

SY27-7269-1
File No. S370-30

Systems

**OS/VS2 MVS VTIOC and TCAS
Logic**

**VTAM Terminal I/O
Coordinator (VTIOC)**

Terminal Control Address Space (TCAS)

TSO/VTAM Level 2

IBM

Second Edition (January 1978)

This edition applies to the selectable unit TSO/VTAM for use with OS/VS2 MVS. This edition obsoletes SY27-7269-0 and Technical Newsletter SN31-0676. Refer to the Summary of Amendments dated January 30, 1978, for a description of changes.

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

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

A form for readers' comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to IBM Corporation, Programming Publications, Department 63T, Kingston, New York 12401. Comments become the property of IBM.

SUMMARY OF AMENDMENTS (January 30, 1978)
TO SY27-7269-0 BY REVISION SY27-7269-1

OS/VS2 MVS TSO/VTAM

Changed Documentation

- Support is added for IBM LU2-type terminals.
- There are nine new object modules: IKTIMLU2, IKTOMLU2, IKT3270I, IKT3270O, IKTATTN, IKTEXIT, IKTIOFRR, IKT0009D, and IKTIST00. IKTIMLU2 and IKTOMLU2 manage terminal input and output for IBM LU2-type terminals. IKT3270I and IKT3270O handle data input and output for all 3270 terminals. IKTATTN, IKTEXIT, and IKTIOFRR perform service functions for all 3270 terminals and are used by IKTIMIDS, IKTOMIDS, IKTIMLU2, and IKTOMLU2. IKT0009D determines the appropriate branch for terminal control macros, and IKTIST00 schedules the lost terminal exit routine.
- There are two new load modules: IKTIOM00 and IKTIOM03. IKTIOM00 contains IKTEXIT, IKTIOFRR, and IKTIST00. IKTIOM03 contains IKTIMLU2 and IKTOMLU2, IKTATTN is in IKTIOM02, and IKT0009D is in IGC0009D.



PREFACE

This publication describes the internal organization and logic of the VTAM terminal I/O coordinator (VTIOC) and the terminal control address space (TCAS). It is intended for people who are debugging or modifying VTIOC or TCAS.

This book is actually two manuals in one; VTIOC and TCAS each have introduction, method of operation, program organization, directory, data areas, and diagnostic aids sections. In addition, there are two appendixes. Following is a synopsis of the information contained in this manual:

- The introduction sections provide overviews of VTIOC and TCAS, and summarize the functions they perform.
- The method of operation sections use HIPO (hierarchy plus input-process-output) diagrams to describe the functions performed.
- The program organization sections provide descriptions of each module and diagrams showing module-to-module control flow.
- The directory sections list module names, show their relationships to each other, and provide references to other sections in the manual where the modules are described.
- The data areas sections contain diagrams showing the relationships among VTIOC and TCAS control blocks and work areas, and descriptions of several key internal data areas.
- The diagnostic aids sections list the modules that issue, detect, and contain VTIOC and TCAS messages. The TCAS diagnostic aids section also describes how TCAS records errors.
- Appendix A gives the meanings of all abbreviations used in this publication.
- Appendix B provides information about installation-written exit routines.

Prerequisite Publication

The reader should be familiar with the information presented in the following publication:

OS/VS2 MVS TSO/VTAM System Information, GC27-0046, which contains a complete list of the publications that support TSO/VTAM and the appropriate selectable unit (SU) supplement order numbers.

Associated Publications

Additional information is available in the following publications:

OS/VS2 System Logic Library (7 volumes), SPOF-8210
OS/VS2 System Data Areas (microfiche), SYB8-0606
OS/VS2 System Programming Library: Debugging Handbook (2 volumes), GBOF-8211
OS/VS2 TCAM Logic, SY30-2040
Introduction to VTAM Logic, SY27-7256
VTAM Macro Language Reference, GC27-6995
OS/VS Message Library: VS2 System Messages, GC38-1002
OS/VS2 TSO Terminal Messages Directory, SY28-0654

OS/VS Message Library: VS2 System Codes, GC38-1008
OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor, GC28-0648
OS/VS2 Data Area Usage Table (microfiche), SYB8-0742
OS/VS2 Symbol Usage Table (microfiche), SYB8-0744

