC    X'00'           RESERVED
         DC    X'410384'       RECFMS
         DC    CL8'CNMALT  '   APPLNAME
         END
```

Note that in the above example, CNMAPPL1 and CNMALT have each been defined to receive the same type of unsolicited requests (RECMS and RECFMS). In this case, both application programs cannot be active (that is, have open ACBs for application programs whose network names are CNMALT and CNMAPPL1) at the same time.

Note: If you are using a user-written alias name translation facility, you must include it in your CNM routing table as the receiver of the TR-INQ and ROUTE-INOP request units. If you are using the alias name translation facility supplied as a function of the NetView program, the default CNM routing table already contains the information needed for routing.

Logon-Interpret Routine Requirements

When a session-establishment request is received, VTAM uses the interpret table to determine which application program is to be notified. This section discusses the logon-interpret routine requirements.

See *VTAM Resource Definition Reference* for information on the following:

- Coding logon-interpret routines (APPLID routines)
- Interpret table macroinstructions
- Installing and changing interpret tables.

Entry from: VTAM to entry point *routinename.*

Contents of registers at entry:

Register 0:	Length of logon message.
Register 1:	Address of first byte of logon message. (See note 3.)
Register 2:	Address of an 8-byte LU name.
Register 13:	Address of a 72-byte save area provided by VTAM.
Register 14:	Return address.
Register 15:	Address of entry point of this routine.

Operation: The logon-interpret routine is run synchronously in pageable storage under the control of VTAM and not under the control of an application program. For the application program to receive the logon, this routine must validate the logon, obtain the symbolic name of the application program to receive control, and provide this name to VTAM. Otherwise, the routine specifies that the logon is invalid or that the name of the application program was not found.

Because the logon-interpret routine operates at VTAM's main task dispatching priority, there is a possibility of lockout if a wait requires another task action. The routine gets control in supervisor state with a VTAM storage key, so errors within the routine could cause damage to VTAM or to system control blocks and modules.

The logon-interpret routine must also:

- Save and restore the contents of registers 2 – 14 when receiving and passing control
- Use re-enterable code (the routine must not store anything within itself or modify itself during execution)
- Perform no I/O operations; an I/O request causes the routine to terminate abnormally.

Notes:

1. The logon message that is passed to the interpret routine is read-only, and cannot be modified.
2. MVS All data is addressable only in 24-bit mode.
3. For LOGON requests, VTAM again searches the interpret table, after USS translation, looking just for the specified APPLID. After USS translation, register 1 contains the address of the first byte of the APPLID.

Contents of Registers at Exit: Registers 0 and 1 contain the name of the application program (in EBCDIC characters) with which the LU is to establish a session:

Register 0: First 4 characters of name (left-justified).
Register 1: Last 4 characters of name (left-justified).
Registers 2 – 14: Restored to condition at entry.
Register 15: Return code:

00: Application program was found and the name placed in registers 0 and 1.

Nonzero: Application program was not found and the name is not placed in registers 0 and 1.

If the name of the application program contains fewer than 8 characters, use blanks to provide a name with 8 characters.

Installing Tables and Modules in VM

A tool called VMFLKED is available to simplify the process of link-editing the modules for class-of-service tables, interpret tables, logon mode tables, USS tables, CNM routing tables, and VTAM constants.

Use the following procedure to create or update one of these modules using VMFLKED.

1. Copy the IBM-supplied module (IST*xxxxx* ASSEMBLE) on the VTM191 disk and update it, or create your own ASSEMBLE file.
2. Access the disk containing the VTAM macroinstruction libraries (the default is 191).
3. Issue the command:

   ```
   GLOBAL MACLIB VTAMBLD VTAMAC
   ```

4. Assemble the table. This creates files with file types of 'LISTING' and 'TEXT'.
5. Use VMFLKED to link-edit the text file from the assemble step:

   a. Create the input control file as VTAMUSER LKEDCTRL

      Note: This file name is arbitrary. However, it is suggested that you use VTAMUSER for VTAM modules.

   b. Place the following records in the file:

      ```
      %LEPARMS REUS       (starts in column 1)
       INCLUDE ISTxxxxx   (starts in column 2)
       NAME ISTxxxxx(R)
      ```

   c. Issue the command:

      ```
      VMFLKED VTAMUSER (PRINT
      ```

      The output is a load library named VTAMUSER LOADLIB whose load module name is the same as you specified above. The command also creates a link-edit map named VTAMUSER LKEDIT.

6. After the library is created, the VTAM virtual machine needs to:

   a. Access the disk containing the VTAMUSER LOADLIB.
   b. Issue a global command for the VTAMUSER LOADLIB, for example:

      ```
      GLOBAL LOADLIB VTAMUSER VTAM VSCS
      ```

      You can put this command into your EXEC that starts VTAM.

Writing VSCS Exit Routines for VM

VSCS Data Manipulation Exit Routines 93
Exit Routines for Keyboard/Printer and TWX Devices 93
Exit Routines for Display Devices 96
Installing VSCS Data Manipulation Exit Routines 97

Chapter 3. Writing VSCS Exit Routines for VM

This chapter contains information specific to customizing VM/SNA console support (VSCS). It contains information on data manipulation exit routines for VSCS.

VSCS Data Manipulation Exit Routines

You can write data manipulation exit routines to customize the translation of input and output data streams. You can write these exit routines for two categories of devices:

- Keyboard/printers and TWX devices
- Display devices.

IBM supplies a skeleton exit routine, which you must recode in order to activate the data manipulation feature. The following sections provide information you need to recode the data manipulation exit routines. Figure 11 on page 94 shows the source code of the IBM-supplied skeleton exit routine. See "Installing VSCS Data Manipulation Exit Routines" on page 97 for information on installing these routines.

Exit Routines for Keyboard/Printer and TWX Devices

The keyboard/printer and TWX device data manipulation exit routines enable you to manipulate the data stream for a keyboard/printer or TWX device. The name of the routine that manipulates input data is DTIPDNTI. The name of the routine that manipulates output data is DTIPDNTO.

Use the information in Figure 11 on page 94 when coding these exit routines.

```
DTIPXXXX CSECT
         USING *,R15
         B     PROLOG               BRANCH OVER IDENTIFIER
         DC    AL1(15)              LENGTH OF MODULE IDENTIFIER
         DC    C'DTIPXXXX 89.249'  MODULE IDENTIFIER
DTIPXXXX CSECT ,                    CSECT FOR DTIPXXXX
PROLOG   STM   R14,R12,12(R13)      SAVE R14-R12 IN SAVE AREA
         BALR  R12,0                SET UP BASE REGISTER
PSTART   DS    0H
         USING PSTART,R12
         DROP  R15
         L     R14,8(,R13)          GET NEW SAVE AREA ADDRESS
         ST    R13,4(,R14)          SET BACKWARD CHAIN POINTER
         LR    R13,R14              SWITCH TO NEW SAVE AREA
         A     R14,SIZDATAD         ADDRESS OF NEXT SAVE AREA
         ST    R14,8(,R13)          SET FORWARD CHAIN POINTER
         USING DATAD,R13            BASE THE DATAD AREA ON REGISTER 13
         MVC   0(4,R13),MODNAME     PUT MODULE ID IN SAVE AREA
***********************************************************************
*
*           User-Coded Routine Starts Here for DTIPXXXX
*
***********************************************************************
         STC   R15,3(,R13)          PUT RETURN CODE IN SAVE AREA
         L     R13,4(,R13)          SWITCH TO PREVIOUS SAVE AREA
         L     R14,12(,R13)         RESTORE RETURN REGISTER
         LM    R00,R12,20(R13)      RESTORE R00-R12
         BR    R14                  RETURN TO CALLER
DATAD    DSECT                      DSECT FOR DYNAMIC STORAGE
         DS    0F
         DS    18F                  SPACE FOR 18 FULLWORD SAVE AREA
WORK1    DS    CL4                  USER DYNAMIC STORAGE
WORK2    DS    CL8                  USER DYNAMIC STORAGE
ENDDATAD EQU   *                    END OF DYANAMIC STORAGE
DTIPXXXX CSECT                      STATIC STORAGE IN MODULE
         DS    0F
SIZDATAD DC    AL1(0)
         DC    AL3(((ENDDATAD-DATAD+7)/8)*8) DYNAMIC STORAGE LENGTH
MODNAME  DC    C'XXXX'             MODULE IDENTIFIER FOR DTIPXXXX
R00      EQU   00                  REGISTER 0 EQUATE
R12      EQU   12                  REGISTER 12 EQUATE
R13      EQU   13                  REGISTER 13 EQUATE
R14      EQU   14                  REGISTER 14 EQUATE
R15      EQU   15                  REGISTER 15 EQUATE
         DS    0D
ENDDATAS EQU   *                   END OF STATIC STORAGE
         END   DTIPXXXX
```

Note: XXXX is a placeholder for the last four characters of the name of the input or output data manipulation exit routine.

Figure 11. IBM-Supplied Skeleton Data Manipulation Exit Routine for VSCS

© Copyright IBM Corp. 1984, 1990

Initial Register and Parameter List Contents

When the routine gains control, register contents are as follows:

Register 1:	Address of parameter list (described in Table 39).
Register 13:	Address of a standard 18-fullword save area for use by this routine.
Register 14:	Return address.
Register 15:	Address of the entry point of this routine.

Table 39. Parameter List for Keyboard/Printer and TWX-Device Data Manipulation Exit Routines

Dec(Hex) Offset	Size (Bytes)	Description
0(0)	4	Address of data (in ASCII line code format)
4(4)	4	Length of data
8(8)	4	Device subtype: 3767 = X'00000001' 2741 = X'00000002' TWX = X'00000003' **Note:** X'00000000' indicates no value was set for the TERM operand for this device.

Design Requirements and Restrictions

Keep in mind the following information when writing these routines:

- The length of the data cannot be changed (which prevents users from translating APL and TEXT data).
- Data outbound to a TWX device is in the form of reverse ASCII line codes with mark parity supplied by the control program.
- You must follow the entry and exit linkage shown in Figure 11 on page 94 and in Table 39.
- The online message for TWX devices is not passed to the user exit (DTIPDNTO) because the data is not in reverse ASCII format.

Final Register Contents

These routines must leave the register status as follows on return:

Registers 0 – 14:	Should be returned intact.
Register 13:	Points to the save area and must be the same upon return.

In addition, DTIPDNTI and DTIPDNTO must leave register 15 with the following status:

Register 15: Valid return codes:

0, VSCS translates the output data stream from ASCII to EBCDIC.

8, Override VSCS translation.

Note: If return code 8 is used to bypass VSCS translation, the data must be in EBCDIC form as required by the NCP on exit from this routine.

Exit Routines for Display Devices

IBM provides an exit routine to modify the inbound data stream for display devices (DTIPDDSI) and one to modify the outbound data stream for display devices (DTIPDDSO). If you replace these exit routines, be sure that your exit routine does not attempt to alter the data stream when the LU state is a VSCS internal mode. Similarly, be careful not to destroy user-embedded screen control commands.

The following sections provide information needed to code these routines.

Initial Register and Parameter List Contents

When the routine gains control, register contents are as follows:

Register 1: Address of parameter list (described in Table 40).

Register 13: Address of a standard 18-fullword save area for use by this routine.

Register 14: Return address.

Register 15: Address of the entry point of this routine.

Table 40. Parameter List for Display-Device Data Manipulation Exit Routines

Dec(Hex) Offset	Size (Bytes)	Description
0(0)	4	Address of data
4(4)	4	Length of data
8(8)	1	Device attributes: X'80' Alternate size available X'40' New APL/TEXT (3278 or 3279) X'1C' Color/highlight functions: X'10' Extended data stream available X'08' Extended color available X'04' Extended highlight available
9(9)	1	Character set being used: X'00' EBCDIC X'01' APL X'02' TEXT
10(A)	1	LU state at time of entry **X'80' or >** A VSCS internal mode; the data stream should not be altered if the LU is in one of these states.
11(B)	1	Not used – reserved
12(C)	1	Screen width (VSCS uses this in console mode)
13(D)	1	Screen depth (VSCS uses this in console mode)

Final Register Contents

These routines must leave the register status as follows on return:

Registers 0–14: Should be returned intact.
Register 13: Points to the save area and must be the same upon return.

In addition, DTIPDDSI and DTIPDDSO must leave register 15 with the following status:

Register 15: Valid return codes:

0, Permit VSCS translation: VSCS translates APL or TEXT data streams.

8, Override VSCS translation: VSCS does not translate APL and TEXT data streams.

Installing VSCS Data Manipulation Exit Routines

To install one of the VSCS data manipulation exit routines, you must:

- Code and assemble the user exit routine, being sure to use the correct name for each exit routine being replaced
- Replace the user exit routine on the merge disk, and link-edit VSCS
- Activate the user exit routines by setting the option KEXIT=Y (to activate keyboard/printer and TWX exits), setting the option DEXIT=Y (to activate display exits), or setting both options. This can be done by either:
  - Starting VSCS with a PARM that specifies a DTIUSER(0–9) in which KEXIT and DEXIT are coded as desired, or
  - Starting VSCS, then issuing VSCS CHANGE DEXIT=Y, or VSCS CHANGE KEXIT=Y, or issuing both.

Writing TSO/VTAM Exit Routines for MVS

IKTCASX1 — Error Handling for Non-Supported Terminals 102
IKTGETXT — Editing on Non-Supported Terminals 102
IKTIDSX1 — Output Editing for IBM 3270 Terminals 103
IKTIDSX2 — Input Editing for IBM 3270 Terminals 103
IKTIDSX3 — Attention Handler for IBM 3270 Terminals 103
IKTIDSX4 — TGET Edit for IBM 3270 Terminals 104
IKTINX1 — Logon Edit 104
IKTINX2 — I/O Manager Initialization 105
IKTRTX1 — Output Edit for IBM 3767, 3770, and 2741 Terminals 105
IKTRTX2 — Input Edit for IBM 3767 and 3770 Terminals 106
IKTRTX3 — Attention Handler for IBM 3767 and 3770 Terminals 106
IKTRTX4 — Edit for IBM 3767, 3770, and 2741 Terminals 106
IKTWTX1 — Output Edit for WTTY and TWX Terminals 107
Installing TSO/VTAM Exit Routines 107

Chapter 4. Writing TSO/VTAM Exit Routines for MVS

This chapter contains information on writing exit routines for TSO/VTAM. These routines can be written to:

- Perform input and output editing that replaces or supplements IBM-supplied editing
- Perform attention handling that replaces IBM-supplied attention handling
- Provide support for terminals not supported by TSO/VTAM.

  Note: An installation must write its own terminal input manager and terminal output manager, in addition to exit routines IKTGETXT, IKTINX2, and IKTINX1, to support terminals that are not supported by TSO/VTAM.

These exit routines are optional, but before you can use an exit routine, you must link-edit it with the object module that calls it. For additional information, see "Installing TSO/VTAM Exit Routines" on page 107.

Note: If you do not write these exit routines, you will get unresolved external reference messages during link-editing of the load modules that call them.

To determine whether any of the input and output editing exit routines are needed, see the sections on the TPUT and TGET options in the appropriate *TSO Guide to Writing a Terminal Monitor Program.*

At appropriate points during VTIOC and TCAS processing, a check is made to determine if a particular exit routine exists. If it exists, it is called; if it does not exist, normal processing continues.

The exit routines are summarized in Table 41. This chapter describes the routines in alphabetical sequence and states what the required input and output is for each routine. For information about coding your own exit routines, see *VTAM Programming.*

Table 41 (Page 1 of 2). Summary of the TSO/VTAM Exit Routines

Name	Purpose	Terminal Type	Caller
VTIOC			
IKTGETXT	Edit input data	Unsupported	IKTVTGET
IKTIDSX1	Replace or supplement IBM-supplied output editing	3270	IKT3270O
IKTIDSX2	Supplement IBM-supplied input editing	3270	IKT3270I
IKTIDSX3	Supplement IBM-supplied attention handling	3270	IKT3270I
IKTIDSX4	Replace or supplement IBM-supplied input editing	3270	IKTVTGET
IKTINX2	Initialize user-written I/O managers	Unsupported	IKTXINIT

Table 41 (Page 2 of 2). Summary of the TSO/VTAM Exit Routines

Name	Purpose	Terminal Type	Caller
IKTRTX1	Replace or supplement IBM-supplied output editing	3767/3770 2741	IKT3767O
IKTRTX2	Supplement IBM-supplied input editing	3767/3770 2741 WTTY TWX	IKT3767I
IKTRTX3	Replace IBM-supplied attention handling	3767/3770 (LU1)	IKTMLU1
IKTRTX4	Replace or supplement IBM-supplied input editing	3767/3770 (LU1)	IKTVTGET
IKTWTX1	Replace or supplement IBM-supplied output editing	TWX WTTY	IKTWTTYO
TCAS			
IKTCASX1	Replace or supplement IBM-supplied logon error messages	Unsupported	IKTCAS31
IKTINX1	Set terminal type and buffer size	Unsupported	IKTCAS23

IKTCASX1 — Error Handling for Non-Supported Terminals

You must write this routine if you want to send an error message to a terminal that is not supported by TSO/VTAM. IKTCAS31 calls IKTCASX1 if the maximum number of users is reached or if IKTCAS31 cannot obtain storage for either of the TSBs or the logon buffer. It also calls IKTCASX1 if the address space for this session cannot be obtained.

Input from IKTCASX1

When VTAM passes control to IKTCASX1, register contents are as follows:

Register 0:	Address of the message buffer.
Register 1:	Address of the message number.
Register 10:	Address of a work element, IKTWESTD (terminal type defined).

Output from IKTCASX1

The final register contents are as follows:

Register 1:	New request unit size.	
Register 15:	Return code:	
	0:	IBM routine performed editing.
	Nonzero:	Exit routine performed all editing.
MSGBUF:	Message buffer (contains user's message).	

IKTGETXT — Editing on Non-Supported Terminals

You must write this routine if you want to use a terminal not supported by TSO/VTAM. IKTVTGET calls IKTGETXT instead of using the IBM-supplied code at statement label EDIT3270 (for 3270 terminals) or EDIT3767 (for 3767 and 3770 terminals) in IKTVTGET. IKTGETXT must scan the input data, edit it, and move the edited data from the input queue to the TGET requester's data area.

Input to IKTGETXT

When VTAM passes control to IKTGETXT, register contents are as follows:

Register 0:	Address of SVRB extended save area.
Register 1:	Address of IKTIPARM.

Output from IKTGETXT

None.

IKTIDSX1 — Output Editing for IBM 3270 Terminals

You may write this routine to perform 3270 output editing in place of or in addition to that performed by the IBM-supplied routine IKT32700 (before IKT32700 moves the data from the output queue into the output request unit).

Input to IKTIDSX1

The input to the IKTIDSX1 routine is as follows:

Register 1:	Address of IKTOPARM.
OPACBUFA:	Current buffer address.
OPACBUFL:	Current buffer length.

Output from IKTIDSX1

Register 15:	Return code:	
	0:	IBM-supplied routine IKT32700 should perform editing; data is still on the output queue.
	Nonzero:	Exit routine performed all editing; portions of IKT32700 are bypassed.

IKTIDSX2 — Input Editing for IBM 3270 Terminals

You may write this routine to perform input scanning and editing in addition to that performed by the IBM-supplied routine IKT3270I. If provided, IKTIDSX2 is called after the data is translated (if necessary) from ASCII code to EBCDIC, but before it is scanned for input line delimiters; it is then broken into line segments and placed on the input queue.

Input to IKTIDSX2

The input to the IKTIDSX2 routine is as follows:

Register 1:	Address of IKTMPL.

Output from IKTIDSX2

MPLTXTA:	Address of available input data.
MPLTXTL:	Length of available input data.

IKTIDSX3 — Attention Handler for IBM 3270 Terminals

You may write this routine to handle attention interruptions from 3270 terminals during input editing instead of using the IBM-supplied routine IKTATTN. One use of a user-coded routine is to clear the queues conditionally (rather than unconditionally) when an attention interruption is received.

Input to IKTIDSX3

The input to the IKTIDSX3 routine is as follows:

PSAAOLD:	Address of ASCB.
ASCBTSB:	Address of TSB.
TSBEXTNT:	Address of TSB extension.
TSBXTVWA:	Address of TVWA.

Output from IKTIDSX3

None.

IKTIDSX4 — TGET Edit for IBM 3270 Terminals

You may write this routine to perform 3270 editing in place of or in addition to that performed by the IBM-supplied routine IKTVTGET. IKTVTGET scans for invalid data and 3270 control characters, and moves the data from the input queue to the TGET requester's data area. You might write an edit exit routine to change TGET EDIT editing criteria.

Input to IKTIDSX4

The input to the IKTIDSX4 routine is as follows:

Register 0:	Address of IKTXSA (SVRB extended save area).
Register 1:	Address of IKTIPARM.

Output from IKTIDSX4

The output from the IKTIDSX4 routine is as follows:

Register 15: Return code:

X'00': Exit routine performed the entire edit operation and moved the input data to the TGET data area.

X'04': Exit routine performed only a data scan; the IBM-supplied code (EDIT3270) should perform editing.

IKTINX1 — Logon Edit

You must write this routine if you want to use a terminal not supported by TSO/VTAM. IKTCAS23 calls IKTINX1, if provided, when a logon request is encountered from a terminal other than an IBM 3270, 3767, or 3770. (If TSO/VTAM is used with NTO, the 2741, WTTY, and TWX Models 33 and 35 are also supported.) IKTINX1 must verify that the terminal is supported by user-written routines (that is, a terminal input manager, a terminal output manager, and edit routine IKTGETXT), and it must provide the terminal type of hex 03, the buffer size, and the device bind image.

Input to IKTINX1

The input to the IKTINX1 routine is as follows:

Register 1: Address of a parameter list containing:

- Address of the RPL
- Address of 1 byte (WETMTP) indicating terminal type
- Address of 2 bytes (WETMBF) indicating terminal buffer size
- Address of WEBIND (bind image).

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Output from IKTINX1

The output from the IKTINX1 routine is as follows:

Register 15: Return code:

X'00':	Recognized terminal type; logon processing continues.
X'04':	Unrecognized terminal type; logon processing terminates.

IKTINX2 — I/O Manager Initialization

You must write this routine if you want to use terminal input managers and terminal output managers you have written. IKTINX2 should perform the same function for user-coded input and output managers that module IKTIIOM performs for the IBM-supplied I/O managers. IKTINX2 allocates storage for and initializes the I/O manager SRBs.

Input to IKTINX2

None.

Output from IKTINX2

The output from the IKTINX2 routine is as follows:

Register 15: Return code:

0:	Successful initialization.
Nonzero:	Unsuccessful initialization; logon processing terminates.

IKTRTX1 — Output Edit for IBM 3767, 3770, and 2741 Terminals

You may write this routine to perform 3767, 2741, or 3770 output editing in place of or in addition to that performed by the IBM-supplied routine IKT3767O (before IKT3767O moves the data from the output queue to the output request unit). The IBM-supplied code scans data, edits it according to TPUT operands specified (EDIT, ASIS, or CONTROL), and provides user-specified character translation.

Input to IKTRTX1

The input to the IKTRTX1 routine is as follows:

Register 1:	Address of IKTOPARM.
OPACBUFA:	Current buffer address.
OPACBUFL:	Current buffer length.

Output from IKTRTX1

The output from the IKTRTX1 routine is as follows:

Register 15: Return code:

0:	IBM-supplied code should perform editing; data is still on the output queue.
Nonzero:	Exit routine performed the entire edit operation; the IBM-supplied code is bypassed.

"Restricted Materials of IBM"
Licensed Materials — Property of IBM

IKTRTX2 — Input Edit for IBM 3767 and 3770 Terminals

You may write this routine to perform input scanning and editing in addition to that performed by the IBM-supplied routine IKT3767I. If provided, IKTRTX2 is called after the data is translated (if necessary) from ASCII code to EBCDIC and the user-supplied character translation is performed on the data, but before the data is scanned for input line delimiters, broken into single lines, and placed on the input queue.

Input to IKTRTX2

The input to the IKTRTX2 routine is as follows:

Register 1:	Address of IKTMPL.
MPLTXTA:	Address of buffer.
MPLTXTL:	Length of buffer.

Output from IKTRTX2

The output from the IKTRTX2 routine is as follows:

MPLTXTA:	Address of buffer.
MPLTXTL:	Length of buffer.

IKTRTX3 — Attention Handler for IBM 3767 and 3770 Terminals

You may write this routine to handle attention interruptions from 3767 or 3770 terminals instead of using the IBM-supplied routine IKTATTN. One use of a user-supplied attention handler might be to clear the queues conditionally (rather than unconditionally) when an attention interruption is received.

Input to IKTRTX3

None.

Output from IKTRTX3

None.

IKTRTX4 — Edit for IBM 3767, 3770, and 2741 Terminals

You may write this routine to perform 3767 or 3770 editing in place of or in addition to that performed by the IBM-supplied routine IKTVTGET. IKTVTGET scans for invalid data and moves the data from the input queue to the TGET requester's data area. You might write an edit exit routine to change TGET EDIT editing criteria.

Input to IKTRTX4

The input to the IKTRTX4 routine is as follows:

Register 0:	Address of IKTXSA (SVRB extended save area).
Register 1:	Address of IKTIPARM.

Output from IKTRTX4

The output from the IKTRTX4 routine is as follows:

Register 15: Return code:

X'00': Exit routine performed the entire edit operation and moved the input data to the TGET data area.

X'04': Exit routine performed only a data scan; the IBM-supplied code (EDIT3767) should perform editing.

IKTWTX1 — Output Edit for WTTY and TWX Terminals

You may write this routine to perform TWX or WTTY output editing in place of or in addition to that performed by the IBM-supplied routine IKTWTTYO. The IBM-supplied code scans data, edits it according to the TPUT operands specified (EDIT, ASIS, or CONTROL), and provides user-specified character translation.

Input to IKTWTX1

The input to the IKTWTX1 routine is as follows:

Register 1:	Address of IKTOPARM.
OPACBUFA:	Current buffer address.
OPACBUFL:	Current buffer length.

Output from IKTWTX1

The output from the IKTWTX1 routine is as follows:

Register 15: Return code:

0:	Continue with normal editing.
Nonzero:	No further editing.

Installing TSO/VTAM Exit Routines

You must link-edit exit routines you have coded with the object modules that call them. The following table shows the calling module and load module for each exit routine.

Table 42. Routine — Module Cross-Reference

Name	Calling Module	Load Module
IKTCASX1	IKTCAS31	IKTCAS30
IKTGETXT	IKTVTGET	IGC0009C
IKTIDSX1	IKT3270O	IKTIOM02
IKTIDSX2	IKT3270I	IKTIOM02
IKTIDSX3	IKT3270I	IKTIOM02
IKTIDSX4	IKTVTGET	IGC0009C
IKTINX1	IKTCAS23	IKTCAS20
IKTINX2	IKTXINIT	IKJEFLA
IKTRTX1	IKT3767O	IKTIOM01
IKTRTX2	IKT3767I	IKTIOM01
IKTRTX3	IKTIMLU1	IKTIOM01
IKTRTX4	IKTVTGET	IGC0009C
IKTWTX1	IKTWTTYO	IKTIOM01

Appendixes

Appendix A. IBM-Supplied CNM Routing Table 111

Appendix B. Sample Session Management Exit Routine 113
Initialization 114
Function Selection 116
Begin Function 117
Obtaining User Storage 117
Loading NETID Registration Table 118
Returning to VTAM 121
Secondary Authorization Function 121
Gateway Path Selection Function for MVS and VM 125
Examining LU Names 126
Modifying Gateway Path Selection List 128
Alias Selection Function for MVS and VM 130
Translating DLU Name 131
Translating OLU Name 133
Final Accounting Function 136
End Function 138

Appendix A. IBM-Supplied CNM Routing Table

```
*/*FUNCTION:   THIS IS THE CNM ROUTING TABLE FOR IBM-SUPPLIED
*/*            CNM APPLICATIONS. EACH ENTRY IN THE TABLE
*/*            REPRESENTS AN APPLICATION.
ISTMGC01 CSECT
         DS    0F
         DC    S(ENTRIES)     NUMBER OF ENTRIES
         DC    X'000C'        LENGTH OF EACH ENTRY
         DC    XL4'00000000'  RESERVED
         DC    XL4'00000000'  RESERVED
ENTRIES  EQU   22             NUMBER OF ENTRIES
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410386'    ER_TESTED RU
         DC    CL8'ISTNOP  '  NETWORK OPERATOR SERVICES NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410289'    ROUTE_INOP RU
         DC    CL8'ISTNOP  '  NETWORK OPERATOR SERVICES NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'010381'    RECMS RU
         DC    CL8'BNHDSERV'  NPDA ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'010381'    RECMS RU
         DC    CL8'BNJDSERV'  NPDA VERSION 2 ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410384'    RECFMS RU
         DC    CL8'BNHDSERV'  NPDA ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410384'    RECFMS RU
         DC    CL8'BNJDSERV'  NPDA VERSION 2 ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'3F0233'    INIT LOAD RU
         DC    CL8'DLUPULP '  DOWNSTREAM LOAD UTILITY ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410384'    RECFMS RU
         DC    CL8'AAUTSKLP'  NLDM ACB NETWORK NAME WITHOUT NPDA
*
         DC    XL1'40'        FLAG - DO NOT IMBED IN DELIVER RU
         DC    XL3'3F0814'    TRANSLATE INQUIRE RU
         DC    CL8'ALIASAPL'  ALIAS ACB NETWORK NAME
*
*              NLDM R2 UNIQUE SUPPORT
*
         DC    XL1'40'        FLAG - DO NOT IMBED IN DELIVER RU
         DC    XL3'810814'    CNM AMRU
         DC    CL8'BNHDSERV'  NPDA ACB NETWORK NAME
*
         DC    XL1'40'        FLAG - DO NOT IMBED IN DELIVER RU
         DC    XL3'810814'    CNM AMRU
         DC    CL8'BNJDSERV'  NPDA VERSION 2 ACB NETWORK NAME
*
         DC    XL1'40'        FLAG - DO NOT IMBED IN DELIVER RU
         DC    XL3'810814'    CNM AMRU
         DC    CL8'AAUTSKLP'  NLDM ACB NETWORK NAME WITHOUT NPDA
*
         DC    XL1'80'        FLAG - ALSO SEND TO NETWORK OPERATOR
         DC    XL3'410289'    ROUTE INOP RU
         DC    CL8'BNHDSERV'  NPDA ACB NETWORK NAME
*
         DC    XL1'80'        FLAG - ALSO SEND TO NETWORK OPERATOR
         DC    XL3'410289'    ROUTE INOP RU
         DC    CL8'BNJDSERV'  NPDA VERSION 2 ACB NETWORK NAME
*
         DC    XL1'80'        FLAG - ALSO SEND TO NETWORK OPERATOR
         DC    XL3'410289'    ROUTE INOP RU
         DC    CL8'AAUTSKLP'  NLDM ACB NETWORK NAME WITHOUT NPDA
*
*              NLDM R3 UNIQUE SUPPORT
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'41038D'    NMVT RU
         DC    CL8'BNJDSERV'  NPDA VERSION 2 ACB NETWORK NAME
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'41038D'    NMVT RU
         DC    CL8'AAUTSKLP'  NLDM ACB NETWORK NAME WITHOUT NPDA
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'010381'    RECMS RU
         DC    CL8'DSICRTR '  NETVIEW ROUTER
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'410384'    RECFMS RU
         DC    CL8'DSICRTR '  NETVIEW ROUTER
*
         DC    XL1'40'        FLAG - DO NOT IMBED IN DLV RU
         DC    XL3'810814'    CNM RU
         DC    CL8'DSICRTR '  NETVIEW ROUTER
*
         DC    XL1'80'        FLAG - ALSO SEND TO NOS
         DC    XL3'410289'    ROUTE INOP RU
         DC    CL8'DSICRTR '  NETVIEW ROUTER
*
         DC    XL1'00'        FLAG BYTE
         DC    XL3'41038D'    NMVT RU
         DC    CL8'DSICRTR '  NETVIEW ROUTER
*
*              END OF REFERENCED TABLE ENTRIES
*
         DC    XL1'00'        UNUSED TABLE ENTRY
         DC    XL3'000000'    IF REFERENCED THEN ALSO UPDATE
         DC    CL8'        '  NUMBER OF ENTRIES COUNT
END      ISTMGC01
```

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Appendix B. Sample Session Management Exit Routine

This session management exit routine controls various functions associated with VTAM logical unit session processing. These functions include:

- Gateway path selection
- Alias name translation
- Adjacent SSCP selection
- Session authorization
- Session accounting
- Session takeover accounting.

This sample session management exit routine is only prototype code that can be included in a more complete and comprehensive session management exit routine. The sample code illustrates how the initial session management exit routine environment is established, the type of information that is available in each parameter list for the specific sample environment, and some techniques used to examine the various parameter lists. The sample session management exit routine is written specifically for the SNA network environment illustrated in Figure 12 on page 114. The sample uses only the following session management exit routine functions:

- Begin
- Gateway path selection
- Alias translation
- Secondary session authorization
- End.

A portion of a final accounting function is also included, however, the code is specifically related to the alias function. The initial and final accounting functions are normally used to record session connection time.

Note: The examples in this appendix are intended as an educational tool. They show how various functions in a session management exit routine can be used to control VTAM functions. The examples shown are not complete and may not work as presented here.

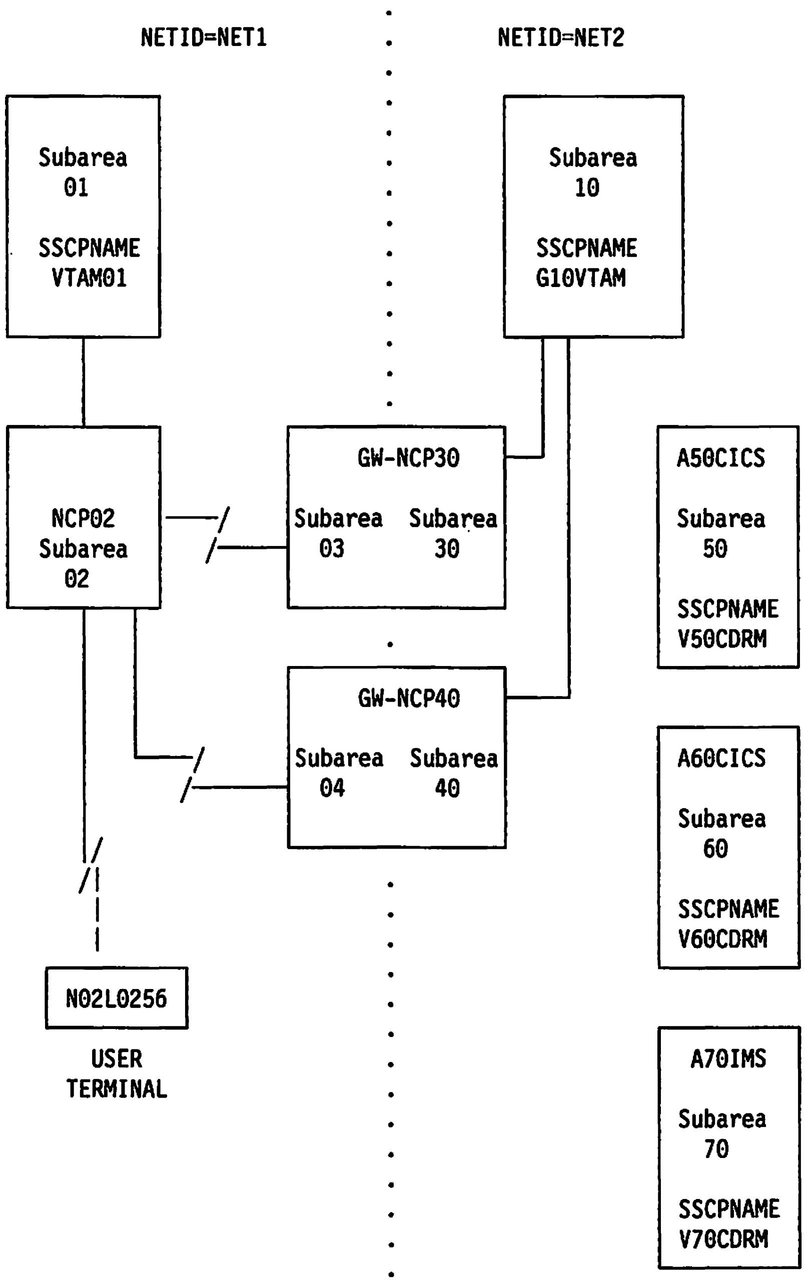


Figure 12. Network Environment for Sample Session Management Exit Routine

Initialization

Each time VTAM invokes the session management exit routine, the exit routine must perform its initial programming functions. For example, the session management exit routine must save the register contents into a save area provided by VTAM. The exit routine entry point address, which establishes the necessary routine addressing, is contained in register 15.

VSE The ABEND function in the exit routine would cause VTAM to ABEND, so it is advisable not to execute the ABEND function in the exit routine. The prologue code that performs these initial tasks follows.

```
***********************************************************************
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
***********************************************************************
*        SME ROUTINE PROLOGUE
***********************************************************************
ISTEXCAA CSECT
*
** SAVE REGISTERS IN VTAM PROVIDED SAVEAREA
*
         STM   R14,R12,12(R13)    SAVE VTAM'S REGISTERS
*
** ESTABLISH SME ROUTINE ADDRESSABILITY
*
INITS1   EQU   *
         LR    R12,R15            ESTABLISH ADDRESSABILITY
         USING ISTEXCAA,R12       BASE REGISTER USED EQUALS R12
         B     INITS1A            SAVE PARAMETER LIST ADDRESS
*
** MODULE NAME AND ASSEMBLY DATE
*
         DC    CL8'ISTEXCAA'      CSECT NAME
         DC    CL8'&SYSDATE'      ASSEMBLY DATE
**
*
INITS1A  EQU   *
         LR    R11,R1             ADDRESS FUNCTION PARAMETER LIST
*
** ESTABLISH SME ROUTINE REGISTER SAVE AREA
*
INITS1B  EQU   *
         GETMAIN R,LV=72
         LTR   R15,R15
         BNZ   ABEND
INITS1C  EQU   *
         ST    R13,4(R1)          BACKWARD SAVE AREA POINTER
         ST    R1,8(R13)          FORWARD SAVE AREA POINTER
         LR    R13,R1             SME ROUTINE REGISTER SAVE AREA
         B     INITS2             CONTINUE INITIALIZATION PROCESSING
```

```
** TERMINATE SME ROUTINE ABNORMALLY DUE TO ERRORS:
** COMPLETION CODE=999, REASON=SMEF
*
ABEND    ABEND 999,REASON=X'E2D4C5C6',DUMP
***********************************************************************
```

Function Selection

VTAM invokes the session management exit routine to perform various LU-session processing functions. You must design the session management exit routine to process these functions. You indicate in the begin function which LU-session processes require VTAM to invoke the session management exit routine. After VTAM invokes the session management exit routine, the routine must read the function code in the function parameter list to determine which LU-session processing to perform.

The sample session management exit routine coding compares the exit routine function code to a value associated with the functions selected. The function code values for this sample exit routine follow:

Function	Value
Begin	X'FE'
Secondary session authorization	X'01'
Gateway path selection	X'04'
Alias name translation	X'07'
End	X'FF'

The address of the exit routine function code is always the second field in the function parameter list. The session management exit routine examines the code and branches to the appropriate processing within the session management exit routine. The following code illustrates this technique.

```
***********************************************************************
INITS2   EQU   *
*
** LOCATE EXIT ROUTINE FUNCTION CODE
*
         L     R1,4(R11)          EXIT ROUTINE FUNCTION CODE ADDRESS
*
**  EXAMINE EXIT ROUTINE FUNCTION CODE TO DETERMINE LU SESSION PROCESSING
*
         CLI   0(R1),X'FE'        TEST FOR BEGIN FUNCTION
         BE    BEGIN              YES, BRANCH TO FUNCTION PROCESSING
*
         CLI   0(R1),X'01'        TEST FOR SECONDARY SESSION AUTHORIZATION
         BE    SECAUTH            YES, BRANCH TO FUNCTION PROCESSING
*
         CLI   0(R1),X'04'        TEST FOR GATEWAY PATH SELECTION
         BE    GWPATH             YES, BRANCH TO FUNCTION PROCESSING
*
         CLI   0(R1),X'07'        TEST FOR ALIAS NAME TRANSLATION FUNCTION
         BE    ALIAS              YES, BRANCH TO FUNCTION PROCESSING
*
         CLI   0(R1),X'FF'        TEST FOR END FUNCTION
         BE    END                YES, BRANCH TO FUNCTION PROCESSING
*
```

```
         B     RETURN
*
************************************************************************
```

Begin Function

The begin function is a required element in any session management exit routine. This function is invoked at VTAM initialization. The begin function indicates to VTAM the functions for which the exit routine is to be invoked.

The fourth entry in the parameter list passed to the begin function contains the address of the exit routine functions. The session management exit routine must set the proper bit on (B'1') for each function for which VTAM is to call the routine. The bit settings for each of the functions associated with the sample routine follow:

Function	Value
Secondary session authorization	Byte 0, bit 1 or hex 40
Gateway path selection	Byte 0, bit 3 or hex 10
Alias name translation	Byte 1, bit 2 or hex 20
End function	Byte 0, bit 4 or hex 08

Obtaining User Storage

In this example, the begin function sets these indicators in the exit options field. It also obtains storage, using a GETMAIN macroinstruction, to maintain information used by the various function routines. The format of the user storage area is shown in Figure 13 on page 118.

The function parameter list, which is passed by VTAM to each exit routine, contains the address of a user data field. This field stores the address of the storage area acquired by the begin function. Functions subsequently called by VTAM can then read the user data field.

DEC(HEX) Offset	
0 (00)	TABLE INDICATOR
4 (04)	ADDRESS OF TABLE
8 (08)	PREFERRED GWPATH NAME
16 (0F)	PREFERRED GWPATH SUBAREA
20 (14)	OLD GWPATH NAME
28 (1C)	OLD GWPATH SUBAREA ADDRESS
32 (20)	ALIAS NAME POOL ADDRESS
36 (24)	RESERVED
40 (28)	RESERVED
44 (2C)	RESERVED

Figure 13. User Data Storage Area Format

Loading NETID Registration Table

The begin function also loads a table that contains registered, valid network identifiers (NETIDs). The session management exit routine functions use this table to ensure that an LU session request is initiated from or directed to a registered network. In this example, the begin function sets an indicator in the user storage area to inform other exit routine functions that the table is loaded and contains valid entries. The format of the NETID registration table is shown in Figure 14.

DEC(HEX) Offset	
0 (0)	TABLE ENTRY COUNT
4 (4)	NETWORK IDENTIFIER #1
12 (C)	NETWORK IDENTIFIER #2
28 (F)	
NN (NN)	NETWORK IDENTIFIER #N

Figure 14. Network Identifier (NETID) Registration Table

Since the begin function, which is invoked only at VTAM initialization, loads the table, there is no direct mechanism provided by VTAM to reload the table without restarting VTAM. However, there are commands that can prompt the session management exit routine to reload the table dynamically.

For example, when a new table is ready to be loaded, the system programmer can issue the following VTAM operator command to initiate a "dummy" LU session:

```
VARY NET,LOGON,ID=RELOAD,LOGON=TABLE
```

The command in this example is issued at the gateway VTAM (GWVTAM10). In an attempt to initiate the "dummy" LU session between the secondary logical unit, RELOAD, and the primary logical unit, TABLE, VTAM invokes the specified session management exit routine functions. For example, the secondary authorization function can determine if the session request is for the "dummy" LU session. When the session management exit function discovers the session setup is for the "dummy" LU session, the NETID registration table is reloaded. After the table is reloaded, the function routine can reject the session request with a proper return code to VTAM.

The sample code below also checks the NETID registration table to ensure that the table entry count is not zero. If there are no entries in the table, the session management exit begin function sets a table indicator in the first byte of the user data storage area. Any session management exit function routines subsequently invoked by VTAM use this table indicator. If the table indicator is set to hex FF, the function routine does not allow a cross-network session to be established. The routine exits with the proper return code.

The begin function also obtains another user data storage area for use by the session management exit alias translation function. This storage is initialized and maintained by the session management exit routine. The begin function performs a GETMAIN request to obtain 4096 bytes of virtual storage for an alias name pool and saves the address of the storage in the user data storage area. For more information concerning how this table is used, refer to "Alias Selection Function for MVS and VM" on page 130.

The following sample code obtains the user data storage, loads a NETID registration table, and sets the bits appropriately in the exit options field.