CONTENTS

PREFACE i

VTIOC INTRODUCTION 1
 Components 1
 Terminals Supported 2

VTIOC METHOD OF OPERATION 5
 Conventions Used in HIPO Diagrams 5
 VTIOC Visual Table of Contents 7
 VTIOC HIPO Diagrams 8
 MO 1.1 VTIOC Initialization Routine (IKTXINIT) 8
 MO 1.1.1 I/O Manager Initialization Routine (IKTIOM) 9
 MO 1.1.2 OPNDST RPL Asynchronous Exit Routine (IKTRPLIT) 10
 MO 1.2 Extended Logon Routine (IKTXLOG) 11
 MO 1.3 Logon Reconnect Routine (IKTLOGR) 12
 MO 1.4 Extended Logoff Routine (IKTLOGFF) 13
 MO 1.5 LOSTERM Exit Routine (IKLTERM) 14
 MO 1.6 TCAS Termination Routine (IKTAY81) 16
 MO 1.7 Terminal User Address Space Termination
 Routine (IKTAY82) 17
 MO 1.8 TSO/VTAM QTIP 29 Equivalent (IKTAY83) 18
 MO 2.0 Overview of TGET/TPUT Routine 19
 MO 2.1 TGET Routine (IKTVTGET) 20
 MO 2.2 TPUT without ASID Routine (IKVTPUT) 22
 MO 2.3 Calling Address Space TPUT with ASID Routine (IKTASTPT) . 23
 MO 2.3.1 Target Address Space TPUT with ASID
 Routine (IKTXMTPT) 24
 MO 2.4 ESTAE Routine for SVC 93 (IKT93EST) 25
 MO 3.1 3270 (LU0) Terminal Input Manager (IKTIMIDS) 26
 MO 3.2 3270 (LU0) Terminal Output Manager (IKTOMIDS) 28
 MO 3.2.1 Terminal Output Manager Null RU Routine (IKTOMJR) . . . 29
 MO 3.3 3767/3770 Terminal Input Manager (IKTTIMRT) 30
 MO 3.4 3767/3770 Terminal Output Manager (IKTTOMRT) 32
 MO 3.5 3270 (LU2) Terminal Input Manager (IKTIMLU2) 34
 MO 3.6 3270 (LU2) Terminal Output Manager (IKTOMLU2) 36
 MO 3.7 Exit Routine (IKTEXIT) 38
 MO 3.8 Attention Routine (IKTATTN) 40
 MO 3.9 Input Data Handling Routine (IKT3270I) 41
 MO 3.10 Output Data Handling Routine (IKT3270O) 42
 MO 3.11 I/O Functional Recovery Routine (IKTIOFFR) 43
 MO 3.12 Lost Terminal Exit Scheduler (IKTIST00) 45
 MO 4.1 Input Queue Manager (IKTQHIN) 46
 MO 4.2 Output Queue Manager (IKTQHOUT) 49
 MO 5.1 Terminal Control Macro Branching Routine (IKT0009D) 52
 MO 5.2 TCLEARQ Terminal Control Macro Routine (IKT09401) 53
 MO 5.3 STBREAK Terminal Control Macro Routine (IKT09404) 54
 MO 5.4 STCOM Terminal Control Macro Routine (IKT09405) 55
 MO 5.5 STAUTOLN Terminal Control Macro Routine (IKT09409) 56
 MO 6.5 STSIZE Terminal Control Macro Routine (IKT0940A) 57
 MO 5.7 STAUTOCP Terminal Control Macro Routine (IKT0940C) 58
 MO 5.8 SPAUTOPT Terminal Control Macro Routine (IKT0940D) 59
 MO 5.9 RTAUTOPT Terminal Control Macro Routine (IKT0940E) 60
 MO 5.10 STRAN Terminal Control Macro Routine (IKT0940) 61
 MO 5.11 STFSMODE Terminal Control Macro Routine (IKT09412) 62
 MO 5.12 STLINENO Terminal Control Macro Routine (IKT09413) 63
 MO 5.13 STTMPMD Terminal Control Macro Routine (IKT09414) 64

VTIOC PROGRAM ORGANIZATION	65
Module-to-Module Control Flow	66
VTIOC Initialization Module Flow	66
VTIOC Termination Module Flow	67
VTIOC TGET/TPUT Module Flow	68
VTIOC Terminal Input Manager Module Flow	69
VTIOC Terminal Output Manager Module Flow	71
VTIOC Terminal Control Macros Module Flow	72
Module Description	73
 VTIOC DIRECTORY	 119
 VTIOC DATA AREAS	 123
Control Block Overview	124
Input Queue Manager Parameter List	125
Output Queue Manager Parameter List	126
Input Queue Element	128
Output Queue Element	129
 VTIOC DIAGNOSTIC AIDS	 131
Messages	132
System Messages	132
Terminal Messages	132
Abend Codes	133
 TCAS INTRODUCTION	 135
 TCAS METHOD OF OPERATION	 137
Conventions Used in HIPO Diagrams	137
TCAS Visual Table of Contents	137
TCAS HIPO Diagrams	138
MO 6.0 TCAS Main Task	138
MO 6.1 TCAS VTAM Interface	140
MO 6.2 TCAS User Interface	142
MO 6.3 TCAS Console Communication	144
 TCAS PROGRAM ORGANIZATION	 145
Module-to-Module Control Flow	146
Module Descriptions	148
 TCAS DIRECTORY	 169
 TCAS DATA AREAS	 171
Control Block Overview	172
Work Element	173
 TCAS DIAGNOSTIC AIDS	 175
Messages	176
Abend Codes	177
How TCAS Records Errors	177
 APPENDIX A. ABBREVIATIONS	 179
 APPENDIX B. INSTALLATION-WRITTEN EXIT ROUTINES	 181
 INDEX	 183

FIGURES

Figure 1. The Major Elements in a TSO/VTAM Time-Sharing System . . . 1
Figure 2. Legend for HIPO Figures 6
Figure 3. VTIOC Visual Table of Contents 7
Figure 4. VTIOC Initialization Module Flow 66
Figure 5. VTIOC Termination Module Flow 67
Figure 6. VTIOC TGET/TPUT Module Flow 68
Figure 7. VTIOC Terminal Input Manager Module Flow 69
Figure 8. VTIOC Terminal Output Manager Module Flow 71
Figure 9. VTIOC Terminal Control Macros Module Flow 72
Figure 10. VTIOC Control Block Overview 124
Figure 11. TCAS Visual Table of Contents 137
Figure 12. TCAS Module Flow 146
Figure 13. TCAS Control Block Overview 172
Figure 14. A TCAS Work Element 174



VTIOC INTRODUCTION

COMPONENTS

The VTAM terminal I/O coordinator (VTIOC) controls the movement of data between TSO and a user terminal in a TSO/VTAM time-sharing environment (see Figure 1). VTIOC has five functional components:

- Initialization/termination routines. Initialization routines allocate storage for and initialize VTIOC and VTAM control blocks, the input and output queues, and the I/O manager SRBs. They also connect user terminals to TSO. Termination routines free storage and disconnect user terminals from TSO.
- TGET/TPUT routines (SVC 93). These routines handle the movement of data between TSO and the terminal, principally by using the services of the terminal I/O managers and the queue manager.
- Terminal I/O managers. Terminal input managers (TIMs--two for IBM 3270 LU0 and LU2 terminals, one for IBM 3767 and IBM 3770 terminals) obtain input from the terminal for placement on the input queue by issuing the RECEIVE macro. Terminal output managers (TOMs--two for IBM 3270 LU0 and LU2 terminals, one for IBM 3767 and IBM 3770 terminals) send output from the output queue to the terminal by issuing the SEND macro.
- Queue manager. The queue manager adds elements to and removes elements from the input queue and the output queue.
- Terminal control macro routines (SVC 94). These routines allow a command processor to control terminal functions and attributes.