```
***********************************************************************
*        SME ROUTINE - BEGIN FUNCTION
***********************************************************************
BEGIN    EQU   *
*
** ISSUE GETMAIN TO OBTAIN USER DATA STORAGE AREA
*
BEGINS1  EQU   *
         GETMAIN R,LV=48          OBTAIN USER DATA STORAGE AREA
         LTR   R15,R15            DETERMINE IF GETMAIN SUCCESSFUL
         BZ    BEGINS1A           YES, CONTINUE PROCESSING
         B     ABEND              TERMINATE SME ROUTINE
*
** SAVE ADDRESS OF USER DATA STORAGE IN USER DATA FIELD
*
BEGINS1A EQU   *
         L     R2,8(R11)          PARAMETER LIST - USER DATA FIELD ADDRESS
         ST    R1,0(R2)           SAVE USER DATA STORAGE - GETMAIN
         LR    R10,R1             RETAIN ADDRESS OF USER DATA STORAGE
         XC    0(32,R10),0(R10)    CLEAR USER DATA STORAGE AREA
*
** LOAD NETID REGISTRATION TABLE
*
BEGINS2  EQU   *
         LA    R2,BEGINS2A         LOAD ERROR CONDITION RETURN ADDRESS
         LOAD  EP=NETIDTAB,        LOAD MODULE NAME - NETID REGISTRATION TABLE X
               ERRET=(R2)          LOAD ERROR RETURN ROUTINE
         B     BEGINS2B            NO, CONTINUE PROCESSING
*
** LOAD FAILURE - SET TABLE INDICATOR
**
** NO NETID REGISTRATION TABLE:
**
**    X'FF' - INDICATE TO OTHER SME FUNCTIONS DO NOT
**            PERMIT CROSS NETWORK SESSIONS TO SETUP
*
BEGINS2A EQU   *
         MVI   0(R10),X'FF'        LOAD FAILURE - SET TABLE INDICATOR
         B     BEGINS3             RETURN TO VTAM
*
** SAVE LOAD MODULE ADDRESS OF NETID REGISTRATION TABLE IN
** USER DATA STORAGE AREA AND TEST FOR VALID NETID ENTRIES
*
BEGINS2B EQU   *
         ST    R0,4(R10)           SAVE LOAD MODULE ADDRESS - USER DATA STORAGE
         L     R2,0(R0)            LOAD TABLE ENTRY COUNT
         LTR   R2,R2               TEST ENTRY COUNT GREATER THAN ZERO
         BNZ   BEGINS3             YES, CONTINUE PROCESSING
*
** INVALID NETID REGISTRATION TABLE - SET TABLE INDICATOR
**
** NO VALID NETID ENTRIES:
**
**    X'FF' - INDICATE TO OTHER SME FUNCTIONS DO NOT
**            PERMIT CROSS NETWORK SESSIONS TO SETUP
*
BEGIN2C  EQU   *
         MVI   0(R10),X'FF'        INVALID NETID TABLE - SET TABLE INDICATOR
*
** OBTAIN STORAGE FOR ALIAS NAME POOL
*
BEGINS3  EQU   *
         GETMAIN R,LV=4096         OBTAIN ALIAS NAME POOL STORAGE
         LTR   R15,R15             DETERMINE IF GETMAIN SUCCESSFUL
         BZ    BEGIN3A             YES, CONTINUE PROCESSING
         MVI   32(R10),X'FF'       NO ALIAS NAME POOL - SET INDICATOR
         B     BEGINS4             YES, CONTINUE PROCESSING
*
** SAVE ADDRESS OF USER DATA STORAGE IN USER DATA FIELD
*
BEGINS3A EQU   *
         ST    R1,32(R10)          SAVE ALIAS NAME POOL STORAGE ADDRESS
         LA    R0,16               INITIALIZE COUNTER TO CLEAR STORAGE
BEGINS3B EQU   *
         XC    0(32,R1),0(R1)      CLEAR ALIAS NAME POOL STORAGE
         LA    R1,256(R1)          INCREMENT ADDRESS TO NEXT 256 STORAGE BYTES
         BCT   R0,BEGINS3B         CONTINUE PROCESSING ALL STORAGE
*
```

```
** SET SME EXIT OPTIONS FOR SUBSEQUENT SME FUNCTIONS
*
BEGINS4  EQU   *
         L     R2,12(R11)         PARAMETER LIST - EXIT OPTIONS ADDRESS
         OI    0(R2),X'40'        SME OPTION - SECONDARY AUTHORIZATION
         OI    0(R2),X'10'        SME OPTION - GATEWAY PATH SELECTION
         OI    1(R2),X'20'        SME OPTION - ALIAS NAME TRANSLATION
         OI    0(R2),X'08'        SME OPTION - END FUNCTION
         B     RETURN             RETURN TO VTAM
***********************************************************************
```

Returning to VTAM

Upon completion of these tasks, the begin function returns to VTAM. It is not executed again until VTAM is reinitialized. The code that performs the return to VTAM must free the storage used as the register save area for the session management exit routine, reestablish the registers for VTAM, and ensure the return is properly set. All function routines can use the following code to exit to VTAM.

```
***********************************************************************
*
** RETURN TO VTAM
*
RETURN   EQU   *
         L     R2,4(R13)          RELOAD VTAM'S REGISTER SAVE AREA ADDRESS
         ST    R15,16(R2)         SET SME FUNCTION RETURN CODE IN VTAM'S REGISTER 15
         FREEMAIN R,LV=72,A=(R13) FREE REGISTER SAVE AREA STORAGE
         LR    R13,R2             RESET REGISTER 13 TO VTAM'S SAVE AREA ADDRESS
         LM    R14,R12,12(R13)    RESTORE VTAM'S REGISTERS
         BR    R14                EXIT TO VTAM
***********************************************************************
```

Secondary Authorization Function

Since the session management exit routine does not use the initial authorization function, all LU sessions are authorized by the secondary authorization function. This sample session management exit routine function examines the NETID of the secondary and primary logical units (SLU and PLU).

If the NETID associated with both LUs is the same as the host's (GWVTAM), which is passed to the routine in the environment vectors, the session setup is allowed to continue. However, if the NETID of either the SLU or the PLU differs from the host's NETID, then the NETID registration table is examined.

If the table indicator in the first byte of the user data storage area is set to hex FF, the cross-network session setup is rejected. If the table indicator is not set to that value, the NETID of the appropriate LU is compared to the entries in the NETID registration table to determine if the request is valid. If the NETID is not contained in the table, the session request is rejected.

The secondary authorization function parameter list contains the addresses of the PLU and the SLU resource identification control vectors (RICs). These RICs contain the network identifier associated with both the real and alias LU names.

Using the sample network environment shown in Figure 12 on page 114, assume there is a cross-network session request from a terminal (N02L0256) in NET1 for a destination logical unit (A50CICS) in NET2. A sample portion of the PLU RIC that is passed to the session management exit routine in the gateway VTAM (G10VTAM) for this terminal-initiated session is shown in Figure 15 on page 122.

DEC(HEX) Offset	
0 (00)	19 (VECTOR KEY)
1 (01)	nn (VECTOR LENGTH)
2 (02)	00 (REROUTE COUNT)
3 (03)	40 (USAGE INDICATOR)
4 (04)	07 (SSCPNAME LENGTH)
5 (05)	V50VTAM (SSCPNAME)
12 (0C)	04 (REAL NETID LENGTH)
13 (0D)	NET2 (REAL NETID NAME)
17 (11)	07 (REAL LU NAME LENGTH)
18 (12)	A50CICS (REAL LU NAME)
25 (19)	04 (ALIAS NETID LENGTH)
26 (1A)	NET1 (ALIAS NETID NAME)
30 (1E)	07 (ALIAS LU NAME LENGTH)
31 (1F)	A50CICS (ALIAS LU NAME)
32 (20)	1A (CONTROL VECTOR KEY)

Figure 15. Sample Portion of PLU Resource Identification Control Vector

RICs contain more information related to the LU session request than is used by the sample session management exit routine. For example, the PLU and SLU RICs contain the real and alias names of the LU session partners. The session management exit routine can be expanded to authorize session setup based on the LU name of the DLU or OLU partner. The session management exit routine might analyze a specific naming convention associated with the LUs, or the NETID registration table might be expanded to include LU name registration, depending upon the security requirements of the network or application session partners.

After it receives the RIC, the session management exit routine must locate the NETID of the gateway VTAM host. This information is the first entry following the vector list header.

```
************************************************************************
*       SME ROUTINE - SECONDARY AUTHORIZATION FUNCTION
************************************************************************
SECAUTH  EQU   *
         L     R2,0(R11)           ADDRESS OF ENVIRONMENT VECTORS
         CLI   3(R2),X'06'         ENSURE HOST NETID VECTOR
         BNE   ABEND               TERMINATE - NO NETID CONTROL VECTOR
         IC    R3,2(R2)            LENGTH OF HOST NETID
         LTR   R3,R3               ENSURE HOST NETID AVAILABLE
         BZ    ABEND               TERMINATE - NO HOST NETID START OPTION
         LA    R2,4(R2)            ADDRESS OF HOST NETID
************************************************************************
```

The next task is to examine the network identifiers that are passed for the PLU and SLU in the RICs. There are two NETIDs in each RIC, one associated with the real name and the other with the alias name of the LU.

In this sample, the only network identifier that is important is the real NETID for each LU. The real NETID in each RIC must be examined to determine the direction of session setup. The direction of session setup is determined by which LU initiates the session, or the mechanism used to initiate the session (terminal-initiated logon, automatic logon, or third party logon), or both the LU and mechanism used.

The usage indicator (byte 4, bit 1) in the RIC identifies whether the resource, either PLU or SLU, is the target. Therefore, this indicator can be checked to determine whether the RIC for the PLU is the DLU or the OLU. The same task can be performed for the SLU by examining the indicator in the RIC.

The following sample code is used to examine the real NETID in the RICs for the primary and secondary LUs. The NETID in the RIC for the PLU is first compared to the host NETID. If both are the same, the real NETID in the RIC for the SLU must be examined. If the PLU and host NETIDs are not the same, the real NETID for the PLU is compared to the entries in the NETID registration table. If there is no match, the session request is considered to be from an invalid network and the session is rejected.

Note: In the sample, an execute (EX) instruction is used to compare the NETIDs and the PLU or the SLU names because the name length in the RIC appears to be variable. VTAM, however, always uses 8-character names that are padded with blanks. Therefore, you are not required to use the EX instruction. You can use a compare (CLC) instruction with a length of eight to perform this task.

```
************************************************************************
*
** LOCATE PLU RESOURCE INFORMATION CONTROL VECTOR - REAL NETID
*
SECAUS1  EQU   *
         LA    R15,0               RETURN CODE - ASSUME ACCEPT SESSION REQUEST
         L     R4,12(R11)          PLU RIC ADDRESS
         LA    R4,4(R4)            PLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)            PLU RIC SSCPNAME LENGTH
         AR    R4,R5               PLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)            PLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R4)            PLU RIC NETID LENGTH
         LTR   R5,R5               PLU RIC TEST NETID FOR ZERO LENGTH
         BZ    ABEND              PLU RIC NETID LENGTH INCORRECT
         BCTR  R5,0               PLU RIC NETID LENGTH MINUS ONE
         LA    R4,1(R4)           PLU RIC NETID ADDRESS
*
** COMPARE THE PLU RIC NETID TO HOST NETID IN THE ENVIRONMENT VECTOR
** IF PLU RIC NETID EQUALS HOST NETID THEN PROCESS SLU RIC NETID
** ELSE EXAMINE NETID REGISTRATION TABLE FOR VALID NETWORK IDENTIFIER
*
         EX    R5,NETID           COMPARE PLU RIC REAL NETID TO HOST NETID
         BNE   SECAUS3            PLU NETID NOT EQUAL GATEWAY VTAM
*
** LOCATE SLU RESOURCE INFORMATION CONTROL VECTOR - REAL NETID
*
SECAUS2  EQU   *
         L     R4,16(R11)         SLU RIC ADDRESS
         LA    R4,4(R4)           SLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)           SLU RIC SSCPNAME LENGTH
         AR    R4,R5              SLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R5)           SLU RIC NETID LENGTH
         LTR   R5,R5              SLU RIC TEST NETID FOR ZERO LENGTH
         BZ    ABEND              SLU RIC NETID LENGTH INCORRECT
         BCTR  R5,0               SLU RIC NETID LENGTH MINUS ONE
         LA    R4,1(R4)           SLU RIC NETID ADDRESS
*
** COMPARE THE SLU RIC NETID TO HOST NETID IN THE ENVIRONMENT VECTOR
** IF SLU RIC NETID EQUALS HOST NETID THEN MUST BE SAME NETWORK SESSION
** ELSE EXAMINE NETID REGISTRATION TABLE FOR VALID NETWORK IDENTIFIER
*
         EX    R5,NETID           COMPARE PLU RIC REAL NETID TO HOST NETID
         BE    RETURN             SLU NETID EQUALS GATEWAY VTAM - SAME NETWORK
*
** EXAMINE NETID REGISTRATION TABLE FOR VALID NETWORK IDENTIFIER
*
SECAUS3  EQU   *
         L     R2,8(R11)          USER DATA FIELD ADDRESS
         L     R2,0(R2)           USER DATA STORAGE AREA ADDRESS
         L     R2,4(R2)           NETID REGISTRATION TABLE ADDRESS
         L     R3,0(R2)           NETID TABLE ENTRY COUNT
         L     R2,4(R2)           NETID TABLE ENTRY ADDRESS
*
** COMPARE THE PLU OR SLU RIC REAL NETID TO NETID REGISTRATION TABLE ENTRY
*
SECAUS3A EQU   *
         EX    R5,NETID           COMPARE PLU/SLU RIC NETID TO TABLE ENTRY
         BE    RETURN             ACCEPT SESSION REQUEST - VALID NETID ENTRY
SECAUS3B EQU   *
         LA    R2,8(R2)           NEXT NETID TABLE ENTRY ADDRESS
         BCT   R3,SECAUS3A        NETID TABLE ENTRY COUNT NOT ZERO - CONTINUE SEARCH
*
** END NETID REGISTRATION TABLE SEARCH - NO VALID ENTRY
** RETURN CODE - REJECT CROSS NETWORK SESSION REQUEST
*
SECAUS3C EQU   *
         LA    R15,8              NO ENTRY IN NETID REGISTRATION TABLE
         B     RETURN             EXIT TO VTAM
*
```

```
** INSTRUCTION EXECUTED TO COMPARE: PLU RIC REAL NETID TO HOST NETID
**                                  PLU RIC REAL NETID TO NETID REGISTRATION TABLE
**                                  SLU RIC REAL NETID TO HOST NETID
**                                  SLU RIC REAL NETID TO NETID REGISTRATION TABLE
*
NETID    CLC   0(0,R4),0(R2)      NETID COMPARE INSTRUCTION
*
************************************************************************
```

Gateway Path Selection Function for MVS and VM

The gateway path selection function can be used when a SNA network interconnection environment uses multiple gateway NCPs. The gateway VTAM always attempts to establish a cross-network session using the same gateway NCP over which the CDRM-CDRM session is established. The session management exit routine can modify this default gateway path selection. The gateway path selection function can either reorder the list of available gateway NCPs or delete entries from the list.

Each entry in the list is associated with a GWPATH definition statement coded following a CDRM definition statement. For example, assume the following CDRM and GWPATH definition statements are coded in the gateway VTAM (G10VTAM) in NET2:

```
VTAM01   CDRM
GWNCP30  GWPATH SUBAREA=30,ADJNET=NET1,ADJNETSA=1,ADJNETEL=1
GWNCP40  GWPATH SUBAREA=40,ADJNET=NET1,ADJNETSA=1,ADJNETEL=1
```

Assuming that both gateway NCPs (GWNCP30 and GWNCP40) are operative and all of the links are active, the first gateway NCP establishes the SSCP-SSCP session between the gateway VTAM (G10VTAM) and the VTAM host (VTAM01) in NET1.

The list of GWPATH definition statements passed to the session management exit routine is in the order in which they are coded. The gateway NCP that establishes the SSCP-SSCP session, however, is always placed first in the list. Therefore, in the example above, if GWNCP30 was not initially available for the SSCP-SSCP session, but was operative later, the order of the GWPATH entries is reversed in the list.

Even though the gateway NCP is operative, links between the gateway NCP and the adjacent subarea may not be operative. As a result, a specific gateway NCP is selected to setup an LU session; however, the route from gateway NCP to the destination subarea is not available.

If the VTAM in NET1 (VTAM01) is gateway capable only, the gateway path selection function is invoked with the list of gateway paths as described above. However, if VTAM01 is a gateway SSCP, and a gateway path selection function in that VTAM orders the list of gateway paths, then the entire list of gateway paths is not passed to the session management exit in G10VTAM. Only the first entry in the list that was selected by the VTAM01 and passed in the CDINIT is supplied to the gateway path selection function in G10VTAM.

Examining LU Names

In the sample network, assume that GWNCP40 is selected for the session between the terminal (N02L0256) and the application (A50CICS). Route activation for this session does not take place until the BIND for the session flows. In this case, the BIND flows from the VTAM host (V50CDRM) to the gateway NCP (GWNCP40). Assume that the link is not operative between the gateway NCP (GWNCP40) and the NCP subarea (NCP02) in NET1. As a result, the virtual route (VR) cannot be activated and the LU-LU session fails. There is no information provided to the session management exit routine indicating that the route is unavailable, even though the gateway NCP is available. If the end user at the terminal attempts to re-initiate the session, the session management exit routine, using the same logic, continues to select the same gateway NCP over which the complete network route is unavailable.

In the sample portion of the session management exit routine, the gateway path selection function performs the following logic to select a GWPATH from the list:

1. Examines the real LU name of the application program in the PLU and the SLU RIC to determine if the LU name is either A50CICS or A60CICS. If the DLU or OLU is neither of these names, the session management exit code uses the default gateway path selection.
2. If the real LU name is A50CICS, ensures that GWNCP30 is the first entry in the list. If the LU name is A60CICS, reorders the list, if necessary, to ensure that GWNCP40 is the first entry in the GWPATH selection list

The following code locates the real LU name for the PLU or SLU. If the LU name is not a CICS application, control is returned to VTAM without modifying the GWPATH list.

```
***********************************************************************
*        SME ROUTINE - GATEWAY PATH SELECTION FUNCTION
***********************************************************************
GWPATH   EQU   *
*
** LOCATE USER DATA STORAGE AREA - GWPATH NAME AND SUBAREA ADDRESS
*
GWPTHS1A EQU   *
         L     R1,8(R11)          USER DATA FIELD ADDRESS
         L     R1,0(R1)           USER DATA STORAGE AREA ADDRESS
*
** LOCATE PLU RESOURCE INFORMATION CONTROL VECTOR – REAL LUNAME
*
GWPTHS1B EQU   *
         L     R4,12(R11)         PLU RIC ADDRESS
         LA    R4,4(R4)           PLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)           PLU RIC SSCPNAME LENGTH
         AR    R4,R5              PLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R4)           PLU RIC NETID LENGTH
         AR    R4,R5              PLU RIC REAL LUNAME ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC REAL LUNAME LENGTH ADDRESS
         IC    R5,0(R4)           PLU RIC REAL LUNAME LENGTH
         BCTR  R5,0               PLU RIC REAL LUNAME LENGTH MINUS ONE
         LA    R4,1(R4)           PLU RIC REAL LUNAME ADDRESS
*
```

```
** COMPARE THE PLU RIC LUNAME TO APPLICATION NAME: A50CICS OR A60CICS
*
         EX    R5,CICS50          COMPARE PLU RIC REAL LUNAME TO ACICS50
         BE    GWPTHS2A           YES, REORDER GWPATH AS NECESSARY
         EX    R5,CICS60          COMPARE PLU RIC REAL LUNAME TO ACICS60
         BE    GWPTHS2B           YES, REORDER GWPATH AS NECESSARY
*
** LOCATE SLU RESOURCE INFORMATION CONTROL VECTOR - REAL LUNAME
*
GWPTHS1C EQU   *
         L     R4,16(R11)         SLU RIC ADDRESS
         LA    R4,4(R4)           SLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)           SLU RIC SSCPNAME LENGTH
         AR    R4,R5              SLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R5)           SLU RIC NETID LENGTH
         AR    R4,R5              SLU RIC REAL LUNAME ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC REAL LUNAME LENGTH ADDRESS
         IC    R5,0(R4)           SLU RIC REAL LUNAME LENGTH
         BCTR  R5,0               SLU RIC REAL LUNAME LENGTH MINUS ONE
         LA    R4,1(R4)           SLU RIC REAL LUNAME ADDRESS
*
** COMPARE THE SLU RIC LUNAME TO APPLICATION NAME: A50CICS OR A60CICS
*
         EX    R5,CICS50          COMPARE PLU RIC REAL LUNAME TO ACICS50
         BE    GWPTHS2A           YES, REORDER GWPATH AS NECESSARY
         EX    R5,CICS60          COMPARE PLU RIC REAL LUNAME TO ACICS60
         BE    GWPTHS2B           YES, REORDER GWPATH AS NECESSARY
*
** IF NEITHER THE PLU OR SLU RIC REAL LUNAME EQUALS A50CICS OR A60CICS
** THEN SET RETURN CODE TO PROCEED WITH DEFAULT GWPATH SELECTION LIST
*
         LA    R15,4              RETURN CODE TO USE DEFAULT GWPATH LIST
         B     RETURN             RETURN CONTROL TO VTAM
*
** PLU OR SLU RIC REAL LUNAME EQUALS A50CICS -
** SAVE FIRST GWPATH ENTRY NAME AND SUBAREA ADDRESS IN USER DATA STORAGE
*
GWPTHS2A EQU   *
         MVC   8(8,R1),GWNCP30    FIRST GWPATH NAME MUST BE GWNCP30
         LA    R5,30              FIRST GWPATH SUBAREA MUST BE 30
         ST    R5,16(R1)          SAVE IN USER DATA STORAGE
         B     GWPTHS3            CONTINUE, REORDER GWPATH LIST IF NECESSARY
*
** PLU OR SLU RIC REAL LUNAME EQUALS A60CICS -
** INITIALIZE REGISTERS TO MODIFY GWPATH SELECTION LIST AS NECESSARY
*
GWPTHS2B EQU   *
         MVC   8(8,R1),GWNCP40    FIRST GWPATH NAME MUST BE GWNCP40
         LA    R5,40              FIRST GWPATH SUBAREA MUST BE 40
         ST    R5,16(R1)          SAVE IN USER DATA STORAGE
         B     GWPTHS3            CONTINUE, REORDER GWPATH LIST IF NECESSARY
*
```

```
** INSTRUCTIONS EXECUTED TO COMPARE: PLU RIC REAL LUNAME TO A50CICS OR A60CICS
**                                  SLU RIC REAL LUNAME TO A50CICS OR A60CICS
*
CICS50   CLC   0(0,R4),C'A50CICS'  LUNAME COMPARE INSTRUCTION
CICS60   CLC   0(0,R4),C'A60CICS'  LUNAME COMPARE INSTRUCTION
*
** CONSTANTS - GATEWAY NCP NAMES                                       0CICS
*
GWNCP30  DC    CL8'GWNCP30'         GATEWAY NCP NAME - GWNCP30
GWNCP40  DC    CL8'GWNCP40'         GATEWAY NCP NAME - GWNCP40
*
**********************************************************************
```

Modifying Gateway Path Selection List

After the session management exit routine determines that the real LU name is associated with CICS, the gateway path selection function reorders the gateway path selection list as appropriate. If the application program name is A50CICS, the first entry in the list must be GWNCP30. Conversely, if the LU name is A60CICS, the first GWPATH entry must be GWNCP40. The above sample code locates the LU name in the RIC and, if the proper CICS name is determined, branches to the sample code below. In this case, the user data storage area is initialized with the gateway NCP name and the corresponding subarea address of the first entry in the list.

The session management exit routine must next locate the GWPATH selection list. Since only one gateway NCP may be available, the gateway path function must determine if the list contains more than one entry. If there is only one entry in the list, the exit routine returns to VTAM and uses the default list. However, if there is more than one entry in the gateway selection list, the code must read the first entry and reorder the list, if necessary.

The gateway path selection list is reordered by moving the first entry name and subarea address into user data storage and moving the preferred gateway NCP name and subarea address to the first entry in the list. The entry which was previously the first entry in the list must then be moved from user data storage back into the gateway path list to complete the swap.

If the gateway path NCP name (GWN) is blank, the preferred gateway NCP is inactive. In that case, the first entry in the selection list must be reset to the original name and subarea address.

Note: In the sample, the GWN is checked to determine if the preferred gateway NCP is inactive because the SUBAREA operand was used on the GWPATH definition statement. If you use the GWN operand instead of the SUBAREA operand, the subarea address is set to zero, indicating that the preferred gateway path is inactive. If both operands are coded, there is no means of determining if that gateway is inactive.

The following sample session management exit code performs these tasks. Figure 16 on page 129 also illustrates the contents of the entries in the gateway path selection list for the sample network environment.

DEC(HEX) Offset	
0 (0)	nn (TOTAL LENGTH)
2 (2)	NET1 (with 4 blanks)
10 (A)	0002
12 (C)	14
13 (D)	GWNCP30 (with 1 blank)
21 (15)	0000001E
25 (19)	NET1 (with 4 blanks)
33 (21)	14
34 (22)	GWNCP40 (with 1 blank)
21 (15)	00000028
25 (19)	NET1 (with 4 blanks)

Figure 16. Gateway Path Selection List for Sample Session Management Exit Routine