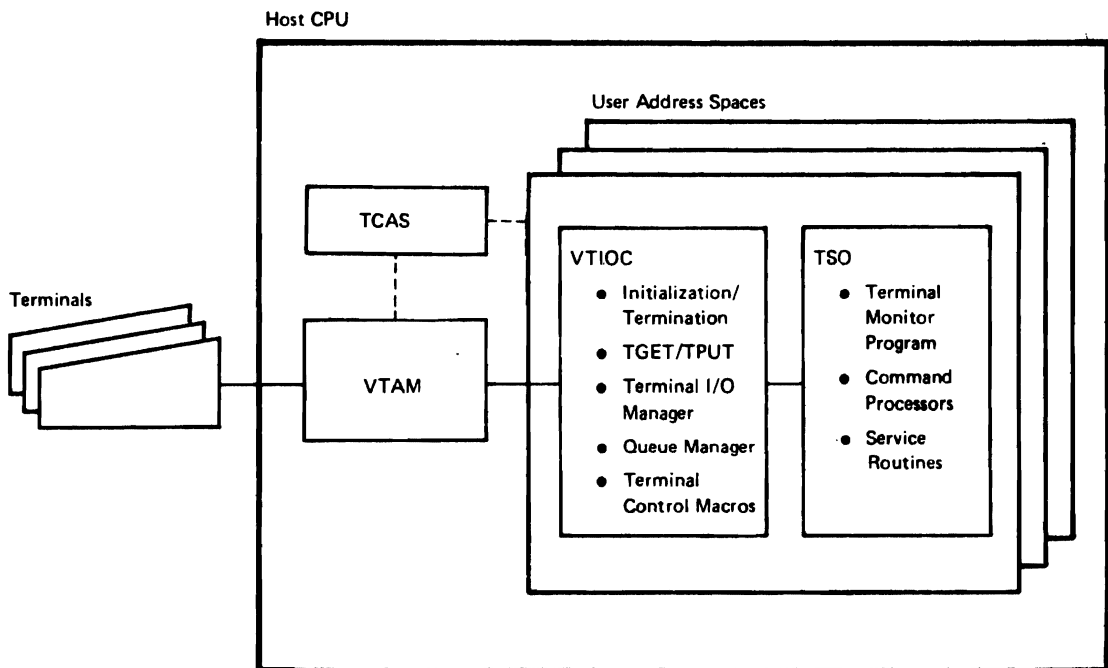


Figure 1. The Major Elements in a TSO/VTAM Time-Sharing System

The basic function of VTIOC is to respond to TGET and TPUT requests. TSC issues a TGET request to obtain input from a terminal. VTAM ensures that the input is available in VTAM's buffer. The terminal input manager, which is scheduled as a service request block (SRB) during VTIOC initialization and is always active, issues a RECEIVE macro to move the input from VTAM's buffer to the terminal input manager's buffer. (Because the terminal input manager is always active a RECEIVE request is always outstanding.) The input queue manager takes the input from the terminal input manager's buffer and puts it on the input queue in one or more queue elements. The VTIOC TGET routine takes the data from the input queue, edits it if requested, and moves it to the TGET requester's (TSO's) buffer.

TSC issues a TPUT request to send output to a terminal. The VTIOC TPUT routine requests the output queue manager to move the output from the TPUT requester's (TSO's) buffer to the output queue. It then schedules the terminal output manager (as a SRB) which issues a SEND macro to send the output to a VTAM buffer. VTAM ensures that the output moves from its buffer to the terminal.

TERMINALS SUPPORTED

The following terminals are supported by TSO/VTAM:

Local terminals:

IBM 3270 Information Display System:

- 3272 Control Unit Models 1 and 2 with attachable 3277 Display Station Models 1 and 2
- 3274 Control Unit Models 1A (SNA) and 1B (non-SNA) with attachable 3277 Display Station Models 1 and 2 or 3278 Display Station Models 1***, 2, 3**, 4** (Model 1A for Level 2 only)*

IBM 3790 Communication System (TSO/VTAM Level 2 only):*

- 3791 Controller Models 1A, 1B, 1C, 2A, and 2B, with attachable 3277 Display Station Models 1 and 2, or 3276 Control Unit Display Station Model 12 (optionally with attachable 3278 Display Station Model 2)

ESC terminals (on nonswitched lines):

IBM 3270 Information Display System:

- 3271 Control Unit Models 1 and 2 with attachable 3277 Display Station Models 1 and 2
- 3274 Control Unit Model 1C with attachable 3277 Display Station Models 1 and 2 or 3278 Display Station Models 1***, 2, 3**, 4**
- 3275 Display Station Models 1 and 2
- 3276 Control Unit Display Station Models 1***, 2, 3**, 4** (optionally with attachable 3278 Display Station Models 1***, 2, 3**, 4**)

SLLC Terminals:

IBM 3270 Information Display System on nonswitched lines:

- 3271 Control Unit Models 11 and 12 with attachable 3277 Display Station Models 1 and 2
- 3274 Control Unit Model 1C (Level 2 only)* with attachable 3277 Display Station Models 1 and 2 or 3278 Display Station Models 1***, 2, 3**, 4**
- 3275 Display Station Models 11 and 12
- 3276 Control Unit Display Station Models 11***, 12, 13**, 14** (optionally with attachable 3278 Display Station Models 1***, 2, 3**, 4**) (Level 2 only)*

IBM 3767 Communication Terminal Models 1, 2, and 3

IBM 3770 Data Communication System:

- 3771 Communication Terminal Models 1, 2, and 3
- 3773 Communication Terminal Models 1, 2, and 3
- 3774 Communication Terminal Models 1 and 2
- 3775 Communication Terminal Model 1

IBM 3790 Communication System (TSO/VTAM Level 2 only):*

- 3791 Controller Models 1A, 1B, 1C, 2A, and 2B, with attachable 3277 Display Station Models 1 and 2, or 3276 Control Unit Display Station Model 12 (optionally with attachable 3278 Display Station Model 2)

In the above list, and throughout this publication, different terminal types have different classifications based on their logical unit (LU) presentation services profile. The LU profile represents a type of logical unit that uses a unique subset of the SNA-defined protocols and data streams for its operation. Terminal groups in the above list marked with an asterisk (*) are classified LU2. The remaining 3270-series terminals are LU0, while the 3767 and 3770 groups are LU1. For further information, see VTAM Macro Language Reference, GC27-6995.

*LU2 device

**Supported as a 1920-character display station

***Supported as a 480-character display station



VTIOC METHOD OF OPERATION

This section uses the HIPO (hierarchy plus input-process-output) technique to graphically describe the functions performed by VTIOC. It contains a visual table of contents and diagrams.

The visual table of contents (Figure 3) contains the names and identification numbers of all the diagrams. There is a diagram for each VTIOC routine. (A routine is a functional unit of processing identified by an object module name or an alternate entry point name.) The level of detail of the diagrams is such that they show the function performed by each routine, calls to other routines, and significant inputs and outputs; they do not show organization of a routine or processing flow within a routine.

Conventions Used in HIPO Diagrams

At the top of each diagram is a diagram ID consisting of an identification number (MC number), a routine name, and a description of the routine. The identification number provides a way of locating a diagram through the visual table of contents or the directory ("VTIOC Directory"). Below the diagram ID are input-process-output blocks and extended description blocks.

The input block shows data that serves as input to the processing steps in the processing block; the output block shows data that is output from the processing steps. The symbols used in and between these blocks are explained in the legend below (see Figure 2). Each processing step is numbered; the number corresponds to a note in the extended description block. The notes provide additional information for the processing steps. The routine name and labels identify the code that performs the function of each step. The references column gives the MC numbers of related diagrams.