```
***********************************************************************
GWPTHS3  EQU   *
         LA    R15,4              RETURN CODE ASSUMES DEFAULT GWPATH LIST
         L     R2,28(R11)         GATEWAY PATH SELECTION LIST ADDRESS
         LH    R3,10(R2)          OBTAIN THE NUMBER OF GWPATH ENTRIES
         LA    R2,12(R2)          FIRST GWPATH ENTRY ADDRESS
         LA    R4,1               INITIALIZE REGISTER WITH VALUE OF ONE (1)
         CR    R3,R4              GWPATH ENTRIES GREATER THAN ONE
         BH    GWPTHS3A           YES, LOCATE NAME OF FIRST GWPATH ENTRY
         B     RETURN             RETURN CONTROL TO VTAM
*
** EXAMINE FIRST GWPATH SUBAREA ADDRESS (SUBAREA OPERAND)
**
** COMPARE SUBAREA ADDRESS ENTRY TO PREFERRED ADDRESS IN USER DATA STORAGE
** IF BOTH ENTRIES EQUAL RETURN TO VTAM AND USE DEFAULT SELECTION LIST
*
GWPTHS3A EQU   *
         CLC   9(4,R2),16(R1)     FIRST GWNCP SAME AS PREFERRED GWPATH
         BE    RETURN             PREFERRED GWNCP FIRST, USE DEFAULT LIST
*
** FIRST GWPATH ENTRY NOT EQUAL TO PREFERRED GWPATH
** SAVE CURRENT FIRST ENTRY IN SELECTION LIST IN USER DATA STORAGE
** SWAP PREFERRED ENTRY WITH FIRST ENTRY IN SELECTION LIST
*
GWPTHS3B EQU   *
         MVC   20(8,R1),1(R2)     SAVE GWNAME IN USER DATA STORAGE
         MVC   28(4,R1),9(R2)     SAVE SUBAREA ADDRESS IN USER DATA
*                                   STORAGE
```

```
         MVC   1(8,R2),8(R1)      SWAP PREFERRED GWNAME FROM USER DATA
*                                   STORAGE
         MVC   9(4,R2),16(R1)     SWAP PREFERRED SUBAREA ADDRESS FROM
*                                   USER DATA STORAGE
*
** LOCATE PREFERRED GWPATH ENTRY IN SELECTION LIST TO COMPLETE SWAP
**
** COMPARE THE NEXT SUBAREA ADDRESS ENTRY IN THE GWPATH SELECTION LIST
** TO THE PREFERRED ENTRY IN USER DATA STORAGE. IF EQUAL, ENSURE THE
** PREFERRED GWPATH IS NOT INACTIVE. COMPARE THE GWNCP NAME TO BLANKS.
** IF ACTIVE, COMPLETE THE GWPATH SWAP, OTHERWISE RESET THE FIRST ENTRY
** TO THE ORIGINAL CONDITION.
*
GWPTHS3C EQU   *
         IC    R4,0(R2)           OBTAIN GWPATH ENTRY LENGTH
         AR    R2,R4              NEXT GWPATH ENTRY ADDRESS
         CLC   9(4,R2),16(R1)     GWPATH SUBAREA ADDRESS USED AS FIRST
*                                   ENTRY
         BNE   GWPTHS3D           NO, EXAMINE NEXT ENTRY
         CLC   1(8,R2),XL8'40'    CHECK IF GWPATH NAME BLANK
         BE    GWPTHS3E           PREFERRED GWNCP NOT ACTIVE, RESET
*                                   FIRST ENTRY
         MVC   1(8,R2),20(R1)     SWAP FIRST ENTRY GWNAME FROM USER
*                                   DATA STORAGE
         MVC   9(4,R2),28(R1)     SWAP FIRST ENTRY SUBAREA ADDRESS
*                                   FROM USER DATA STORAGE
         LA    R15,0              RETURN CODE, USE MODIFIED GWPATH LIST
         B     RETURN             RETURN CONTROL TO VTAM
*
** EXAMINE NEXT ENTRY IN GWPATH SELECTION LIST
**
** IF LIST EXHAUSTED AND PREFERRED GWPATH ENTRY NOT LOCATED THEN
** SWAP FIRST ENTRY BACK TO ORIGINAL GWPATH BECAUSE THE PREFERRED
** GWPATH ENTRY IS NOT AVAILABLE IN THE LIST.
*
GWPTHS3D EQU   *
         BCT   R3,GWPTH3C         LAST GWPATH ENTRY, NO CONTINUE PROCESS
GWPTHS3E EQU   *
         L     R2,28(R11)         GATEWAY PATH SELECTION LIST ADDRESS
         LA    R2,12(R2)          FIRST GWPATH ENTRY ADDRESS
         MVC   1(8,R2),20(R1)     SWAP FIRST ENTRY GWNAME FROM USER
*                                   DATA STORAGE
         MVC   9(4,R2),28(R1)     SWAP FIRST ENTRY SUBAREA ADDRESS
*                                   FROM USER DATA STORAGE
         B     RETURN             USE ORIGINAL DEFAULT GWPATH LIST
***********************************************************************
```

Alias Selection Function for MVS and VM

The alias selection function of the session management exit routine performs various name translation tasks. VTAM maintains both a real and an alias network-qualified name for each LU session partner. VTAM can invoke this function to determine any one of the following four names:

- Real network-qualified name for the PLU or DLU
- Alias network-qualified name for the PLU or DLU
- Real network-qualified name for the SLU or OLU
- Alias network-qualified name for the SLU or OLU.

In addition, the alias selection function can map a class-of-service (COS) name for one network to the COS name used in another network. It can also map logon mode entry names or supply the SSCP owner for a particular resource name.

In this sample, the alias function translates the real LU name of a terminal in NET1 into an alias name in NET2. The primary applications (CICS and IMS) in NET2 require the terminal names to be generated in these subsystems. All terminals in NET1 that require access to these applications must be generated into the subsystems. Since the subsystem maintenance and coordination associated with a large quantity of other-network terminals can be difficult, the alias function translates a real terminal name into an alias terminal name.

The alias terminal name is selected from a preassigned pool of alias names. This pool can be pre-generated into the IMS and CICS applications. When all the names in the pool are allocated to concurrent cross-network sessions, subsequent session requests fail. Once a name is freed as a result of a session termination, another cross-network session can be established. Therefore, the preassigned alias name pool needs to contain sufficient entries to accommodate the peak concurrent cross-network session count for CICS and IMS.

The session management exit routine must first examine the indicators associated with the LU names to determine what translation is necessary. For example, assume there is a terminal-initiated session from N02L0256 for the application program A70IMS in NET2. There are four network qualified names associated with this session request, as shown in Figure 17.

	Origin Logical Unit	Destination Logical Unit
Real Name	NET1.N02L0256	NET2.A70IMS
Alias Name	NET2.ALIASnnn	NET1.A70IMS

Where nnn is the suffix added to the alias terminal name allocated from the alias name pool for terminals in NET1.

Figure 17. Real and Alias Network-Qualified Names

Translating DLU Name

For a terminal-initiated session using CDINIT processing (or DSRLST), the first entry in the alias input parameter list is the DLU, A70IMS. The alias function examines the OLU/DLU status indicator and the DLU name that requires translation. If the DLU name is a NET2 application program (IMS or CICS), the same LU name is used for the real LU name and only the NETID of NET2 needs to be supplied.

The session management exit routine initializes the output fields to a value of A70IMS for the LU name (RTNNAME1) and NET2 for the NETID (RTNNET1). However, if the first entry is not the DLU, or the LU name is not one of the NET2 application programs, the direction of session setup does not permit the alias function to provide an alias name; that is, a session initiated from the CICS or IMS application requires the real name of the terminal. This sample session management exit routine cannot map an alias terminal that is pre-generated in IMS or CICS into a specific terminal name in the attached network (NET1). Therefore, this sample session management exit routine returns control to VTAM and indicates that no translation is performed. The session setup proceeds assuming the correct names are supplied. The following sample code processes the first entry in the alias input parameter list:

```
***********************************************************************
*        SME ROUTINE - ALIAS SELECTION
***********************************************************************
ALIAS    EQU   *
*
** LOCATE THE ALIAS PARAMETER LIST - INPUT AND OUTPUT ADDRESSES
*
ALIASS1  EQU   *
         L     R2,44(R11)         ALIAS INPUT PARAMETER LIST ADDRESS
         LA    R2,8(R2)           ALIAS FIRST LUNAME ADDRESS - REAL/ALIAS NAME
         L     R3,48(R11)         ALIAS OUTPUT PARAMETER LIST ADDRESS
*
** PROCESS FIRST LOGICAL UNIT NAME IN ALIAS INPUT PARAMETER LIST:
**
** IF FIRST ENTRY IS THE DLU AND NAME IS 'A50CICS', 'A60CICS', OR 'A70IMS'
** RETURN SAME NAME TO VTAM AND CHANGE THE NETID TO 'NET2', OTHERWISE,
** EXIT TO VTAM WITHOUT ALIAS TRANSLATION.
**
** REASONS: 1. TERMINAL INITIATED SESSION TO APPLICATION WHICH DOES NOT
**             REQUIRE ALIAS TRANSLATION.
**          2. APPLICATION INITIATED SESSION FOR WHICH SME EXIT CANNOT
**             MAP THE ALIAS NAME TO A REAL TERMINAL NAME
**          3. CDINIT-OTHER PROCESSING FOR WHICH SME EXIT DOES NOT
**             PROVIDE NECESSARY TRANSLATION.
*
ALIASS1A EQU   *
         TM    1(R2),X'20'        FIRST ENTRY - DLU NAME FOR CDINIT/DSRLST
*                                   PROCESSING
         BNO   ALIASS1B           NO, ALIAS TRANSLATION NOT PERFORMED
         CLC   8(R2),CL8'A50CICS' ALIAS NAME EQUAL A50CICS APPLICATION
         BE    ALIASS1C           YES, RETURN SAME LUNAME AND NET2 NETID
         CLC   8(R2),CL8'A60CICS' ALIAS NAME EQUAL A60CICS APPLICATION
         BE    ALIASS1C           YES, RETURN SAME LUNAME AND NET2 NETID
         CLC   8(R2),CL8'A70IMS'  ALIAS NAME EQUAL A70IMS APPLICATION
         BE    ALIASS1C           YES, RETURN SAME LUNAME AND NET2 NETID
ALIASS1B EQU   *
         LA    R15,0              NO TRANSLATION PERFORMED - NO ALIAS
*                                   APPLICATION
         B     RETURN             RETURN TO VTAM
ALIASS1C EQU   *
         MVC   0(8,R3),8(R2)      REAL DLU NAME EQUALS ALIAS APPLICATION NAME
         MVC   48(8,R3),CL8'NET2' REAL NETID FOR DLU EQUALS NET2
         B     ALIASS2            CONTINUE PROCESSING, SECOND ALIAS ENTRY
***********************************************************************
```

Translating OLU Name

The next step is to translate the OLU name. The indicator for the OLU fields (NAME2) specifies that the real name is provided and an alias name must be returned. The sample exit routine allocates up to 999 names. The names consist of a required prefix, "ALIAS," and a 3-digit suffix allocated from an alias-suffix table. This table is initialized and maintained in a user storage area by the exit routine. An alias table can be loaded in the begin function similar to the NETID registration table. The sample code below examines the second entry of the alias input parameters and invokes a subroutine to allocate an alias name from the table.

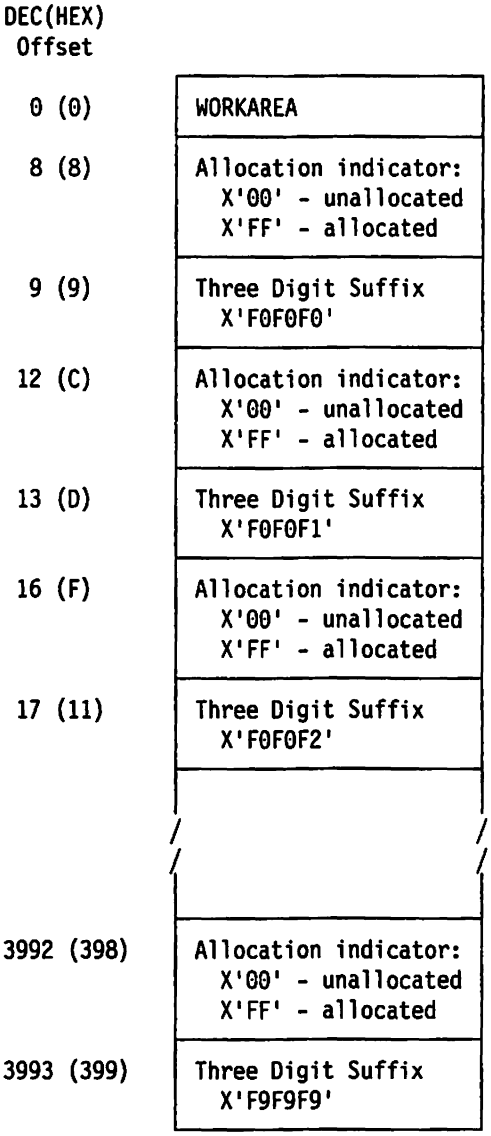
Note: For this example you must set the CDRSCTI start option to zero so that the alias LU name assigned to the terminal can be deleted and reassigned to another real terminal once the session is terminated. CDRSCTI may have a value other than zero for real operations.

```
************************************************************************
ALIASS2  EQU   *
         LA    R2,32(R2)          ALIAS SECOND LUNAME ADDRESS - REAL/ALIAS NAME
*
** PROCESS SECOND LOGICAL UNIT NAME IN ALIAS INPUT PARAMETER LIST:
**
** IF SECOND ENTRY IS THE REAL NAME OF THE TERMINAL, THEN
** ALLOCATE AN ALIAS NAME FROM THE ALIAS NAME POOL, OTHERWISE,
** EXIT TO VTAM WITHOUT ALIAS TRANSLATION.
**
** REASON: 1. THE TERMINAL IS THE OLU AND THE NAME REQUESTED BY VTAM
**            IS THE REAL NAME FOR THE OLU AND THE SME ROUTINE ONLY
**            PROVIDES ALIAS NAMES.
*
ALIASS2A EQU   *
         TM    2(R2),X'80'        SECOND ENTRY - REAL NAME SUPPLIED
         BO    ALIASS2C           YES, PERFORM ALIAS TRANSLATION
ALIASS2B EQU   *
         LA    R15,8              SOME TRANSLATION PERFORMED - NO ALIAS
*                                   APPLICATION
         B     RETURN             RETURN TO VTAM
ALIASS2C EQU   *
*
** IF THE INDICATOR IN THE FIRST BYTE OF THE ALIAS STORAGE ADDRESS IN
** THE USER DATA STORAGE AREA IS X'FF', THEN STORAGE WAS NOT AVAILABLE
** FOR THE TABLE. NO ALIAS PROCESSING CAN BE PERFORMED, THEREFORE EXIT.
**
** OTHERWISE, BRANCH TO SUBROUTINE THAT ALLOCATES AN AVAILABLE ALIAS NAME
** SUFFIX FROM THE ALIAS NAME POOL. IF ALL ENTRIES ARE USED, REGISTER 15 (R15)
** CONTAINS A NON-ZERO RETURN CODE FOR VTAM WHICH DOES NOT ALLOW SESSION
** INITIATION TO CONTINUE. OTHERWISE, THE ADDRESS FOR THE ALIAS NAME SUFFIX
** IS ADDRESSABLE AT DISPLACEMENT ONE (1) USING REGISTER 1 (R1). THE SUFFIX
** IS APPENDED TO THE FIVE CHARACTER STRING 'ALIAS' TO FORM THE ALIAS NAME
** FOR THE TERMINAL.
*
         L     R4,8(R11)          PARAMETER LIST - USER DATA ADDRESS
         L     R4,0(R4)           USER DATA STORAGE ADDRESS
         CLI   32(R4),X'FF'       EXAMINE IF ALIAS NAME STORAGE OBTAINED
         BNE   ALIAS2E            YES, CONTINUE ALIAS NAME PROCESSING
ALIASS2D EQU   *
         LA    R15,20             NO ALIAS FUNCTION AVAILABLE - FAIL SESSION
*                                   SETUP
```

```
         B      RETURN             RETURN TO VTAM - NO ALIAS APPLICATION
ALIASS2E EQU    *
         L      R4,32(R4)          ALIAS NAME STORAGE ADDRESS
         BAL    R14,ALIASS3        BRANCH TO ALIAS NAME ALLOCATION ROUTINE
         LTR    R15,R15            ALIAS SUFFIX AVAILABLE
         BNZ    RETURN             RETURN TO VTAM - NO ALIAS ENTRY AVAILABLE
ALIASS2C EQU    *
         MVC    8(5,R3),C'ALIAS'   ALIAS OUTPUT PARAMETER - ALIAS NAME PREFIX
         MVC    13(3,R3),1(R1)     ALIAS OUTPUT PARAMETER - ALIAS NAME SUFFIX
         MVC    56(8,R3),CL8'NET2' ALIAS OUTPUT PARAMETER - ALIAS NETID
         LA     R15,8              SOME TRANSLATION PERFORMED - NO COS, LOGMODE
         B      RETURN             RETURN TO VTAM - NO ALIAS APPLICATION
***********************************************************************
```

The subroutine that processes the alias suffix table initializes the storage, obtained in the begin function, during the first pass through the allocation process. During the first pass, no entry is allocated as indicated by a hex 00 in the first byte. After the entry is allocated, a hex FF is moved to the first byte in the alias suffix entry. A binary counter for the entry number is converted to a 3-byte decimal value, the suffix, and is saved in the remaining 3 bytes of the 4-byte entry. The address of the alias suffix entry is then returned to the alias function that invoked the subroutine. The suffix is appended to the prefix string "ALIAS" to form the alias name.

The format of the table is illustrated in Figure 18 on page 135. An example of the subroutine code that performs the suffix allocation follows Figure 18.



DEC(HEX) Offset	
0 (0)	WORKAREA
8 (8)	Allocation indicator: X'00' - unallocated X'FF' - allocated
9 (9)	Three Digit Suffix X'F0F0F0'
12 (C)	Allocation indicator: X'00' - unallocated X'FF' - allocated
13 (D)	Three Digit Suffix X'F0F0F1'
16 (F)	Allocation indicator: X'00' - unallocated X'FF' - allocated
17 (11)	Three Digit Suffix X'F0F0F2'
3992 (398)	Allocation indicator: X'00' - unallocated X'FF' - allocated
3993 (399)	Three Digit Suffix X'F9F9F9'

Figure 18. Alias Terminal Name Pool - Three Digit Suffix

```
***********************************************************************
ALIASS3  EQU   *
*
** ALIAS NAME SUFFIX ALLOCATION SUBROUTINE
**
** R1  = ADDRESS OF ALIAS NAME SUFFIX ENTRY ALLOCATED
** R4  = ADDRESS OF ALIAS NAME SUFFIX STORAGE - WORKAREA FOR CONVERT TO DECIMAL
** R6  = ALIAS SUFFIX ENTRY COUNTER - DECREMENTED
** R7  = ALIAS SUFFIX ENTRY COUNTER - CONSTANT
** R14 = RETURN ADDRESS - MAIN ROUTINE OF ALIAS FUNCTION
** R15 = RETURN CODE
*
         LA    R1,8(R4)            ALIAS SUFFIX ADDRESS - FIRST ENTRY
         LA    R6,1000             ALIAS SUFFIX ENTRY COUNTER - DECREMENT
         LA    R7,1000             ALIAS SUFFIX ENTRY COUNTER - CONSTANT
ALIASS3A EQU   *
```

```
         CLI   0(R1),X'00'        ALIAS SUFFIX ENTRY ALLOCATED
         BNE   ALIASS3B           YES, INCREMENT ADDRESS TO NEXT ENTRY
         MVI   0(R1),X'FF'        NO, ALLOCATE ALIAS SUFFIX ENTRY
         SR    R7,R6              CALCULATE ALIAS SUFFIX ENTRY VALUE - BINARY
         CVD   R7,0(R4)           CONVERT ALIAS SUFFIX ENTRY TO DECIMAL -
*                                   WORKAREA
         UNPK  1(3,R1),6(2,R4)    UNPACK SUFFIX VALUE INTO PROPER SUFFIX ENTRY
         OI    3(R1),X'F0'        ENSURE DECIMAL VALUE - CHANGE SIGN OF LAST
*                                   CHARACTER
         BR    R14                RETURN TO INVOKING ROUTINE
ALIASS3B EQU   *
         LA    R1,4(R1)           INDEX ALIAS SUFFIX ADDRESS TO NEXT ENTRY
         BCT   R6,ALIASS3A        DECREMENT ALIAS SUFFIX COUNTER - CONTINUE IF
*                                   NONZERO
         LA    R15,20             ALL ALIAS SUFFIX ENTRIES ALLOCATED - ENTRIES
*                                   EXHAUSTED
         BR    R14                RETURN TO INVOKING ROUTINE - TERMINATE
*                                   SESSION SETUP
***********************************************************************
```

Final Accounting Function

Since the alias selection function is not invoked at session termination, the session management exit routine can only allocate the alias suffix entries. Another session management exit function must deallocate the alias suffix entry when a session ends. The final accounting function, function code hex 03, performs this task.

The final accounting function examines the alias LU name in the RIC of the PLU and the SLU. If the LU name contains a five character prefix of "ALIAS," the session management exit routine can deallocate the proper alias suffix entry. This is done by converting the last three characters of the alias name to a binary value. That value is an index into the alias suffix storage area. The code then resets the allocation indicator from hex FF to hex 00.

The entire final accounting function is not included in this sample; however, the portion of code that deallocates the appropriate entry at session termination follows:

```
***********************************************************************
*
**       SME ROUTINE - FINAL ACCOUNTING FUNCTION - DEALLOCATE ALIAS
*                                                  SUFFIX ROUTINE
*
ALIASS4  EQU   *
         L     R1,8(R11)          USER DATA FIELD ADDRESS
         L     R1,0(R1)           USER DATA STORAGE AREA ADDRESS
         L     R1,32(R1)          ALIAS SUFFIX STORAGE ADDRESS
         LA    R15,0              NO FINAL ACCOUNTING RETURN CODE
         CLI   32(R1),X'FF'       TEST IF ALIAS SUFFIX STORAGE OBTAINED
         BE    RETURN             NO, ALIAS TRANSLATION NOT PERFORMED -
*                                   VTAM RETURN
         L     R1,32(R1)          ALIAS SUFFIX STORAGE ADDRESS
*
** LOCATE PLU RESOURCE INFORMATION CONTROL VECTOR - ALIAS LUNAME
*
ALIASS4A EQU   *
         L     R4,12(R11)         PLU RIC ADDRESS
         LA    R4,4(R4)           PLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)           PLU RIC SSCPNAME LENGTH
         AR    R4,R5              PLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R4)           PLU RIC NETID LENGTH
         AR    R4,R5              PLU RIC REAL LUNAME ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC REAL LUNAME LENGTH ADDRESS
         IC    R5,0(R4)           PLU RIC REAL LUNAME LENGTH
         AR    R4,R5              PLU RIC ALIAS NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC ALIAS NETID LENGTH ADDRESS
         IC    R5,0(R4)           PLU RIC ALIAS NETID LENGTH
         AR    R4,R5              PLU RIC ALIAS LUNAME ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           PLU RIC ALIAS LUNAME ADDRESS
         CLC   0(5,R4),C'ALIAS'   PLU RIC ALIAS NAME EQUAL 'ALIAS'
         BE    ALIASS4C           PLU ALIAS NAME ALLOCATED BY SME - DEALLOCATE
*
** LOCATE SLU RESOURCE INFORMATION CONTROL VECTOR - REAL LUNAME
*
ALIASS4B EQU   *
         L     R4,16(R11)         SLU RIC ADDRESS
         LA    R4,4(R4)           SLU RIC SSCPNAME ADDRESS
         IC    R5,0(R4)           SLU RIC SSCPNAME LENGTH
         AR    R4,R5              SLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC NETID LENGTH ADDRESS
         IC    R5,0(R5)           SLU RIC NETID LENGTH
         AR    R4,R5              SLU RIC REAL NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC REAL LUNAME LENGTH ADDRESS
         IC    R5,0(R4)           SLU RIC REAL LUNAME LENGTH
         AR    R4,R5              SLU RIC ALIAS NETID ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC ALIAS NETID LENGTH ADDRESS
         IC    R5,0(R4)           SLU RIC ALIAS NETID LENGTH
         AR    R4,R5              SLU RIC ALIAS LUNAME ADDRESS VECTOR MINUS ONE
         LA    R4,1(R4)           SLU RIC ALIAS LUNAME ADDRESS
         CLC   0(5,R4),C'ALIAS'   SLU RIC ALIAS NAME EQUAL 'ALIAS'
         BNE   RETURN             SLU RIC ALIAS NAME NOT ALLOCATED BY SME -
*                                   VTAM RETURN
```

```
*
** ALIAS NAME SUFFIX ALLOCATION SUBROUTINE
**
** R1  = ADDRESS OF ALIAS NAME SUFFIX STORAGE
** R2  = ALIAS SUFFIX ENTRY INDEX
** R4  = ADDRESS OF ALIAS NAME NAME - PLU/SLU RIC
** R14 = RETURN ADDRESS - MAIN ROUTINE OF ALIAS FUNCTION
*
ALIASS4C EQU   *
         PACK  0(8,R1),5(3,R4)    PACK ALIAS SUFFIX VALUE
         CVB   R2,0(R1)           CONVERT ALIAS SUFFIX ENTRY TO BINARY INDEX
*                                   VALUE
         LA    R1,8(R2,R1)        CALCULATE ALIAS SUFFIX ENTRY ADDRESS
         MVI   0(R1),X'00'        DEALLOCATE ALIAS SUFFIX ENTRY
         B     RETURN             EXIT TO VTAM
************************************************************************
```

End Function

The end function of the session management exit is invoked during VTAM termination. The session management exit routine uses this function to perform any cleanup required before VTAM completes termination. In this sample, the begin function obtained virtual storage for maintaining user data and loaded a network registration table. Therefore, the end function must delete the table and free the storage before returning control to VTAM. The following code performs these necessary functions:

```
************************************************************************
*        SME ROUTINE - END FUNCTION
************************************************************************
END      EQU   *
*
** DELETE NETID REGISTRATION TABLE
*
         DELETE EP=NETIDTAB       ISSUE DELETE MACRO TO CANCEL PREVIOUS LOAD
*
** FREE ALIAS NAME POOL STORAGE
*
         L     R2,8(R11)          LOCATE USER DATA FIELD IN PARAMETER LIST
         L     R2,0(R2)           OBTAIN ADDRESS OF USER DATA STORAGE
         L     R3,32(R2)          OBTAIN ADDRESS ALIAS NAME POOL STORAGE
         FREEMAIN R,LV=4096,A=(R3)
*
** FREE USER DATA STORAGE AREA
*
         FREEMAIN R,LV=48,A=(R2)
         B     RETURN
************************************************************************
```

Glossary, Bibliography, and Index

Glossary 141

Bibliography 165
VTAM Publications 165
VTAM V3R3 Publications 165
VTAM V3R2 Publications 166
VTAM V3R1.2 Publications 166
VTAM V3R1.1 Publications 166
VTAM V3R1 Publications 166
Related Publications 167
NetView Release 3 Publications 167
NCP Version 4 Publications 167
NCP Version 5 Publications 167
Other Related Publications 168
MVS/XA Publications 168
MVS/ESA Publications 168
VM/SP Publications 168
VSE/Advanced Functions Publications 168
TSO and TSO/VTAM Publications 168

Index 169

Glossary

This glossary defines important NCP, NetView, NetView/PC, SSP, and VTAM abbreviations and terms. It includes information from the *IBM Dictionary of Computing*, SC20-1699. Definitions from the *American National Dictionary for Information Processing* are identified by an asterisk (*). Definitions from draft proposals and working papers under development by the International Standards Organization, Technical Committee 97, Subcommittee 1 are identified by the symbol **(TC97)**. Definitions from the *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the Consultative Committee on International Telegraph and Telephone of the International Telecommunication Union, Geneva, 1980 are preceded by the symbol **(CCITT/ITU)**. Definitions from published sections of the *ISO Vocabulary of Data Processing*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1 and from published sections of the *ISO Vocabulary of Office Machines*, developed by subcommittees of ISO Technical Committee 95, are preceded by the symbol **(ISO)**.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

Reference Words Used in the Entries

The following reference words are used in this glossary:

Deprecated term for. Indicates that the term should not be used. It refers to a preferred term, which is defined.

Synonymous with. Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

Synonym for. Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

Contrast with. Refers to a term that has an opposed or substantively different meaning.

See. Refers to multiple-word terms that have the same last word.

See also. Refers to related terms that have similar (but not synonymous) meanings.

abend. Abnormal end of task.

abnormal end of task (abend). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

ACB. (1) In VTAM, access method control block. (2) In NCP, adapter control block.

ACB name. (1) The name of an ACB macroinstruction. (2) A name specified in the ACBNAME parameter of a VTAM APPL statement. Contrast with *network name*.

accept. (1) For a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macroinstruction, or a VTAM operator issues a command. See also *acquire (1)*. (2) In NPM, an SMP process that moves distributed code to the distribution libraries.

access method. A technique for moving data between main storage and input/output devices.

access method control block (ACB). A control block that links an application program to VSAM or VTAM.

accounting exit routine. In VTAM, an optional installation exit routine that collects statistics about session initiation and termination.

ACF. Advanced Communications Function.

ACF/NCP. Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

ACF/SSP. Advanced Communications Function for the System Support Programs. Synonym for *SSP*.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

acquire. (1) For a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*. (2) To take over resources that were formerly controlled by an access method in another domain, or to resume control of resources that were controlled by this domain but released. Contrast with *release*. See also *resource takeover*.

activate. To make a resource of a node ready to perform the functions for which it was designed. Contrast with *deactivate*.

active. (1) The state a resource is in when it has been activated and is operational. Contrast with *inactive*, *pending*, and *inoperative*. (2) Pertaining to a major or

© Copyright IBM Corp. 1984, 1990

minor node that has been activated by VTAM. Most resources are activated as part of VTAM start processing or as the result of a VARY ACT command.

active application. The application subsystem currently in an extended recovery facility (XRF) session with a terminal user. See *alternate application.*

adapter. Hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device.

adapter control block (ACB). In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

adaptive session pacing. Synonym for *adaptive session-level pacing.*

adaptive session-level pacing. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission to adapt dynamically to variations in availability and demand of buffers on a session by session basis. Session pacing occurs within independent stages along the session path according to local congestion at the intermediate nodes. Synonymous with *adaptive session pacing.* See *pacing, session-level pacing,* and *virtual route pacing.*

adjacent SSCP table. A table containing lists of the system services control points (SSCPs) that VTAM can be in session with or can use to reach destination SSCPs in the same network or in other networks. The table is filed in the VTAM definition library.

adjacent subareas. Two subareas connected by one or more links with no intervening subareas. See also *subarea.*

Advanced Communications Function (ACF). A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

alias name. A name defined in a host used to represent a logical unit name, logon mode table name, or class-of-service name in another network. This name is defined to a name translation program when the alias name does not match the real name. The alias name translation program is used to associate the real and alias names.

alias name translation facility. A function for converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network. Available with NetView or NCCF licensed programs.

allocate. A logical unit (LU) 6.2 application program interface (API) verb used to assign a session to a conversation for the conversation's use. Contrast with *deallocate.*

alternate application. The subsystem that is prepared to take over a particular active application's extended recovery facility (XRF) sessions with terminal users in case the application fails. See *active application.*

application program. (1) A program written for or by a user that applies to the user's work. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

apply. In NPM, an SMP process that moves distributed code to the system libraries.

ASCII. * American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8-bit including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

attaching device. Any device that is physically connected to a network and can communicate over the network.

authorization exit routine. In VTAM, an optional installation exit routine that approves or disapproves requests for session initiation.

automatic logon. (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session will be between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. See also *controlling application program* and *controlling logical unit.* (2) In VM, a process by which a virtual machine is initiated by other than the user of that virtual machine. For example, the primary VM operator's virtual machine is activated automatically during VM initialization.

autotask. An unattended NetView operator station task that does not require a terminal or a logged-on user. Autotasks can run independent of VTAM and are typically used for automated console operations. Contrast with *logged-on operator.*

available. In VTAM, pertaining to a logical unit what is active, connected, enabled, and not at its session limit.

backup session. The session that replaces the failing primary extended recovery facility (XRF) session between a terminal user and the active subsystem.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

BASE disk. The virtual disk that contains the text decks and macroinstructions for VTAM and VM/SNA console support (VSCS). It also contains control files and sample files used when running VTAM on the VM operating system. See *DELTA disk*, *MERGE disk*, *RUN disk*, and *ZAP disk*.

bidder. In SNA, the LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session to begin a bracket. Contrast with *first speaker*. See also *bracket protocol* and *contention*.

Binary Synchronous Communication (BSC). (1) Communication using binary synchronous line discipline. (2) A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

bind. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

BIU segment. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either a request/response header (RH) followed by all or a portion of a request/response unit (RU), or only a portion of an RU.

blocking of PIUs. In SNA, an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

boundary function. (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *boundary node*, *network addressable unit (NAU)*, *peripheral path control*, *subarea node*, and *subarea path control*.

boundary node. (1) A subarea node with boundary function. See *subarea node*. See also *boundary function*. (2) The programming component that performs FID2 (format identification type 2) conversion, channel data link control, pacing, and channel or device error recovery procedures for a locally attached station. These functions are similar to those performed by a network control program for an NCP-attached station.

bracket protocol. In SNA, a data flow control protocol in which exchanges between the two LU-LU half-sessions are achieved through the use of brackets, with one LU designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder* and *first speaker*.

BSC. Binary Synchronous Communication.

buffer. A portion of storage for temporarily holding input or output data.

bypass. To eliminate an attaching device or an access unit from a ring network by allowing the data to flow in a path around it.

call. (1) * (ISO) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) To transfer control to a procedure, program, routine, or subroutine. (3) The actions necessary to make a connection between two stations. (4) To attempt to contact a user, regardless of whether the attempt is successful.

call connected packet. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to indicate to a calling data terminal equipment (DTE) that the connection for the call has been completely established.

call request packet. A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network.

call-accepted packet. * (ISO) A call supervision packet that a called data terminal equipment (DTE) transmits to indicate to the data circuit-terminating equipment (DCE) that it accepts the incoming call.

calling. * (ISO) The process of transmitting selection signals in order to establish a connection between data stations.

CALLOUT. The logical channel type on which the data terminal equipment (DTE) can send a call, but cannot receive one.

CDRM. Cross-domain resource manager.

CDRSC. Cross-domain resource.

chain. (1) A group of logically linked records. (2) See *RU chain*.

channel. * A path along which signals can be sent, for example, data channel, output channel. See *data channel* and *input/output channel*. See also *link*.

channel-attached. (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

character-coded. Synonym for *unformatted.*

CICS. Customer Information Control System.

CID. Communication identifier.

circuit switching. (1) * (ISO) A process that, on demand, connects two or more data terminal equipments (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (2) Synonymous with *line switching.* (3) See also *message switching* and *packet switching.*

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

cleanup. A network services request, sent by a system services control unit (SSCP) to a logical unit (LU), that causes a particular LU-LU session with that LU to be ended immediately and without the participation of either the other LU or its SSCP.

clear indication packet. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to inform a data terminal equipment (DTE) that a call has been cleared.

clear request packet. A call supervision packet transmitted by a data terminal equipment (DTE) to ask that a call be cleared.

CNM. Communication network management.

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in a FID4 transmission header. See also *VTAM operator command.*

command processor. A user-written module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are invoked as commands.

communication adapter. An optional hardware feature, available on certain processors, that permits communication lines to be attached to the processors.

communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit; for example, the IBM 3725 Communication Controller. It manages the details of line control and the routing of data through a network.

communication identifier (CID). In VTAM, a key for locating the control blocks that represent a session. The key is created during the session-establishment procedure and deleted when the session ends.

communication line. Deprecated term for *telecommunication line* and *transmission line.*

communication management configuration host node. The type 5 host processor in a communication management configuration that does all network-control functions in the network except for the control of devices channel-attached to data hosts. Synonymous with *communication management host.* Contrast with *data host node.*

communication management host. Synonym for *communication management configuration host node.* Contrast with *data host.*

communication network management (CNM). The process of designing, installing, operating, and managing the distribution of information and controls among end users of communication systems.

communication network management (CNM) application program. A VTAM application program that issues and receives formatted management services request units for physical units. For example, the NetView program.

communication network management (CNM) interface. The interface that the access method provides to an application program for handling data and commands associated with communication system management. CNM data and commands are handled across this interface.

communication network management (CNM) processor. A program that manages one of the functions of a communications system. A CNM processor is executed under control of the NetView program.

component. Any part of a network other than an attaching device, such as an access unit.

composite end node (CEN). A group of nodes made up of a single type 5 node and its subordinate type 4 nodes that together support type 2.1 protocols. To a type 2.1 node, a CEN appears as one end node. For example, NCP and VTAM act as a composite end node.

configuration. (1) (TC97) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

configuration services. In SNA, one of the types of network services in the control point (CP) and in the physical unit (PU); configuration services activate, deactivate, and maintain the status of physical units, links, and link stations. Configuration services also shut down and restart network elements and modify path control routing tables and address-translation tables. See also *maintenance services*, *management services*, *network services*, and *session services*.

connection. Synonym for *physical connection*.

contention. A situation in which two logical units (LUs) that are connected by an LU 6.2 session both attempt to allocate the session for a conversation at the same time. The control operator assigns "winner" and "loser" status to the LUs so that processing may continue on an orderly basis. The contention loser requests permission from the contention winner to allocate a conversation on the session, and the contention winner either grants or rejects the request. See also *bidder*.

control block. (1) (ISO) A storage area used by a computer program to hold control information. (2) In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the Adapter Support Interface to request an operation.

control point (CP). (1) A system services control point (SSCP) that provides hierarchical control of a group of nodes in a network. (2) A control point (CP) local to a specific node that provides control of that node, either in the absence of SSCP control (for type 2.1 nodes engaged in peer to peer communication) or to supplement SSCP control.

control program (CP). The VM operating system that manages the real processor's resources and is responsible for simulating System/370s for individual users.

controller. A unit that controls input/output operations for one or more devices.

controlling application program. In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit*.

controlling logical unit. In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program*.

conversation. In SNA, a logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

converted command. An intermediate form of a character-coded command produced by VTAM through use of an unformatted system services definition table. The format of a converted command is fixed; the unformatted system services definition table must be constructed in such a manner that the character-coded command (as entered by a logical unit) is converted into the predefined, converted command format. See also *unformatted*.

COS. Class of service.

CP. (1) Control program. (2) Control point.

cross-domain. In SNA, pertaining to control of resources involving more than one domain.

cross-domain resource (CDRSC). A resource owned by a cross-domain resource manager (CDRM) in another domain but known by the CDRM in this domain by network name and associated CDRM.

cross-domain resource manager (CDRM). In VTAM, the function in the system services control point (SSCP) that controls initiation and termination of cross-domain sessions.

cross-network. In SNA, pertaining to control or resources involving more than one SNA network.

cross-network session. An LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

Customer Information Control System (CICS). A licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It also includes facilities for building, using, and maintaining data bases.

data channel. Synonym for *input/output channel*. See *channel*.

data host. Synonym for *data host node*. Contrast with *communication management configuration host*.

data host node. In a communication management configuration, a type 5 host node that is dedicated to processing applications and does not control network resources, except for its channel-attached or communication adapter-attached devices. Synonymous with *data host*. Contrast with *communication management configuration host node*.

data link. In SNA, synonym for *link*.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a transmission medium connecting two nodes and perform error control for the link connection. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

data link level. In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*.

data packet. At the interface between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE), a packet used to transmit user data over a virtual circuit.

DCE clear confirmation packet. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to confirm that a call has been cleared.

deactivate. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

deallocate. A logical unit (LU) 6.2 application program interface (API) verb that terminates a conversation, thereby freeing the session for a future conversation. Contrast with *allocate*.

default SSCP list. A list of system services control points (SSCPs), either in VTAM's network or another network, that can be used when no predefined cross-domain resource (CDRSC) or name translation function is provided specifying an LU's owning cross-domain resource manager (CDRM). This list is filed as a part of an adjacent SSCP table in the VTAM definition library.

definite response (DR). In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response* and *no response*.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See Figure 19, Figure 20, and Figure 21. See also *macroinstruction*.

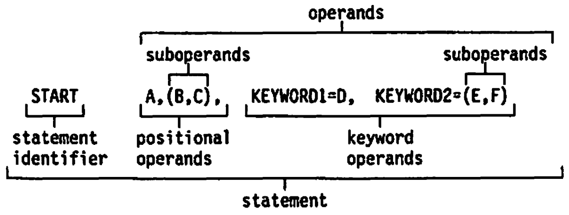


Figure 19. Example of a Language Statement

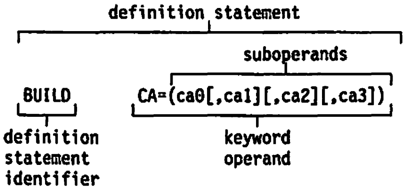


Figure 20. NCP Examples

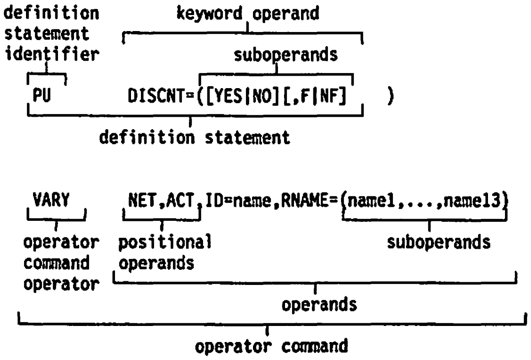


Figure 21. VTAM Examples

DELTA disk. The virtual disk in a VM operating system that contains program temporary fixes (PTFs) that have been installed but not merged. See *BASE disk*, *MERGE disk*, *RUN disk*, and *ZAP disk*.

destination logical unit (DLU). The logical unit to which data is to be sent. Contrast with *origin logical unit (OLU)*.

device. An input/output unit such as a terminal, display, or printer. See *attaching device*.

directory. In VM, a control program (CP) disk that defines each virtual machine's normal configuration.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

disabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited*.

discarded packet. A packet that is intentionally destroyed.

disconnection. The termination of a physical connection.

discontiguous shared segment. An area of virtual storage outside the address range of a virtual machine. It can contain read-only data or reentrant code. It connects discontiguous segments to a virtual machine's address space so programs can be fetched.

display. (1) To present information for viewing, usually on a terminal screen or a hard-copy device. (2) A device or medium on which information is presented, such as a terminal screen. (3) Deprecated term for *panel*.

DLC. Data link control.

DLU. Destination logical unit.

domain. (1) An access method, its application programs, communication controllers, connecting lines, modems, and attached terminals. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See *system services control point domain* and *type 2.1 node control point domain*. See also *single-domain network* and *multiple-domain network*.

domain operator. In a multiple-domain network, the person or program that controls the operation of the resources controlled by one system services control point. Contrast with *network operator* (2).

downstream. In the direction of data flow from the host to the end user. Contrast with *upstream*.

Downstream Load Utility (DSLU). A licensed program that uses the communication network management (CNM) interface to support the load requirements of certain type 2 physical units, such as the IBM 3644 Automatic Data Unit and the IBM 8775 Display Terminal.

drop. In the IBM Token-Ring Network, a cable that leads from a faceplate to the distribution panel in a wiring closet. When the IBM Cabling System is used with the IBM Token-Ring Network, a drop may form part of a lobe.

dump. (1) Computer printout of storage. (2) To write the contents of all or part of storage to an external medium as a safeguard against errors or in connection with debugging. (3) (ISO) Data that have been dumped.

EBCDIC. * Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

element. (1) A field in the network address. (2) The particular resource within a subarea identified by the element address. See also *subarea*.

element address. In SNA, a value in the element address field of the network address identifying a specific resource within a subarea. See *subarea address*.

Emulation Program (EP). An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program*.

enabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is now ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *disabled* and *inhibited*.

end node. A type 2.1 node that does not provide any intermediate routing or session services to any other node. For example, APPC/PC is an end node. See *composite end node*, *node*, and *type 2.1 node*.

end user. In SNA, the ultimate source or destination of application data flowing through an SNA network. An end user may be an application program or a terminal operator.

entry point. An SNA node that provides distributed network management support. It may be a type 2, type 2.1, type 4, or type 5 node. It sends SNA-formatted network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal point initiated commands to manage and control its resources.

EP. Emulation Program.

ER. (1) Explicit route. (2) Exception response.

ESTAE. Extended specify task abnormal exit.

exception response (ER). In SNA, a value in the form-of-response-requested field of a request header (RH). An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response* and *no response*. See also *negative response*.

EXEC. In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

exit routine. Any of several types of special-purpose user-written routines. See *accounting exit routine, authorization exit routine, logon-interpret routine, virtual route selection exit routine, EXLST exit routine*, and *RPL exit routine*.

EXLST exit routine. In VTAM, a routine whose address has been placed in an exit list (EXLST) control block. The addresses are placed there with the EXLST macroinstruction, and the routines are named according to their corresponding operand; hence DFASY exit routine, TPEND exit routine, RELREQ exit routine, and so forth. All exit list routines are coded by the VTAM application programmer. Contrast with *RPL exit routine*.

explicit route (ER). In SNA, the path control network elements, including a specific set of one or more transmission groups, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*. See also *path* and *route extension*.

explicit route length. In SNA, the number of transmission groups in an explicit route.

extended architecture (XA). An extension to System/370 architecture that takes advantage of continuing high performance enhancements to computer system hardware.

extended recovery facility (XRF). Software designed to minimize the effect of failures in MVS, VTAM, the host processor, or IMS/VS on sessions between IMS/VS and designated terminals. It provides an alternate subsystem to take over failing sessions.

extended specify task abnormal exit (ESTAE). An MVS macroinstruction that provides recovery capability and gives control to the user-specified exit routine for processing, diagnosing an abend, or specifying a retry address.

feature. A particular part of an IBM product that a customer can order separately.

field-formatted. Pertaining to a request or response that is encoded into fields, each having a specified format such as binary codes, bit-significant flags, and symbolic names. Contrast with *character-coded*.

first speaker. In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both half-sessions attempt to begin a bracket simultaneously. Contrast with *bidder*. See also *bracket protocol*.

flow control. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units, with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *adaptive session-level pacing, pacing, session-level pacing*, and *virtual route pacing*.

formatted system services. A portion of VTAM that provides certain system services as a result of receiving a field-formatted command, such as an Initiate or Terminate command. Contrast with *unformatted system services (USS)*. See also *field-formatted*.

frame. (1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

gateway. (1) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP. (2) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols.

gateway NCP. An NCP that performs address translation to allow cross-network session traffic. The gateway NCP connects two or more independent SNA networks. Synonymous with *gateway node*.

gateway node. Synonym for *gateway NCP*.

gateway SSCP. An SSCP that is capable of cross-network session initiation, termination, takedown, and session outage notification. A gateway SSCP is in session with the gateway NCP; it provides network name translation and assists the gateway NCP in

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

setting up alias network addresses for cross-network sessions.

GCS. Group control system.

generation. The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

generic unbind. Synonym for *session deactivation request*.

group. In the NetView/PC program, to identify a set of application programs that are to run concurrently.

group control system (GCS). A component of VM that provides multiprogramming and shared memory support to virtual machines. It is a saved system intended for use with SNA products.

half-session. In SNA, a component that provides function management data (FMD) services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also *primary half-session* and *secondary half-session*.

help panel. An online display that tells you how to use a command or another aspect of a product. See *task panel*.

hierarchy. In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

host node. A node providing an application program interface (API) and a common application interface. See *boundary node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

ILU. Initiating logical unit.

IMS. Information Management System/Virtual Storage. Synonymous with *IMS/VS*.

IMS/VS. Information Management System/Virtual Storage. Synonym for *IMS*.

inactive. Describes the state of a resource that has not been activated or for which the VARY INACT command has been issued. Contrast with *active*. See also *inoperative*.

incoming call packet. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a called data terminal equipment (DTE) that another DTE has requested a call.

information (I) format. A format used for information transfer.

Information Management System (IMS). A general purpose system whose full name is Information Management System/Virtual Storage (IMS/VS). It enhances the capabilities of OS/VS for batch processing and telecommunication and allows users to access a computer-maintained data base through remote terminals.

Information/Management. A feature of the Information/System licensed program that provides interactive systems management applications for problem, change, and configuration management.

inhibited. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *enabled* and *disabled*.

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

initiate. A network services request sent from a logical unit (LU) to a system services control point (SSCP) requesting that an LU-LU session be established.

inoperative. The condition of a resource that has been active, but is not. The resource may have failed, received an INOP request, or is suspended while a reactivate command is being processed. See also *inactive*.

input/output channel. (1) (ISO) In a data processing system, a functional unit that handles the transfer of data between internal and peripheral equipment. (2) In a computing system, a functional unit, controlled by a processor, that handles the transfer of data between processor storage and local peripheral devices. Synonymous with *data channel*. See *channel*. See also *link*.

interconnection. See *SNA network interconnection*.

interface. * A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

intermediate routing node (IRN). In SNA, a subarea node with intermediate routing function.

intermediate SSCP. An SSCP along a session initiation path that owns neither of the LUs involved in a cross-network LU-LU session.

interpret table. In VTAM, an installation-defined correlation list that translates an argument into a string of eight characters. Interpret tables can be used to translate logon data into the name of an application program for which the logon is intended.

IPL. (1) * Initial program loader. (2) Initial program load.

IRN. Intermediate routing node.

item. In CCP, any of the components, such as communication controllers, lines, cluster controllers, and terminals, that comprise an IBM 3710 Network Controller configuration.

keyword. (1) (TC97) A lexical unit that, in certain contexts, characterizes some language construction. (2) * One of the predefined words of an artificial language. (3) One of the significant and informative words in a title or document that describes the content of that document. (4) A name or symbol that identifies a parameter. (5) A part of a command operand that consists of a specific character string (such as DSNAME=). See also *definition statement* and *keyword operand*. Contrast with *positional operand*.

keyword operand. An operand that consists of a keyword followed by one or more values (such as DSNAME=HELLO). See also *definition statement*. Contrast with *positional operand*.

keyword parameter. A parameter that consists of a keyword followed by one or more values.

LAN. An industry-wide acronym for local area network.

line. See *communication line*.

line control discipline. Synonym for *link protocol*.

line discipline. Synonym for *link protocol*.

line switching. Synonym for *circuit switching*.

link. In SNA, the combination of the link connection and the link stations joining network nodes; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with *data link*. See Figure 22 on page 151.

link connection segment. A portion of the configuration that is located between two resources listed consecutively in the service point command service (SPCS) query link configuration request list.

link level. (1) A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the duplex link connecting the subscriber's machine to the network node. LAP and LAPB are the link access protocols recommended by the CCITT. (2) See *data link level*.

link protocol. (1) See *protocol*. (2) See also *link level*.

link station. (1) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. (2) In VTAM, a named resource within a subarea node that represents another subarea node that is attached by a subarea link. In the resource hierarchy, the link station is subordinate to the subarea link.

link-attached. Pertaining to devices that are physically connected by a telecommunication line. Contrast with *channel-attached*. Synonymous with *remote*.

load module. (ISO) A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor.

local. Pertaining to a device that is attached to a controlling unit by cables, rather than by a telecommunication line. Synonymous with *channel-attached*.

local address. In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

local area network (LAN). (1) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately sized geographic area such as a single office building, warehouse, or campus, and which do not generally extend across public rights-of-way. Contrast with *wide area network*.

local non-SNA major node. In VTAM, a major node whose minor nodes are channel-attached non-SNA terminals.

local SNA major node. In VTAM, a major node whose minor nodes are channel-attached peripheral nodes.

logged-on operator. A NetView operator station task that requires a terminal and a logged-on user. Contrast with *autotask*.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other LUs. See also *network addressable unit (NAU)*, *peripheral LU*, *phys-*

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

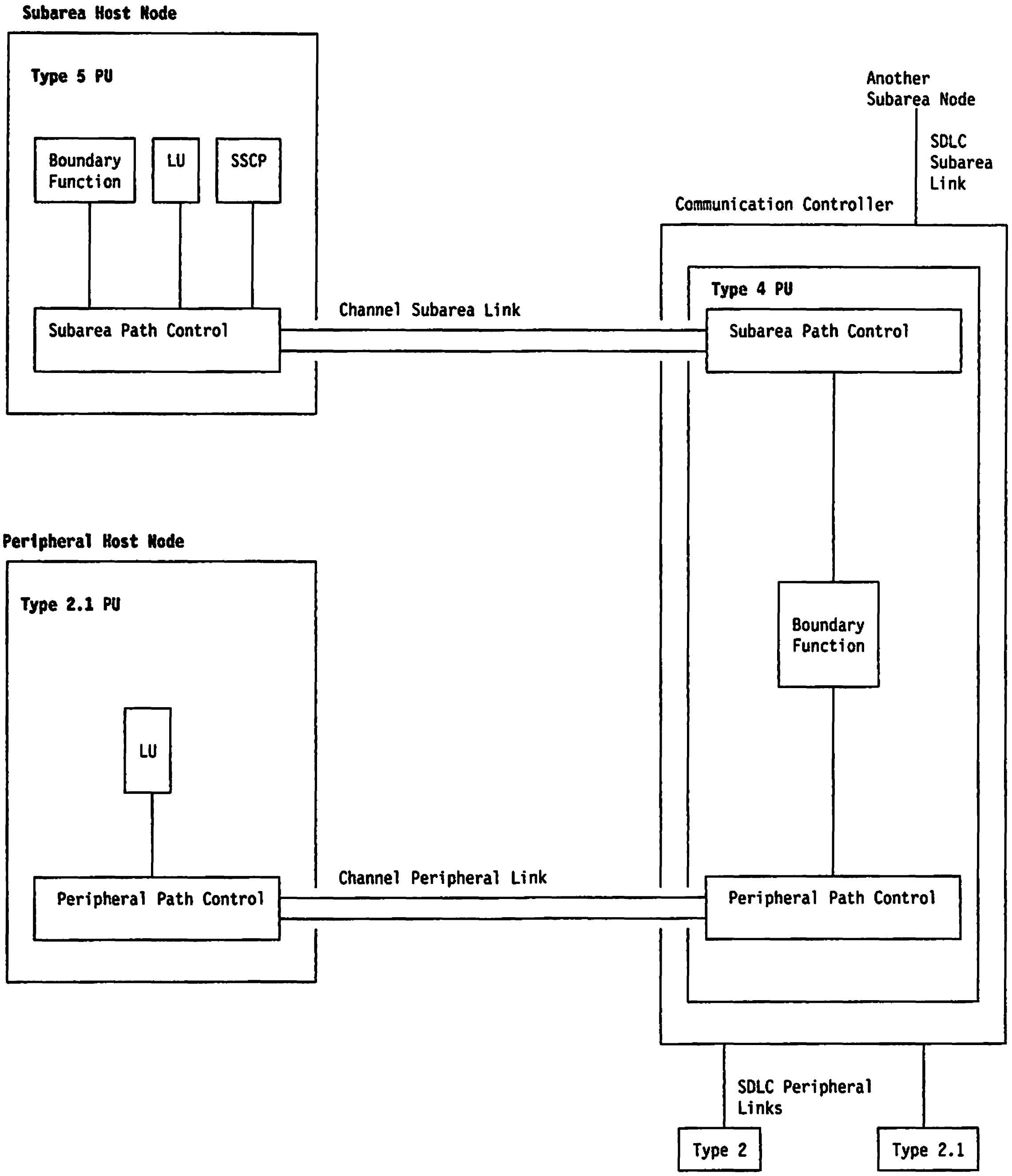


Figure 22. Links and Path Controls

ical unit (PU), *system services control point (SSCP)*, *primary logical unit (PLU)*, and *secondary logical unit (SLU)*.

logical unit (LU) services. In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also *physical unit (PU) services*.

logical unit (LU) 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (1) a peer relationship between session partners, (2) efficient utilization of a session for multiple transactions, (3) comprehensive end-to-end error processing, and (4) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