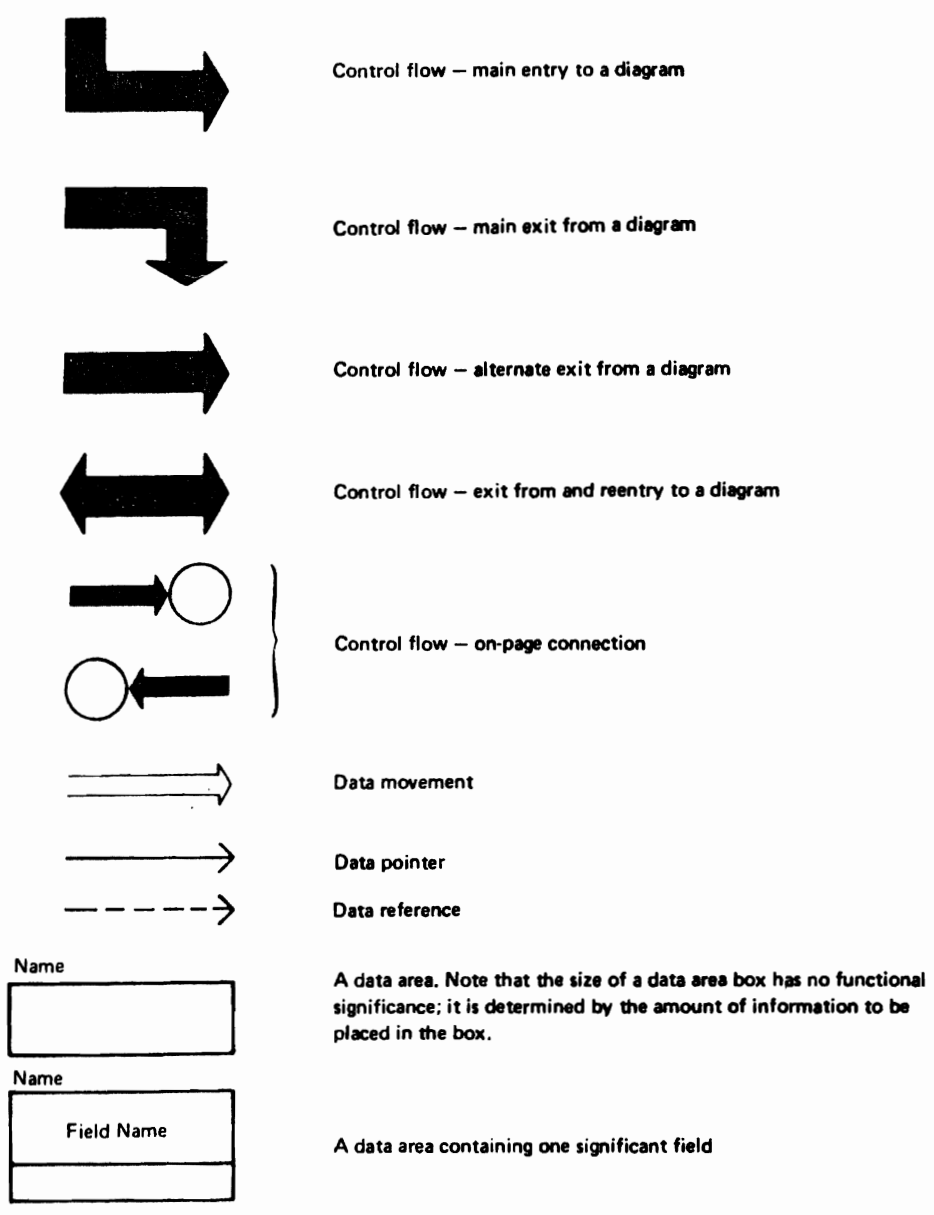


Figure 2. Legend for HIFO Figures

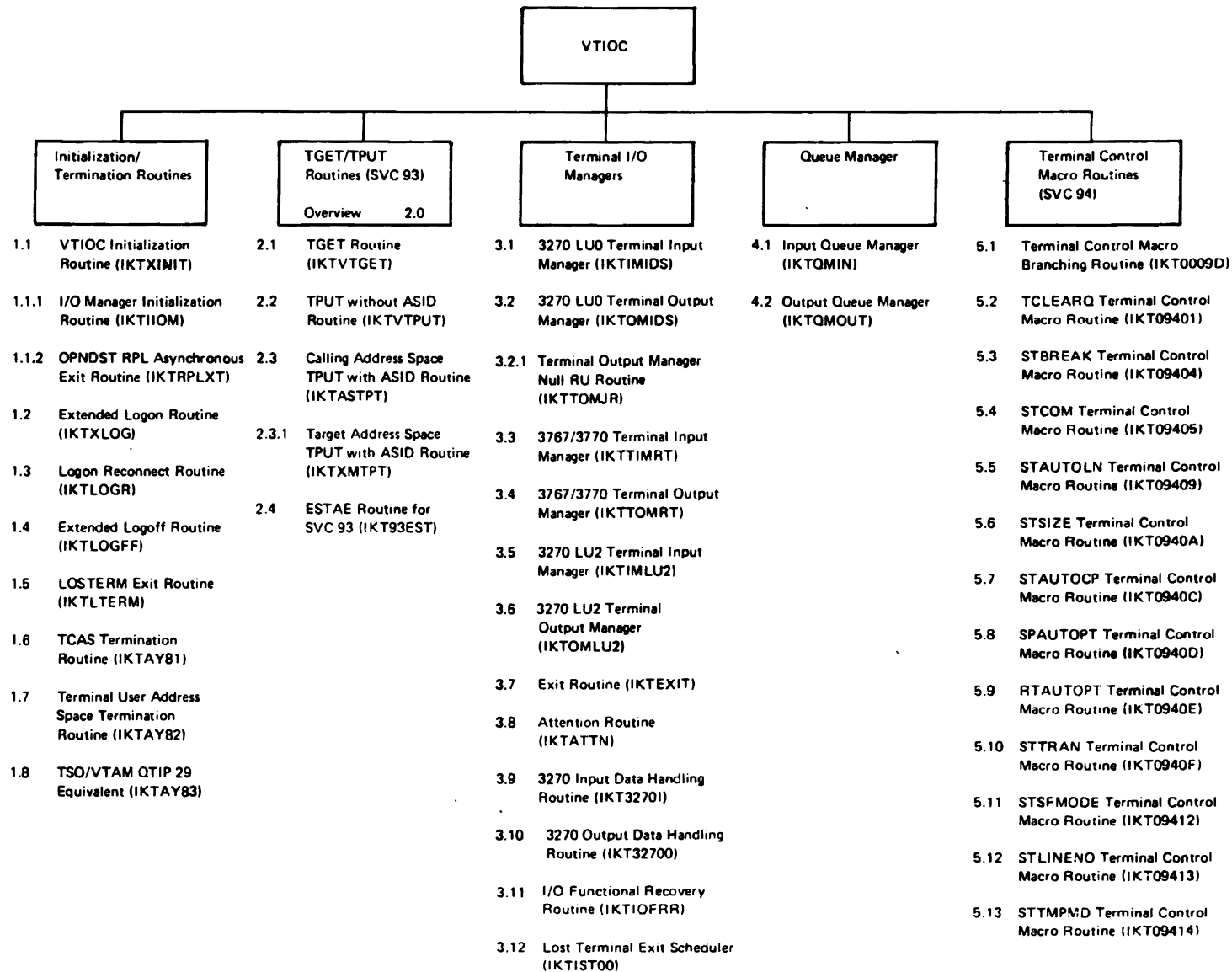
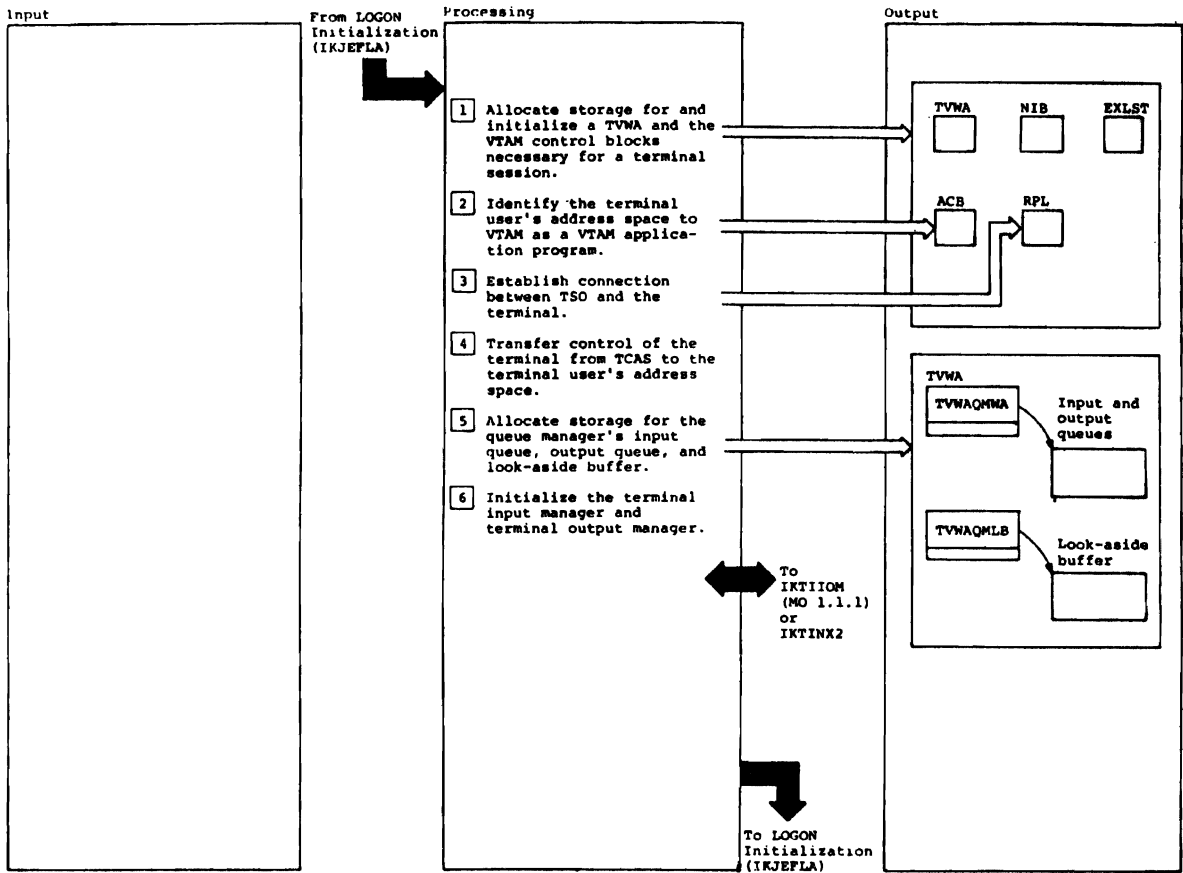


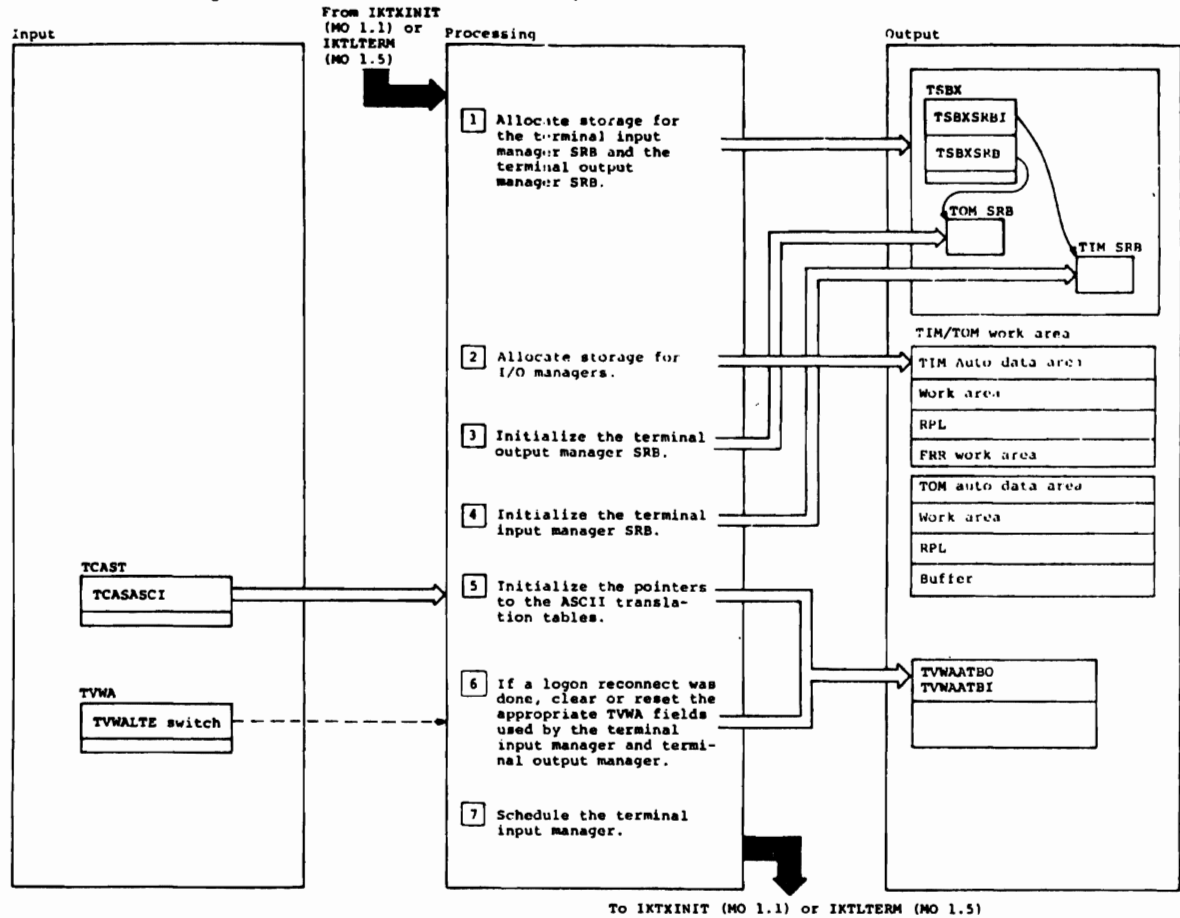
Figure 3. VTIOC Visual Table of Contents

MO 1.1 VTIOC Initialization Routine (IKTXINIT)