logmode table. Synonym for *logon mode table*.

logon. In VTAM, an unformatted session initiation request for a session between two logical units. See *automatic logon* and *simulated logon*. See also *session-initiation request*.

logon data. In VTAM: (1) The user data portion of a field-formatted or unformatted session-initiation request. (2) The entire logon sequence or message from a logical unit (LU). Synonymous with *logon message*.

logon message. Synonym for *logon data*.

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

logon mode table. In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name. Synonymous with *logmode table*.

logon-interpret routine. In VTAM, an installation exit routine, associated with an interpret table entry, that translates logon information. It may also verify the logon.

LU. Logical unit.

LU type. In SNA, the classification of an LU-LU session in terms of the specific subset of SNA protocols and options supported by the logical units (LUs) for that session, namely:

> The mandatory and optional values allowed in the session activation request.
>
> The usage of data stream controls, function management headers (FMHs), request unit (RU) parameters, and sense codes.
>
> Presentation services protocols such as those associated with FMH usage.

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU 6.2. Logical unit 6.2.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU-LU session type. A deprecated term for *LU type*.

macroinstruction. (1) An instruction that when executed causes the execution of a predefined sequence of instructions in the same source language. (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. See also *definition statement*.

maintain system history program (MSHP). A program that facilitates the process of installing and servicing a VSE system.

maintenance services. In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). Maintenance services provide facilities for testing links and nodes and for collecting and recording error information. See also *configuration services, management services, network services*, and *session services*.

major node. In VTAM, a set of resources that can be activated and deactivated as a group. See *node* and *minor node*.

management services. In SNA, one of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management and change management. See also *configuration services, maintenance services, network services*, and *session services*.

maximum SSCP rerouting count. The maximum number of times a session initiation request will be rerouted to intermediate system services control points (SSCPs) before the request reaches the destination SSCP. This count is used to prevent endless rerouting of session initiation requests.

MDR. Miscellaneous data record.

MERGE disk. The virtual disk in the VM operating system that contains program temporary fixes (PTFs) after the VMFMERGE EXEC is invoked. See *BASE disk, DELTA disk, RUN disk*, and *ZAP disk*.

message. (1) (TC97) A group of characters and control bit sequences transferred as an entity. (2) In VTAM, the amount of function management data (FMD) transferred to VTAM by the application program with one SEND request.

message switching. (1) * (ISO) In a data network, the process of routing messages by receiving, storing, and forwarding complete messages. (2) The technique of receiving a complete message, storing, and then forwarding it unaltered to its destination.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

minor node. In VTAM, a uniquely-defined resource within a major node. See *node* and *major node*.

miscellaneous data record (MDR). A record of a network hardware error recorded by the NCP and sent to the VTAM host that owns the failing component. Then VTAM writes the error on the operating system error data set.

mode name. A symbolic name for a set of session charácteristics. For LU 6.2, a mode name and a partner LU name together define a group of parallel sessions having the same characteristics.

module. * A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

monitor. In the IBM Token-Ring Network, the function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

MSHP. Maintain system history program.

Multiple Virtual Storage (MVS). An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

Multiple Virtual Storage for Extended Architecture (MVS/XA). An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for Extended Architecture. Extended architecture allows 31-bit storage addressing. MVS/XA is a software operating system controlling the execution of programs.

multiple-domain network. In SNA, a network with more than one system services control point (SSCP). Contrast with *single-domain network*.

MVS. Multiple Virtual Storage.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/XA. Multiple Virtual Storage for Extended Architecture.

name translation. In SNA network interconnection, converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network. This function can be provided through the NetView program and invoked by a gateway system services control point (SSCP) when necessary. See also *alias name*.

NAU. Network addressable unit.

NCP. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

negative response (NR). In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*. See *exception response*.

NetView. A system 370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

NetView-NetView task (NNT). The task under which a cross-domain NetView operator session runs. See *operator station task*.

network. (1) (TC97) An interconnected group of nodes. (2) In data processing, a user application network. See *path control network, public network, SNA network*, and *user-application network*.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address*. See also *network name*.

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name, network address*, and *path control network*.

network control program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network identifier (network ID). The network name defined to NCPs and hosts to indicate the name of the network in which they reside. It is unique across all communicating SNA networks.

Network Logical Data Manager (NLDM). An IBM licensed program that collects and correlates session-related data and provides online access to this information. It runs as an NCCF communication network management (CNM) application program. Its function is included and enhanced in NetView's session monitor.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network name. (1) In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link, or a link station. See also *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program is its network name and it must be unique across domains. Contrast with *ACB name*. See *uninterpreted name*.

network operator. (1) A person or program responsible for controlling the operation of all or part of a network. (2) The person or program that controls all the domains in a multiple-domain network. Contrast with *domain operator*.

Network Problem Determination Application (NPDA). An IBM licensed program that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. It runs as an NCCF communication network management (CNM) application program. Its function is included and enhanced in NetView's hardware monitor.

network services (NS). In SNA, the services within network addressable units (NAUs) that control network operation through SSCP-SSCP, SSCP-PU, and SSCP-LU sessions. See *configuration services, maintenance services, management services*, and *session services*.

network services (NS) header. In SNA, a 3-byte field in a function management data (FMD) request/response unit (RU) flowing in an SSCP-LU, SSCP-PU, or SSCP-SSCP session. The network services header is used primarily to identify the network services category of the request unit (RU) (for example, configuration services, session services) and the particular request code within a category.

Network Terminal Option (NTO). An IBM licensed program used in conjunction with NCP that allows certain non-SNA devices to participate in sessions with SNA application programs in the host processor. NTO converts non-SNA protocol to SNA protocol when data is sent to the host from a non-SNA device and reconverts SNA protocol to non-SNA protocol when data is sent back to the device.

NLDM. Network Logical Data Manager.

NMVT. Network management vector transport.

no response. In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

node. (1) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. See *boundary node, host node, peripheral node*, and *subarea node*. (2) In VTAM, a point in a network defined by a symbolic name. See *major node* and *minor node*.

node name. In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes. See also *type 2.1 node*.

notify. A network services request that is sent by an SSCP to a logical unit (LU) to inform the LU of the status of a procedure requested by the LU.

NPDA. Network Problem Determination Application.

NTO. Network Terminal Option.

OLU. Origin logical unit.

online. Stored in a computer and accessible from a terminal.

open. (1) In the IBM Token-Ring Network, to make an adapter ready for use. (2) A break in an electrical circuit.

operand. (1) (ISO) An entity on which an operation is performed. (2) * That which is operated upon. An operand is usually identified by an address part of an instruction. (3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

command processor. (4) An expression to whose value an operator is applied. See also *definition statement*, *keyword*, *keyword parameter*, and *parameter*.

operator. (1) In a language statement, the lexical entity that indicates the action to be performed on operands. See also *definition statement*. (2) A person who operates a machine. See *network operator*. (3) A person or program responsible for managing activities controlled by a given piece of software such as MVS, the NetView program, or IMS. See *logged-on operator* and *network operator* See also *autotask* and *operator station task*.

operator station task (OST). The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

origin logical unit (OLU). The logical unit from which data is sent. Contrast with *destination logical unit (DLU)*.

pacing. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See *session-level pacing*, *send pacing*, and *virtual route (VR) pacing*. See also *flow control*.

pacing group. In SNA, (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. Synonymous with *window*. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests.

pacing group size. In SNA, (1) The number of path information units (PIUs) in a virtual route pacing group. The pacing group size varies according to traffic congestion along the virtual route. Synonymous with *window size*. (2) The number of requests in a session-level pacing group.

pacing response. In SNA, an indicator that signifies a receiving component's readiness to accept another pacing group; the indicator is carried in a response header (RH) for session-level pacing, and in a transmission header (TH) for virtual route pacing.

packet. (ISO) A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and possibly error control information are arranged in a specific format. See *call-accepted packet*, *call-connected packet*, *call request packet*, *call supervision packets*, *clear indication packet*, *clear request packet*, *data packet*, *DCE clear confirmation packet*, *discarded packet*, *incoming call packet*, *permit packet*, and *reset packet*.

packet level. (1) The packet format and control procedures for exchange of packets containing control information and user data between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). See also *data link level* and *physical level*. (2) A part of Recommendation X.25 that defines the protocol for establishing logical connections between two DTEs and for transferring data on these connections.

packet major node. In VTAM, a set of minor nodes representing resources, such as switched virtual circuits and permanent virtual circuits, attached through an X.25 port. See *major node*.

packet mode operation. Synonym for *packet switching*.

packet switching. (1) (ISO) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet. On completion of the transmission, the channel is made available for the transfer of other packets. (2) Synonymous with *packet mode operation*. See also *circuit switching*.

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also *scroll*. (3) (ISO) In a virtual storage system, a fixed-length block that has a virtual address and that can be transferred between real storage and auxiliary storage. (4) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. See *help panel* and *task panel*. Contrast with *screen*. (2) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

parallel sessions. In SNA, two or more concurrently active sessions between the same two logical units (LUs) using different pairs of network addresses. Each session can have independent session parameters.

parameter. (1) (ISO) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. See also *keyword*, *keyword parameter*, and *operand*.

path. (1) In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*. (2) In VTAM when defining a switched major node, a potential dial-out port that can be used to reach that node. (3) In the NetView/PC program, a complete line in a configuration that contains all of the resources in the service point command service (SPCS) query link configuration request list.

path control (PC). The function that routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units (BTUs) and one or more PIUs with data link control. Path control differs for peripheral nodes, which use local addresses for routing, and subarea nodes, which use network addresses for routing. See *peripheral path control* and *subarea path control*. See also *link*, *peripheral node*, and *subarea node*.

path control (PC) network. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network* and *user application network*. See also *boundary function*.

path information unit (PIU). In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header*.

PCID. Procedure-correlation identifier.

peripheral host node. A node that provides an application program interface (API) for running application programs but does not provide SSCP functions and is not aware of the network configuration. The peripheral host node does not provide subarea node services. It has boundary function provided by its adjacent subarea. See *boundary node*, *host node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral LU.. In SNA, a logical unit representing a peripheral node.

peripheral node. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node is a physical unit (PU) type 1, 2.0, or 2.1 node connected to a subarea node with boundary function within a subarea. See *boundary node*, *host node*, *node*, *peripheral host node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral path control. The function in a peripheral node that routes message units between units with local addresses and provides the paths between them. See *path control* and *subarea path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

peripheral PU. In SNA, a physical unit representing a peripheral node.

permit packet. At the interface between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE), a packet used to transmit permits over a virtual circuit.

physical connection. In VTAM, a point-to-point connection or multipoint connection. Synonymous with *connection*.

physical level. The mechanical, electrical, functional, and procedural media used to activate, maintain, and deactivate the physical link between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE). See also *data link level* and *packet level*.

physical unit (PU). In SNA, a type of network addressable unit (NAU). A physical unit (PU) manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. See also *peripheral PU* and *subarea PU*.

physical unit (PU) services. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also *logical unit (LU) services*.

PIU. Path information unit.

PLU. Primary logical unit.

polling. (1) * Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (2) (TC97) The process whereby stations are invited, one at a time, to transmit.

positional operand. An operand in a language statement that has a fixed position. See also *definition statement*. Contrast with *keyword operand*.

positive response. A response indicating that a request was received and processed. Contrast with *negative response*.

primary half-session. In SNA, the half-session that sends the session activation request. See also *primary logical unit*. Contrast with *secondary half-session*.

primary logical unit (PLU). In SNA, the logical unit (LU) that contains the primary half-session for a particular

LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU)*.

primary session. An extended recovery facility (XRF) session between the active application subsystem and a terminal user.

problem determination. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environment failure such as a power loss, or a user error.

procedure-correlation identifier (PCID). In SNA, a value used by a control point to correlate requests and replies.

protection key. An indicator that appears in the current program status word whenever an associated task has control of the system. This indicator must match the storage keys of all main storage locks that the task is to use.

protocol. (1) (CCITT/ITU) A specification for the format and relative timing of information exchanged between communicating parties. (2) (TC97) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See also *bracket protocol* and *link protocol*.

PU. Physical unit.

PU-PU flow. In SNA, the exchange between physical units (PUs) of network control requests and responses.

public network. A network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet switched, and leased-circuit services to the public. Contrast with *user-application network*.

real name. The name by which a logical unit (LU), logon mode table, or class-of-service (COS) table is known within the SNA network in which it resides.

real network address. The address by which a logical unit (LU) is known within the SNA network in which it resides.

receive pacing. In SNA, the pacing of message units that the component is receiving. See also *send pacing*.

RECFMS. Record formatted maintenance statistics.

RECMS. Record maintenance statistics.

Recommendation X.21 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for a general purpose interface between data terminal equipment and data circuit equipment for synchronous operations on a public data network.

Recommendation X.25 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching*.

record. (1) (ISO) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (2) (TC97) A set of data treated as a unit. (3) A set of one or more related data items grouped for processing. (4) In VTAM, the unit of data transmission for record mode. A record represents whatever amount of data the transmitting node chooses to send.

record formatted maintenance statistics (RECFMS). A statistical record built by an SNA controller and usually solicited by the host.

record maintenance statistics (RECMS). An SNA error event record built from an NCP or line error and sent unsolicited to the host.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks. For example, the 3710 Network Controller routines may be reentrant.

release. For VTAM, to relinquish control of resources (communication controllers or physical units). See also *resource takeover*. Contrast with *acquire (2)*.

remote. Concerning the peripheral parts of a network not centrally linked to the host processor and generally using telecommunication lines with public right-of-way.

request header (RH). In SNA, control information preceding a request unit (RU). See also *request/response header (RH)*.

request parameter list (RPL). In VTAM, a control block that contains the parameters necessary for processing a request for data transfer, for establishing or terminating a session, or for some other operation.

request unit (RU). In SNA, a message unit that contains control information, end-user data, or both.

request/response header (RH). In SNA, control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

request/response unit (RU). In SNA, a generic term for a request unit or a response unit. See also *request unit (RU)* and *response unit.*

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

reset packet. A packet used to reset a virtual circuit at the interface between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE).

resource. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network.

resource takeover. In VTAM, action initiated by a network operator to transfer control of resources from one domain to another. See also *acquire (2)* and *release*. See *takeover*.

response. A reply represented in the control field of a response frame. It advises the primary or combined station of the action taken by the secondary or other combined station to one or more commands. See also *command.*

response header (RH). In SNA, a header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response, pacing response*, and *positive response*.

response time. (1) The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal. (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

response unit (RU). In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

return code. * A code [returned from a program] used to influence the execution of succeeding instructions.

RH. Request/response header.

ring. A network configuration where a series of attaching devices are connected by unidirectional transmission links to form a closed path.

route. See *explicit route* and *virtual route*.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path, explicit route (ER)* and *virtual route (VR)*.

routing. The assignment of the path by which a message will reach its destination.

RPL. Request parameter list.

RPL exit routine. In VTAM, an application program exit routine whose address has been placed in the EXIT field of a request parameter list (RPL). VTAM invokes the routine to indicate that an asynchronous request has been completed. See *EXLST exit routine*.

RU. Request/response unit.

RU chain. In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated by means of control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

RUN disk. The virtual disk that contains the VTAM and VM/SNA console support (VSCS) load libraries, program temporary fixes (PTFs) and user-written modifications from the ZAP disk. See *BASE disk*, *DELTA disk*, *MERGE disk*, and *ZAP disk*.

same-domain. Refers to communication between entities in the same SNA domain. Contrast with *cross-domain*. See also *single-domain network*.

SAW data. Synonym for *session awareness (SAW) data*.

SC. Session control.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

screen. An illuminated display surface; for example, the display surface of a CRT or plasma panel. Contrast with *panel*.

scroll. To move all or part of the display image vertically to display data that cannot be observed within a single display image. See also *page (2)*.

SDLC. Synchronous Data Link Control.

secondary half-session. In SNA, the half-session that receives the session-activation request. See also *secondary logical unit (SLU)*. Contrast with *primary half-session*.

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU)*.

secondary logical unit (SLU) key. A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

segment. (1) In the IBM Token-Ring Network, a section of cable between components or devices on the network. A segment may consist of a single patch cable, multiple patch cables connected together, or a combination of building cable and patch cables connected together. (2) See *link connection segment*.

send pacing. In SNA, pacing of message units that a component is sending. See also *receive pacing*.

service point (SP). An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

service reminder (SR). In the NetView/PC program, a notification set by the operator that is displayed on a panel and logs a specified message.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session. See *half-session*, *LU-LU session*, *SSCP-LU session*, *SSCP-PU session*, and *SSCP-SSCP session*. See also *LU-LU session type* and *PU-PU flow*.

session activation request. In SNA, a request that activates a session between two network addressable units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Contrast with *session deactivation request*.

session awareness (SAW) data. Data collected by the NetView program about a session that includes the session type, the names of session partners, and information about the session activation status. It is collected for LU-LU, SSCP-LU, SSCP-PU, and SSCP-SSCP sessions and for non-SNA terminals not supported by NTO. It can be displayed in various forms, such as most recent sessions lists.

session control (SC). In SNA, (1) One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification. (2) A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

session deactivation request. In SNA, a request that deactivates a session between two network addressable units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

session management exit routine. An installation-supplied VTAM exit routine that performs authorization, accounting, and gateway path selection functions.

session monitor. The component of the NetView program that collects and correlates session-related data and provides online access to this information.

session parameters. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units. See also *logon mode*.

session partner. In SNA, one of the two network addressable units (NAUs) having an active session.

session services. In SNA, one of the types of network services in the control point (CP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that the SSCP initiate or terminate sessions between logical units. See *configuration services*, *maintenance services*, and *management services*.

session-establishment request. In VTAM, a request to an LU to establish a session. For the primary logical unit (PLU) of the requested session, the session-establishment request is the CINIT sent from the system services control point (SSCP) to the PLU. For the secondary logical unit (SLU) of the requested session, the session-establishment request is the BIND sent from the PLU to the SLU.

session-initiation request. In SNA, an Initiate or logon request from a logical unit (LU) to a control point (CP) that an LU-LU session be activated.

session-level pacing. In SNA, a flow control technique that permits a receiver to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing* and *virtual route pacing*.

shared. Pertaining to the availability of a resource to more than one use at the same time.

simulated logon. A session-initiation request generated when a VTAM application program issues a SIMLOGON macroinstruction. The request specifies a logical unit (LU) with which the application program wants a session in which the requesting application program will act as the primary logical unit (PLU).

single-domain network. In SNA, a network with one system services control point (SSCP). Contrast with *multiple-domain network*.

SLU. Secondary logical unit.

SMF. System management facility.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNA network interconnection. The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

SP. Service point.

SR. Service reminder.

SSCP. System services control point.

SSCP ID. In SNA, a number that uniquely identifies a system services control point (SSCP). The SSCP ID is used in session activation requests sent to physical units (PUs) and other SSCPs.

SSCP rerouting. In SNA network interconnection, the technique used by the gateway system services control point (SSCP) to send session-initiation request units (RUs), by way of a series of SSCP-SSCP sessions, from one SSCP to another, until the owning SSCP is reached.

SSCP-LU session. In SNA, a session between a system services control point (SSCP) and a logical unit (LU); the session enables the LU to request the SSCP to help initiate LU-LU sessions.

SSCP-PU session. In SNA, a session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

SSCP-SSCP session. In SNA, a session between the system services control point (SSCP) in one domain and the SSCP in another domain. An SSCP-SSCP session is used to initiate and terminate cross-domain LU-LU sessions.

SSP. System Support Programs (IBM licensed program). Its full name is Advanced Communications Function for System Support Programs. Synonymous with *ACF/SSP*.

ST. Session configuration screen abbreviation.

start option. In VTAM, a user-specified or IBM-supplied option that determines certain conditions that are to exist during the time a VTAM system is operating. Start options can be predefined or specified when VTAM is started.

statement. A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

station. (1) One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. (2) One or more computers, terminals, or devices at a particular location.

subarea. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea address. In SNA, a value in the subarea field of the network address that identifies a particular subarea. See also *element address*.

subarea host node. A host node that provides both subarea function and an application program interface (API) for running application programs. It provides system services control point (SSCP) functions, subarea node services, and is aware of the network configuration. See *boundary node*, *communication management configuration host node*, *data host node*, *host node*, *node*, *peripheral node*, and *subarea node*. See also *boundary function* and *node type*.

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See *boundary node*, *host node*, *node*, *peripheral node*, and *subarea host node*. See also *boundary function* and *node type*.

subarea path control. The function in a subarea node that routes message units between network addressable units (NAUs) and provides the paths between them. See *path control* and *peripheral path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

subarea PU. In SNA, a physical unit (PU) in a subarea node.

subsystem. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

supervisor. The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations.

switched major node. In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

system management facility (SMF). A standard feature of MVS that collects and records a variety of system and job-related information.

system services control point (SSCP). In SNA, a central location point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

system services control point (SSCP) domain. The system services control point and the physical units (PUs), logical units (LUs), links, link stations and all the resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

System Support Programs (SSP). An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

takeover. The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem. See *resource takeover*.

task. A basic unit of work to be accomplished by a computer. The task is usually specified to a control program in a multiprogramming or multiprocessing environment.

task panel. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

TCAS. Terminal control address space.

telecommunication line. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

teletypewriter exchange service (TWX). Teletypewriter service in which suitably arranged teletypewriter stations are provided with lines to a central office for access to other such stations throughout the U.S. and Canada. Both baudot and ASCII-coded machines are used. Business machines may also be used with certain restrictions.

terminal. A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

terminal control address space (TCAS). The part of TSO/VTAM that provides logon services for TSO/VTAM users.

TERMINATE. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

TH. Transmission header.

threshold. In the NetView program, refers to a percentage value set for a resource and compared to a calculated error-to-traffic ratio.

threshold. In NPM, high or low values supplied by the user to monitor data and statistics being collected.

time sharing option (TSO). An optional configuration of the operating system that provides conversational time sharing from remote stations.

time sharing option for VTAM (TSO/VTAM). An optional configuration of the operating system that provides conversational time sharing from remote stations in a network using VTAM.

token. A sequence of bits passed from one device to another along the token ring. When the token has data appended to it, it becomes a frame.

token ring. A network with a ring topology that passes tokens from one attaching device to another. For example, the IBM Token-Ring Network.

transmission control (TC) layer. In SNA, the layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and deciphers end-user data. Transmission control has two components: the connection point manager and session control. See also *half-session*.

transmission group (TG). In SNA, a group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages. A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

transmission header (TH). In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

transmission line. Synonym for *telecommunication line*.

transmission priority. In SNA, a rank assigned to a path information unit (PIU) that determines its precedence for being selected by the transmission group control component of path control for forwarding to the next subarea node of the route used by the PIU.

TSO. Time sharing option.

TSO/VTAM. Time sharing option for VTAM.

TWX. Teletypewriter exchange service.

type 2.1 node (T2.1 node). A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.0 nodes. Type 2.1 nodes can be directly attached to one another using peer-to-peer protocols. See *end node*, *node*, and *subarea node*. See also *node type*.

type 2.1 node (T2.1 node) control point domain. The CP, its logical units (LUs), links, link stations, and all resources that it activates and deactivates.

UNBIND. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

unformatted. In VTAM, pertaining to commands (such as LOGON or LOGOFF) entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user's unformatted system services definition table. Synonymous with *character-coded*. Contrast with *field-formatted*.

unformatted system services (USS). In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as a logon or logoff request into a field-formatted request for processing by formatted system services and translates field-formatted replies and responses into character-coded requests for processing by a logical unit. Contrast with *formatted system services*. See also *converted command*.

uninterpreted name. In SNA, a character string that a system services control point (SSCP) is able to convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

upstream. In the direction of data flow from the end user to the host. Contrast with *downstream*.

user. Anyone who requires the services of a computing system.

user exit. A point in an IBM-supplied program at which a user routine may be given control.

user exit routine. A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language (HLL).

user-application network. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. Contrast with *public network*.

USERVAR. Contains an application name used to route a session-establishment request to the currently active application subsystem.

USS. Unformatted system services.

value. (1) (TC97) A specific occurrence of an attribute, for example, "blue" for the attribute "color." (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

variable. In the NetView command list language, a character string beginning with & that is coded in a command list and is assigned a value during execution of the command list.

vector. The MAC frame information field.