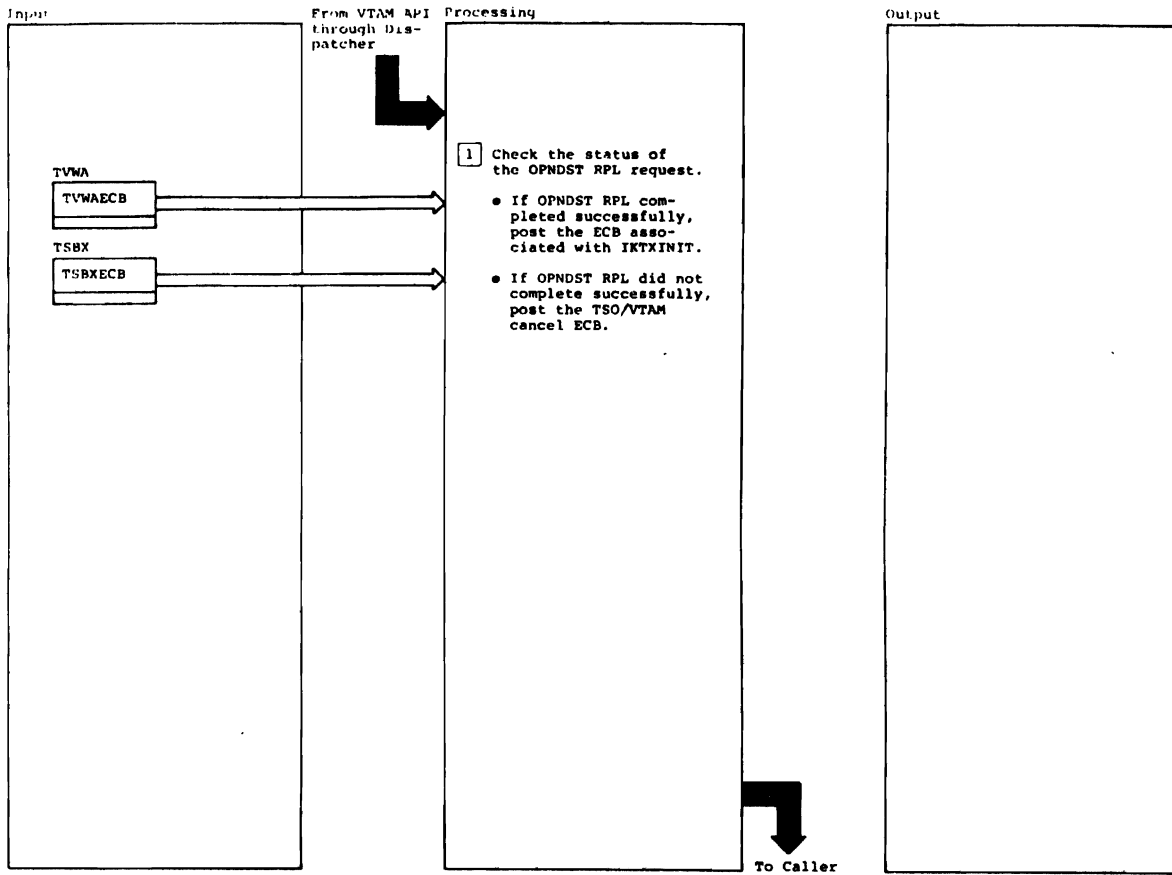
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref	
This routine initializes VTAM control blocks and the TVWA, and transfers control of the terminal from TCAS to the terminal user's address space.	IKTXINIT							
<p>1 Each terminal user's address space has a TVWA, ACB, EXLST, RPL, and NIB associated with it. A GETMAIN macro is used to obtain storage from subpool 229 (key 6); the storage is initialized with the following values:</p> <p>TVWA</p> <ul style="list-style-type: none"> ↑ TCB ↑ NIB ↑ RPL ↑ EXLST ↑ variable storage area ↑ local lock work area ↑ IKTIOM or IKTINX2 size of variable storage area user's application ID <p>ACB</p> <ul style="list-style-type: none"> ↑ application ID password <p>NIB</p> <ul style="list-style-type: none"> ↑ terminal ID ↑ TSB <p>RPL</p> <ul style="list-style-type: none"> ↑ ACB ↑ NIB ↑ IKTRPLXT <p>EXLST</p> <ul style="list-style-type: none"> ↑ IKTLTERM <p>The user address space index placed in the TSBX by TCAS (TSBXAINDX) is used to create the unique application ID assigned to the terminal user's address space. The application ID is stored in TVWAPPL.</p>	IKTXINIT	TVWAINIT						
				<p>2 The terminal's ACB is activated (opened). If the open fails, ABEND OAB is used with reason code 0201.</p> <p>3 OPNDST RPL is issued to acquire the terminal.</p> <p>4 Terminal control is transferred from TCAS to the user's address space by placing a work element with function code X'21' on the terminal handling asynchronous queue of TCAS (TCASSTQN), posting TCAS, then waiting. When TCAS has finished transferring control (by issuing CLSDST PASS in module IKTCAS22), IKTXINIT is posted.</p> <p>5 A GETMAIN macro allocates storage for the queues and for the look-aside buffer. A BLDCPOOL macro divides this storage into equal-size queue elements. (The BUFSIZE value in parmlib member TSOKEY00 specifies the size of each element.) A GETCELL macro allocates one cell for the look-aside buffer. The look-aside buffer is used by the queue manager when there is no more storage in the address space to build a queue element, and a high priority message must be sent.</p> <p>6 The I/O managers are initialized by calling IBM-supplied routine IKTIOM or installation-written routine IKTINX2.</p>	IKTXINIT	DIAGNOSE		
					IKTXINIT	TERMCN		
					IKTXINIT	TERMCN		
					IKTXINIT	IKTQMINT	6.1	
					IKTXINIT		1.1.1	