verb. (1) In SNA, the general name for a transaction program's request for communication services. (2) In VTAM, a programming language element in the logical unit (LU) 6.2 application program interface (API) that causes an LU 6.2 function to be performed.

virtual machine. A functional simulation of a computer and its associated devices.

Virtual Machine (VM). A licensed program whose full name is the Virtual Machine/System Product (VM/SP). It is a software operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

virtual route (VR). In SNA, a logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER)*, *path*, and *route extension*.

virtual route (VR) pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing* and *session-level pacing*.

virtual route pacing response (VRPRS). A nonsequenced, supervisory path information unit (PIU) that flows at network priority. It may overtake VR-sequenced PIUs and consists of a transmission header with no basic information unit (BIU) data.

virtual route selection exit routine. In VTAM, an optional installation exit routine that modifies the list of virtual routes associated with a particular class of service before a route is selected for a requested LU-LU session.

virtual storage. (ISO) The notion of storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations.

Virtual Storage Extended (VSE). An IBM licensed program whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VIT. VTAM internal trace.

VM. Virtual Machine. Its full name is Virtual Machine/System Product. Synonymous with *VM/SP*.

VM/ESA. Virtual Machine/Enterprise Systems Architecture.

VM/SNA console support (VSCS). A VTAM component for the VM environment that provides Systems Network Architecture (SNA) support. It allows SNA terminals to be virtual machine consoles.

VM/SP. Virtual Machine/System Product. Synonym for *VM*.

VR. Virtual route.

VSCS. VM/SNA console support.

VSE. Virtual Storage Extended. Synonymous with *VSE/Advanced Functions*.

VSE/Advanced Functions. The basic operating system support needed for a VSE-controlled installation. Synonym for *VSE*.

VSE/ESA. Virtual Storage Extended/Enterprise Systems Architecture.

VTAM. Virtual Telecommunications Access Method (IBM licensed program). Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

VTAM definition. The process of defining the user application network to VTAM and modifying IBM-defined characteristics to suit the needs of the user.

VTAM internal trace (VIT). A trace used in VTAM to collect data on channel I/O, use of locks, and storage management services.

VTAM operator. A person or program authorized to issue VTAM operator commands. See *domain operator*, *program operator*, and *network operator (2)*.

VTAM operator command. A command used to monitor or control a VTAM domain. See also *definition statement*.

VTAM Terminal I/O Coordinator (VTIOC). The part of TSO/VTAM that converts TSO TGET, TPUT, TPG, and terminal control macroinstructions into SNA request units.

VTIOC. VTAM Terminal I/O Coordinator.

wide area network. A network that provides data communication capability in geographic areas larger than those serviced by local area networks. Wide area networks may extend across public rights-of-way. Contrast with *local area network*.

window. (1) In SNA, synonym for *pacing group*. (2) On a visual display terminal, a small amount of information in a framed-in area on a panel that overlays part of the panel. (3) In data communication, the number of data packets a data terminal equipment (DTE) or data circuit-terminating equipment (DCE) can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets.

window size. (1) The specified number of frames of information that can be sent before receiving an acknowledgment response. (2) In SNA, synonym for *pacing group size*.

X.21. See *Recommendation X.21 (Geneva 1980)*.

XA. Extended architecture.

XRF. Extended recovery facility.

ZAP disk. The virtual disk in the VM operating system that contains the user-written modifications to VTAM code. See *BASE disk*, *DELTA disk*, *MERGE disk*, and *RUN disk*.

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Bibliography

VTAM Publications

VTAM V3R3 Publications

The following paragraphs describe the library for VTAM V3R3 running under MVS, VM, and VSE.

VTAM Directory of Migration Information (GC31-6429)

This manual contains an overview of the new functions in VTAM V3R3. For each function, it includes a brief description and references to the manuals which contain new information related to the function.

VTAM Network Implementation Guide (SC31-6404)

This manual contains information about how to install VTAM, how to define a network to VTAM, how to test your network definitions, and how to tune VTAM. Use this manual in conjunction with *VTAM Resource Definition Reference*.

VTAM Resource Definition Reference (SC31-6412)

This manual contains the VTAM definition statements and start options. It also has information on the operands of NCP definition statements that affect VTAM. To assist VM users, this book contains an appendix describing VSCS start options. Use this book in conjunction with the *VTAM Network Implementation Guide*.

VTAM Storage Estimates (SK2T-2010, a diskette)

This diskette helps you estimate the storage requirements for VTAM. It contains an interactive program that guides you step-by-step through the process for estimating storage.

VTAM Customization (LY43-0046)

This manual enables a system programmer to customize VTAM. It discusses VTAM, VSCS, and TSO/VTAM installation exit routines, the replaceable constants module, and the communication network management (CNM) routing table.

VTAM Operation (SC31-6408)

This manual enables a system programmer to prepare a "run book" for a VTAM network. This book also serves as a reference manual to programmers and operators requiring detailed information about specific operator commands.

VTAM Messages and Codes (SC31-6405)

This manual contains, in alphanumerical order, all messages and codes issued by VTAM. These messages include VTAM messages for network operators, TSO/VTAM messages for network operators, TSO/VTAM messages for terminal users, USS messages for terminal users, and VSCS messages. This manual can be inserted into the operating system messages manual, if desired, or used as a stand-alone manual.

VTAM Programming (SC31-6409)

This manual describes how to use VTAM macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain. Also included is a dictionary of VTAM macroinstructions.

VTAM Programming for LU 6.2 (SC31-6410)

This manual describes the VTAM LU 6.2 programming interface for host application programs. This manual applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this manual, however.)

VTAM Diagnosis (LY43-0042)

This manual assists system programmers in identifying a VTAM problem, classifying it, and collecting information about the problem in preparation for calling the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.

VTAM Data Areas for MVS (LY43-0043)

VTAM Data Areas for VM (LY43-0045)

VTAM Data Areas for VSE (LY43-0053)

These manuals describe VTAM data areas and can be used to read a VTAM dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Reference Summary (LY43-0047)

This manual is designed as a quick reference for system programmers. This manual contains selected reference information that includes VTAM and VSCS commands, VTAM definition statements, VTAM start options, VTAM macroinstructions, and VTAM and VSCS trace formats.

Planning and Reference for NetView, NCP, and VTAM (SC31-6092)

This manual describes how to plan for NetView V2R1, NCP V5R3, SSP V3R5, and VTAM V3R3. It explains the functions available with NetView, NCP, and VTAM, the advantages of using them in different situations, and how to plan for the functions readers want to use. The reference part of the manual contains cross-product or cross-task reference information, which may or may not be related to planning. The manual also contains NCP storage estimates.

Bibliography and Master Index for NetView, NCP, and VTAM (GC31-6430) (when available)

This book contains a list of manuals that might be useful to someone planning, installing, or using a network that contains NetView V1R3, NCP V5R3, SSP V3R5, and VTAM V3R3. It also contains an index of topics discussed in the products' libraries. Each entry in the index is followed by the titles of the manuals discussing that topic.

VTAM V3R2 Publications

You may order additional copies of the VTAM V3R2 publications by the following order numbers.

Note: Several of these manuals also contain information about VTAM V3R1.2 for VM and VSE, V3R1.1 for MVS and VM, and V3R1 for VSE.

VTAM Installation and Resource Definition (SC23-0111)

VTAM Customization (LY30-5614)

VTAM Directory of Programming Interfaces for Customers (GC31-6403)

VTAM Operation (SC23-0113)

VTAM Messages and Codes (SC23-0114)

VTAM Programming (SC23-0115)

VTAM Programming for LU 6.2 (SC30-3400)

VTAM Diagnosis (LY30-5601)

VTAM Data Areas for MVS (LY30-5592)

VTAM Data Areas for VM (LY30-5593)

VTAM Data Areas for VSE (LY30-5594)

VTAM Reference Summary (LY30-5600)

VTAM V3R2 Enhancements (LD35-0270)

VTAM Version 3 for VM/9370 (SD35-0271)

VTAM V3R1.2 Publications

You may order additional copies of the VTAM V3R1.2 publications by the following order numbers:

VTAM Expanded Network Capabilities Support (LD21-0019)

VTAM Directory of Programming Interfaces for Customers (GC31-6402)

VTAM V3R1.1 Publications

You may order additional copies of the VTAM V3R1.1 publications by the following order numbers:

VTAM Installation and Resource Definition (SC23-0111)

VTAM Customization (SC23-0112)

VTAM Operation (SC23-0113)

VTAM Message and Codes (ST23-0114)

VTAM Programming (SC23-0115)

VTAM Diagnosis Guide (SC23-0116)

VTAM Diagnosis Reference (LY30-5582)

VTAM Data Areas for MVS (LY30-5584)

VTAM Data Areas for VM (LY30-5583)

VTAM Reference Summary (SC23-0135)

VTAM V3R1 Publications

You may order additional copies of the VTAM V3R1 publications by using the following pseudonumbers:

VTAM Installation and Resource Definition (ST23-0110)

VTAM Customization (ST23-0112)

VTAM Operation (ST23-0113)

VTAM Programming (ST23-0115)

VTAM Diagnosis Reference (LT70-5582)

VTAM Message and Codes (ST23-0114)

The following VTAM V3R1 publications are still orderable by their original order numbers:

VTAM Messages and Codes for VM (SC30-3275)

VTAM Data Areas for MVS (LY30-5581)

VTAM Data Areas for VM (LY30-5580)

VTAM Data Areas for VSE (LY30-5579)

VTAM Library Supplement for X.21 SHM/MPS (SD21-0010)

Related Publications

NetView Release 3 Publications

The following list shows the publications associated with Release 3 of the NetView program.

Learning About NetView: Operator Training (SK2T-0292)

NetView Installation and Administration Guide (SC31-6018)

NetView Administration Reference (SC31-6014)

NetView Tuning Guide (SC31-6079)

NetView Customization Guide (SC31-6016)

NetView Customization: Using PL/I and C (SC31-6037)

NetView Customization: Using Assembler (SC31-6078)

NetView Customization: Writing Command Lists (SC31-6015)

NetView Operation Primer (SC31-6020)

NetView Operation (SC31-6019)

NetView Command Summary (SX75-0026)

NetView Problem Determination and Diagnosis (LY43-0001)

NetView Problem Determination Supplement for Management Services Major Vectors 0001 and 0025 (LD21-0023)

NetView Resource Alerts Reference (SC31-6024)

NetView Storage Estimates (SK2T-1988, a diskette)

Console Automation Using NetView: Planning (SC31-6058)

NCP Version 4 Publications

The following list shows the publications for NCP Version 4.

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP Migration Guide (SC30-3252)

NCP, SSP, and EP Resource Dèfinition Guide (SC30-3349)

NCP, SSP, and EP Resource Definition Reference (SC30-3254)

NCP Customization Guide (LY30-5571)

NCP Customization Reference (LY30-5612)

SSP Customization (LY43-0021)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP and EP Reference (LY30-5569)

NCP and EP Reference Summary and Data Areas (LY30-5570)

NCP Version 5 Publications

The following list shows the publications for NCP Version 5.

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP Migration Guide (SC30-3440)

NCP, SSP, and EP Resource Definition Guide (SC30-3447)

NCP, SSP, and EP Resource Definition Reference (SC30-3448)

NCP Customization Guide (LY30-5606)

NCP Customization Reference (LY30-5607)

SSP Customization (LY43-0021)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP and EP Reference (LY30-5605)

NCP and EP Reference Summary and Data Areas (LY30-5603)

Other Related Publications

MVS/XA Publications

MVS/XA System Programming Library: Initialization and Tuning (GC28-1149)

MVS/XA System Management Facilities (SMF) (GC28-1153)

MVS/XA Supervisor Services and Macro Instructions (GC28-1154)

MVS/ESA Publications

MVS/ESA System Programming Library: Initialization and Tuning (GC28-1828)

MVS/ESA System Programming Library: System Management Facilities (GC28-1819)

VM/SP Publications

VM/SP Group Control System Command and Macro Reference (SC24-5250)

VM/SP Installation Guide (SC24-5237)

VM/SP Operator's Guide (SC19-6202)

VSE/Advanced Functions Publications

VSE/Advanced Functions System Control Statements (SC33-6095)

TSO and TSO/VTAM Publications

Each of the following books is referred to as *TSO Guide to Writing a Terminal Monitor Program.* Use the one that applies to your operating system.

OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor (GC28-0648)

MVS/XA TSO Guide to Writing a Terminal Monitor Program or Command Processor (GC28-1295)

MVS/XA TSO Extensions TSO Guide to Writing a Terminal Monitor Program or a Command Processor (SC28-1136)

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Index

A

ABEND (abnormal end)
- function in session management exit routine 115
- session management exit routine 7
- subtask limit 73
- VR selection exit routine 53

abnormal end (ABEND)
- function in session management exit routine 115
- session management exit routine 7
- subtask limit 73
- VR selection exit routine 53

accounting
- exit routine (ISTAUCAG) 58
- session management function 13

adjacent SSCP message control 74

adjacent SSCP selection function
- description of 15
- final register contents 16
- parameter list 16

alias name translation control 74

alias name translation facility, for CNM table 85

alias selection function (MVS VM)
- description of 17
- final register contents 18
- input parameter list 33
- output parameter list 38
- parameter list 17
- prologue code
  - first entry of input parameter list 132
  - second entry of input parameter list 133
  - suffix allocation subroutine 135
- sample processing 130
- sample session flow 39—47

authorization
- exit routine (ISTAUCAT) 60
- session management function
  - initial authorization 10
  - secondary authorization 12

B

begin function
- description of 9
- final register contents 10
- parameter list 10
- prologue code 119
- sample processing 117

BSC 3270 poll failures 73

C

CID table size 80

CNM routing table
- description of 85
- example for two application programs 87
- IBM supplied information 85
- installing 85
- request types routed by 86
- sample table 111
- structure
  - header format 86
  - table entry format 86
- unsolicited requests 87
- user-written 85
- with user-written alias translation facility 87

constants module (ISTRACON)
- field, description
  - RACABCNT 73
  - RACABINT 73
  - RACALIAS 74
  - RACBSNAP 72
  - RACBUFSZ 81
  - RACCITSZ 80
  - RACEAS 80
  - RACHNTSZ 80
  - RACHSRT 76
  - RACHXSRT 76
  - RACINNBL 75
  - RACINOP 83
  - RACINOPT 73
  - RACLBFSZ 83
  - RACLTXMN 83
  - RACLTXMX 83
  - RACMCPBF 72
  - RACMIHTM 83
  - RACMLUBF 72
  - RACMXBUF 82
  - RACNTWRE 84
  - RACONSRT 78
  - RACPDBFS 75
  - RACPIULM 84
  - RACSASUP 81
  - RACSAWLM 84
  - RACSAWPK 84
  - RACSMDR 84
  - RACSSDTO 75
  - RACSSMSG 74
  - RACTRFLG 79
  - RACVCNT 75
  - RACXNSRT 78
- function listing 69
- modifying 72

constants module (ISTRACON) *(continued)*
  summary table 70

D

data manipulation exit routines, VSCS
  description of 93
  display devices
    description of 96
    final register contents 97
    initial register contents 96
    parameter list contents 96
  installing 97
  keyboard/printer and TWX devices
    description of 93
    design considerations 95
    final register contents 95
    initial register contents 95
    parameter list contents 95
DCSS (discontiguous shared segment) 66
DELIVER RU 86
discontiguous shared segment (DCSS) 66
display devices
  inbound data stream, DTIPDDSI 96, 97
  outbound data stream, DTIPDDSO 96, 97
  VSCS data manipulation exit routines 96
downstream load utility (DSLU)
  message routing 85
DSLU (downstream load utility)
  message routing 85
DTIPDNTI exit routine 93, 95
DTIPDNTO exit routine 93, 95
dumping/loading a remote NCP 81

E

EAS value
  for VTAM application 80
end function
  description of 18
  final register contents 18
  parameter list 18
  prologue code 138
  sample processing 138
environment vector list 19
ER-TESTED RU 87
exit options 24
exit routines, TSO (time sharing option)
  attention handler for 3270 terminals (IKTIDSX3) 103
  attention handler for 3767 and 3770 terminals (IKTRTX3) 106
  editing for 3767, 3770, and 2741 terminals (IKTRTX4) 106
  editing on unsupported terminals (IKTGETXT) 102
  error handling for unsupported terminals (IKTCASX1) 102
  input edit for 3270 terminals (IKTIDSX2) 103

exit routines, TSO (time sharing option) *(continued)*
  input edit for 3767 and 3770 terminals (IKTRTX2) 106
  I/O manager initialization (IKTINX2) 105
  logon edit (IKTINX1) 104
  output edit for WTTY and TWX terminals (IKTWTX1) 107
  output edit for 3270 terminals (IKTIDSX1) 103
  output edit for 3767, 3770, and 2741 terminals (IKTRTX1) 105
  TGET edit for 3270 terminals (IKTIDSX4) 104
exit routines, user
  replacing 66
  session accounting (ISTAUCAG)
    design requirements 59
    final register contents 59
    general discussion 58
    initial register contents 58
  session authorization (ISTAUCAT)
    design requirements 61
    final register contents 63
    general discussion 60
    initial register contents 61
    parameter list contents 61
  session management (ISTEXCAA)
    abend function in 115
    description of 3
    design requirements 47
    initial register contents 7
    parameter descriptions 19
    parameter list structure 9
    sample exit routine 113
  TPRINT processing (ISTRAEUE)
    final register contents 64
    general discussion 63
    initial register contents 63
    parameter list structure 63
  virtual route selection (ISTEXCVR)
    changing the VR selection list 49
    design requirements 53
    final register contents 54
    general discussion 48
    initial register contents 49
    parameter list contents 49
  VR pacing window size calculation (ISTPUCWC)
    bounds specification 54
    description of 54
    design requirements 56
    final register contents 56
    initial register contents 55
    parameter list contents 55
    used with IMS 57
exit routines, VSCS data manipulation
  description of 93
  display devices
    description of 96
    final register contents 97
    initial register contents 96

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

exit routines, VSCS data manipulation *(continued)*
  display devices *(continued)*
    parameter list contents 96
  installing 97
  keyboard/printer and TWX devices
    description of 93
    design considerations 95
    final register contents 95
    initial register contents 95
    parameter list contents 95
explicit route characteristics table 58
extended recovery facility (XRF)
  secondary function code for 20
  session takeover accounting 14

F

final accounting function
  description of 13
  final register contents 13
  parameter list 13
function code and related session information
  description of 20
  function code 21
  INIT OTHER CD processing 20
  related session information 21, 23

G

gateway
  class-of-service names 29
  information vector (OLU) 29
  NCP name 29
  path selection function (MVS VM)
    description of 13
    final register contents 14
    locating real LU name, sample code 126
    modifying GWPATH list, sample code 129
    parameter list 13
    sample processing 125
  path selection list 29
GLOBAL LOADLIB command 90
GWPATH definition statement 7, 13

H

host node table size 80

I

IKTCASX1, error handling for non-supported terminals 102
IKTGETXT, editing on non-supported terminals 102
IKTIDSX1, output editing for 3270 terminals 103
IKTIDSX2, input editing for 3270 terminals 103
IKTIDSX3, attention handler for 3270 terminals 103
IKTIDSX4, TGET edit for 3270 terminals 104
IKTINX1, logon edit 104
IKTINX2, I/O manager initialization 105
IKTRTX1, output edit for 3767, 3770, and 2741 terminals 105
IKTRTX2, input edit for 3767 and 3770 terminals 106
IKTRTX3, attention handler for 3767 and 3770 terminals 106
IKTRTX4, edit for 3767, 3770 and 2741 terminals 106
IKTWTX1, output edit for WTTY and TWX terminals 107
IMS (Information Management System) 57
Information Management System (IMS) 57
INIT OTHER CD processing
  alias selection
    function 17
    output parameter list 38
  DLU adjacent SSCP vector 33
  function code and related session information 20
  initial authorization function
    description of processing 10, 11
    parameter list 11
    when driven 4
  OLU adjacent SSCP vector 32
  sample cross-network session 6
INIT-LOAD RU 87
initial accounting function
  description of 13
  final register contents 13
  parameter list 13
initial authorization function
  description of 10
  final register contents 11
  for INIT OTHER CD processing 10, 11
  parameter list 11
installing
  CNM routing tables 85
  tables and modules in VM 89
  TPRINT processing exit routine 66
  VSCS data manipulation exit routines 97
  VTAM exit routines 66
IRN transmissions, storing 75
ISTAUCAG, accounting exit routine 66
ISTAUCAT, authorization exit routine 66
ISTEXCAA (session management exit routine)
  adjacent SSCP selection function
    description of 15
    parameter list 16
    register contents 16
  alias selection function (MVS VM)
    description of 3, 17
    INIT OTHER CD processing 17, 38
    input parameter list 33
    output parameter list 38
    parameter list 17
    prologue code 132
    register contents 18
    sample processing 130

ISTEXCAA (session management exit routine) *(continued)*
  begin function
    description of 9
    parameter list 10
    prologue code 119
    register contents 10
    sample processing 117
  description of 3
  design requirements 47
  end function
    description of 18
    parameter list 18
    prologue code 138
    register contents 18
    sample processing 138
  example of functions 4
  for gateway path selection 3
  for SSCP selection 3
  gateway path selection function (MVS VM)
    description of 13
    locating real LU name, sample code 126
    modifying GWPATH list, sample code 129
    parameter list 13
    register contents 14
    sample processing 125
  initial and final accounting functions
    description of 13
    parameter list 13
    register contents 13
  initial authorization function
    description of 10
    INIT OTHER CD processing 10, 11
    parameter list 11
    register contents 11
  parameter descriptions 19
  parameter list structure 9
  parameters
    environment vector list 19
    exit options 24
    function code and related session information 20
    gateway path selection list 29
    OLU gateway information vector 29
    PLU resource identifier control vector 25
    session identifier 28
    SLU resource identifier control vector 25
    time-of-day field 29
    user data field 23
  register contents 7
  sample cross-network session 5
  sample cross-network session for INIT OTHER CD 6
  sample exit routine
    description of 113
    function processing and prologue code 117—138
    function selection 116
    initialization 113

ISTEXCAA (session management exit routine) *(continued)*
  secondary authorization function
    description of 12
    parameter list 12
    prologue code 123
    register contents 12
    sample processing 121
  session takeover accounting function
    description of 14
    parameter list 15
    register contents 15
  with cross-network or single-network sessions 3
ISTEXCVR, virtual route selection routine 48, 66
ISTMGC00, an RU supplemental table 85
ISTMGC01, a default CNM routing table 85
ISTPUCWC, virtual route pacing window size calculation routine 54, 66
ISTRACON (constants module)
  field, description
    RACABCNT 73
    RACABINT 73
    RACALIAS 74
    RACBSNAP 72
    RACBUFSZ 81
    RACCITSZ 80
    RACEAS 80
    RACHNTSZ 80
    RACHSRT 76
    RACHXSRT 76
    RACINNBL 75
    RACINOP 83
    RACINOPT 73
    RACLBFSZ 83
    RACLTXMN 83
    RACLTXMX 83
    RACMCPBF 72
    RACMIHTM 83
    RACMLUBF 72
    RACMXBUF 82
    RACNTWRE 84
    RACONSRT 78
    RACPDBFS 75
    RACPIULM 84
    RACSASUP 81
    RACSAWLM 84
    RACSAWPK 84
    RACSMDR 84
    RACSSDTO 75
    RACSSMSG 74
    RACTRFLG 79
    RACVCNT 75
    RACXNSRT 78
  function listing 69
  modifying 72
  summary table 70

© Copyright IBM Corp. 1984, 1990

ISTRAEUE, TPRINT processing exit routine 64, 66
ISTTRAB, as word 1 in TPRINT 64

K

keyboard/printer devices exit routines
  for input data, DTIPDNTI 93
  for output data, DTIPDNTO 93

M

missing interrupt handling for a CTC 83

N

name field of VTAM macros 39
name translation
  See alias selection function (MVS VM)
NetView
  buffer maximum 75
  message routing 85
network identifier (NETID) registration table, sample 118
NMVT RU 87
NSRU types routed 86

O

OLU (origin logical unit)
  gateway information vector 29
origin logical unit (OLU)
  gateway information vector 29

P

pacing
  response 54
parameter lists
  session management exit routine
    adjacent SSCP selection function 16
    alias selection function 17
    begin function 10
    end function 18
    gateway path selection function 13
    initial and final accounting function 13
    initial authorization function 11
    secondary authorization function 12
    session takeover accounting function for XRF 15
  VSCS data manipulation exit routine
    display devices 96
    keyboard/printer and TWX devices 95
path information unit (PIU)
  data space limit factor 84
  VR pacing windows 54
PIU (path information unit)
  data space limit factor 84
  VR pacing windows 54
PLU resource identifier control vector 25
poll failures 73
PRID (procedure related identifier) 85
private definition library
  VSE and CNM tables 85
procedure related identifier (PRID) 85
process scheduling services trace control 79

R

RACABCNT, ABEND occurrence count 73
RACABINT, ABEND time interval 73
RACALIAS, alias name translation control 74
RACBSNAP, VTAM buffer requests 72
RACBUFSZ, VTAM request unit size 81
RACCITSZ, CID index table size 80
RACEAS, default EAS value 80
RACHNTSZ, host node table size 80
RACHSRT, host sort directory size 76
RACHXSRT, host sort directory size 76
RACINNBL, virtual storage size 75
RACINOPT, BSC 3270 timeout value 73
RACINOP, dump option 83
RACLBFSZ, token ring adapter buffer size 83
RACLTXMN, token ring adapter transmit minimum count 83
RACLTXMX, token ring adapter transmit maximum count 83
RACMCPBF, maximum RU for SSCP 72
RACMIHTM, missing interrupt handling for CTC 83
RACMLUBF, maximum RU for LU-LU 72
RACMXBUF, maximum SAW buffers 82
RACNTWRE, IST5301/5321 message pair limit 84
RACONSRT, other network sort directories 78
RACPDBFS, maximum buffers for NetView 75
RACPIULM, PIU data space limit factor 84
RACSASUP, maximum subareas 81
RACSAWLM, SAW data space limit factor 84
RACSAWPK, SAW data space packing factor 84
RACSMDR, MDR recording option 84
RACSSDTO, switched subarea disconnect timeout 75
RACSSMSG, SSCP table message definition 74
RACTRFLG, trace entry generation 79
RACVCNT, maximum SSCP rerouting count 75
RACXNSRT, other network sort directories 78
RECFMS RU 87
RECMS RU 87
replacement tables and modules, defining 69
request types routed by CNM interface 86
ROUTE-INOP RU 86, 87
RU length, setting maximum 72

S

SAW (session awareness) buffer limit 82
secondary authorization function
  description of 12

secondary authorization function *(continued)*
  final register contents 12
  parameter list 12
  prologue code 123
  sample processing 121
session accounting exit routine (ISTAUCAG)
  description of 58
  design requirements 59
  final register contents 59
  initial register contents 58
session authorization exit routine (ISTAUCAT)
  design requirements 61
  final register contents 63
  initial register contents 61
  parameter list contents 61
session awareness (SAW) buffer limit 82
session identifier 28
session management exit routine (ISTEXCAA)
  adjacent SSCP selection function
    description of 15
    parameter list 16
    register contents 16
  alias selection function (MVS VM)
    description of 3, 17
    INIT OTHER CD processing 17, 38
    input parameter list 33
    output parameter list 38
    parameter list 17
    prologue code 132
    register contents 18
    sample processing 130
  begin function
    description of 9
    parameter list 10
    prologue code 119
    register contents 10
    sample processing 117
  description of 3
  design requirements 47
  end function
    description of 18
    parameter list 18
    prologue code 138
    register contents 18
    sample processing 138
  example of functions 4
  for gateway path selection 3
  for SSCP selection 3
  gateway path selection function (MVS VM)
    description of 13
    locating real LU name, sample code 126
    modifying GWPATH list, sample code 129
    parameter list 13
    register contents 14
    sample processing 125
  initial and final accounting functions
    description of 13
    parameter list 13
session management exit routine (ISTEXCAA) *(continued)*
  initial and final accounting functions *(continued)*
    register contents 13
  initial authorization function
    description of 10
    INIT OTHER CD processing 10, 11
    parameter list 11
    register contents 11
  parameter descriptions 19
  parameter list structure 9
  parameters
    environment vector list 19
    exit options 24
    function code and related session information 20
    gateway path selection list 29
    OLU gateway information vector 29
    PLU resource identifier control vector 25
    session identifier 28
    SLU resource identifier control vector 25
    time-of-day field 29
    user data field 23
  register contents 7
  sample cross-network session 5
  sample cross-network session for INIT OTHER CD 6
  sample exit routine
    description of 113
    function processing and prologue code 117—138
    function selection 116
    initialization 113
  secondary authorization function
    description of 12
    parameter list 12
    prologue code 123
    register contents 12
    sample processing 121
  session takeover accounting function
    description of 14
    parameter list 15
    register contents 15
  with cross-network or single-network sessions 3
session takeover accounting function
  description of 14
  final register contents 15
  parameter list 15
SLU resource identifier control vector 25
snapshot dump 72
SSCP message control, adjacent 74
SSCP rerouting count, maximum 75
SSCP selection function, adjacent
  description of 15
  final register contents 16
  parameter list 16
start-stop devices exit routines
  for input data, DTIPDNTI 93

© Copyright IBM Corp. 1984, 1990

start-stop devices exit routines *(continued)*
  for output data, DTIPDNTO 93
subarea, maximum 81
switch processing function parameter list 15
symbol resolution table size 76, 78
SYSLOG 63
SYS1.LPALIB 66
SYS1.VTAMLIB
  installing CNM tables 85
  link-edit routines 66

T

time sharing option (TSO) exit routines
  attention handler for 3270 terminals (IKTIDSX3) 103
  attention handler for 3767 and 3770 terminals (IKTRTX3) 106
  editing for 3767, 3770, and 2741 terminals (IKTRTX4) 106
  editing on unsupported terminals (IKTGETXT) 102
  error handling for unsupported terminals (IKTCASX1) 102
  input edit for 3270 terminals (IKTIDSX2) 103
  input edit for 3767 and 3770 terminals (IKTRTX2) 106
  I/O manager initialization (IKTINX2) 105
  logon edit (IKTINX1) 104
  output edit for WTTY and TWX terminals (IKTWTX1) 107
  output edit for 3270 terminals (IKTIDSX1) 103
  output edit for 3767, 3770, and 2741 terminals (IKTRTX1) 105
  TGET edit for 3270 terminals (IKTIDSX4) 104
time-of-day field 29
TPRINT processing exit routine (ISTRAEUE)
  description of 63
  final register contents 64
  initial register contents 63
  parameter list structure 63
TR-INQ RU 87
translation, name
  See alias selection function (MVS VM)
TSO (time sharing option) exit routines
  attention handler for 3270 terminals (IKTIDSX3) 103
  attention handler for 3767 and 3770 terminals (IKTRTX3) 106
  editing for 3767, 3770, and 2741 terminals (IKTRTX4) 106
  editing on unsupported terminals (IKTGETXT) 102
  error handling for unsupported terminals (IKTCASX1) 102
  input edit for 3270 terminals (IKTIDSX2) 103
  input edit for 3767 and 3770 terminals (IKTRTX2) 106
  I/O manager initialization (IKTINX2) 105
  logon edit (IKTINX1) 104
  output edit for WTTY and TWX terminals (IKTWTX1) 107

TSO (time sharing option) exit routines *(continued)*
  output edit for 3270 terminals (IKTIDSX1) 103
  output edit for 3767, 3770, and 2741 terminals (IKTRTX1) 105
  TGET edit for 3270 terminals (IKTIDSX4) 104
TWX devices exit routines
  for input data, DTIPDNTI 93
  for output data, DTIPDNTO 93

U

unsolicited network services RUs 85
user data field 23

V

virtual route
  descriptor block table 52
  pacing window size calculation 54
  selection exit routine (ISTEXCVR)
    changing the virtual route selection list 49
    discussed 48
    final register contents 53
    initial register and parameter list contents 49
VMFLKED, for link-editing COS modules 89
VR pacing window size calculation exit routine 54
VRPWSnn operand (PATH definition statement) 55
VSCS data manipulation exit routines
  description of 93
  display devices
    description of 96
    final register contents 97
    initial register contents 96
    parameter list contents 96
  installing 97
  keyboard/printer and TWX devices
    description of 93
    design considerations 95
    final register contents 95
    initial register contents 95
    parameter list contents 95
VTAM
  CNM routing table 85
  constants module (ISTRACON) field
    RACABCNT 73
    RACABINT 73
    RACALIAS 74
    RACBSNAP 72
    RACBUFSZ 81
    RACCITSZ 80
    RACHNTSZ 80
    RACHSRT 76
    RACHXSRT 76
    RACINNBL 75
    RACINOP 83
    RACINOPT 73
    RACLBFSZ 83
    RACLTXMN 83

VTAM *(continued)*
  constants module (ISTRACON) field *(continued)*
    RACLTXMX 83
    RACMCPBF 72
    RACMIHTM 83
    RACMXBUF 82
    RACNTWRE 84
    RACONSRT 78
    RACPDBFS 75
    RACPIULM 84
    RACSASUP 81
    RACSAWLM 84
    RACSAWPK 84
    RACSMDR 84
    RACSSDTO 75
    RACSSMSG 74
    RACTRFLG 79
    RACVCNT 75
    RACXNSRT 78
  exit routines
    replacing 66
    session accounting (ISTAUCAG) 58
    session authorization (ISTAUCAT) 60
    session management (ISTEXCAA) 3
    TPRINT processing (ISTRAEUE) 63
    virtual route selection (ISTEXCVR) 48
    VR pacing window size calculation (ISTPUCWC) 54
    VSCS data manipulation 93
VTAMUSER LKEDIT 89
VTAMUSER LOADLIB
  installing CNM tables 85
  link-edit routines 66
  with installing tables and modules 89

W

window size calculation
  default algorithm 54
writing exit routines 3

X

XRF (extended recovery facility)
  secondary function code for 20
  session takeover accounting 14

LY43-0046-01 © Copyright IBM Corp. 1984, 1990

Reader's Comments

VTAM
Customization
Version 3 Release 3 for MVS, VM, and VSE

Publication No. LY43-0046-01

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply. If we have questions about your comment, may we call you? If so, please include your phone number.

Name

Address

Company or Organization

Phone No.

Reader's Comments
LY43-0046-01

Cut or Fold Along Line

Fold and Tape   **Please do not staple**   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK NORTH CAROLINA 27709-9990

Fold and Tape   **Please do not staple**   Fold and Tape

Cut or Fold Along Line

IBM®

Publication Number
LY43-0046-1

File Number
S370/4300/30XX-50

Program Numbers
5685-085 (MVS/ESA)
5684-052 (VM/9370 for VM/SP)
5665-289 (MVS/XA)
5664-280 (VM/SP)
5684-095 (VM/ESA)
5666-363 (VSE/ESA)

Printed in USA

Licensed Materials-Property of IBM
Restricted Materials of IBM
©Copyright IBM Corp. 1984, 1990

LY43-0046-01