MO 1.1.1 I/O Manager Initialization Routine (IKTIOM)



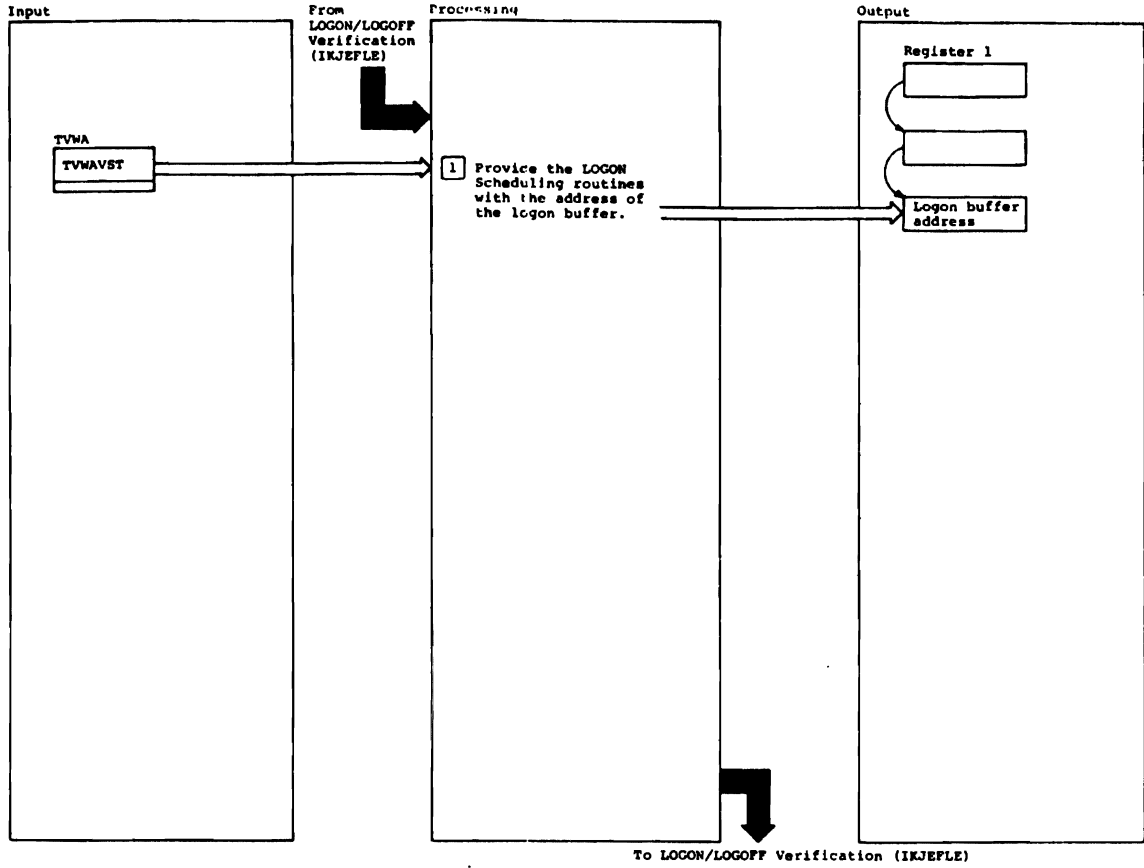
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
This routine initializes the terminal input manager SRB and the terminal output manager SRB.	IKTIOM			SRBPARM = *TOM work area SRBSAVE = 0 SRBPKF = 0 SRBPRIOR = 0			
1 The key is changed to 0 (MODESET macro) and GETMAINs (subpool 239) are issued. If they are unsuccessful, control returns to the caller.	IKTIOM			4 The following fields are set in the TIM SRB: SRBID = 0 SRBFLNK = 0 SRBASCB = *ASCB (PSAAOLD) SRBCPAFF = 0 SRBPASID = address space ID (ASCBASID) SRBPTCB = *TCB for SRB (TVWATCB) SRBEP = *IKTIMIDS or *IKTTIMRT or IKTOMLU2 (TSBXTIM) SRBRMTR = *RMTR for PURGEDQ (TCASBR14) SRBPARM = *TIM work area SRBSAVE = 0 SRBPKF = 0 SRBPRIOR = 0	IKTIOM		
2 If the GETMAIN is unsuccessful the SRBs are freed (FREEMAIN macro) and control returns to the caller. Sizes are: TIM auto data - 1000 bytes (TIMRT) 400 bytes (others) Work area - 3000 bytes RPL - 112 bytes FRR - 400 bytes TOM auto data - 320 bytes (TOMRT) 240 bytes (others) Work area - 744 bytes (TOMRT) 856 bytes (others) RPL - 112 bytes Buffer - 4112 (TOMRT) 4148 (others)				5	IKTIOM	LSTMRTN	
3 The following fields are set in the TOM SRB: SRBID = 0 SRBFLNK = 0 SRBASCB = *ASCB (PSASOLD) SRBCPAFF = 0 SRBPASID = address space ID (ASCBASID) SRBPTCB = *TCB for SRB (TVWATCB) SRBEP = *IKTOMIDS or *IKTOMRT or IKTOMLU2 (TSBXTOM) SRBRMTR = *RMTR for PURGEDQ (TCASBR14)	IKTIOM			6	IKTIOM	LSTMRTN	
				7 The appropriate terminal input manager is scheduled (SCHEDULE SRB) once for each execution of VTIOC; it is always active so that it can receive input from the terminal. (The terminal output manager is scheduled whenever it is not already active and there is output to be sent to the terminal.)	IKTIOM		3.1 3.3

MO 1.1.2 OPNDST RPL Asynchronous Exit Routine (IKTRPLXT)



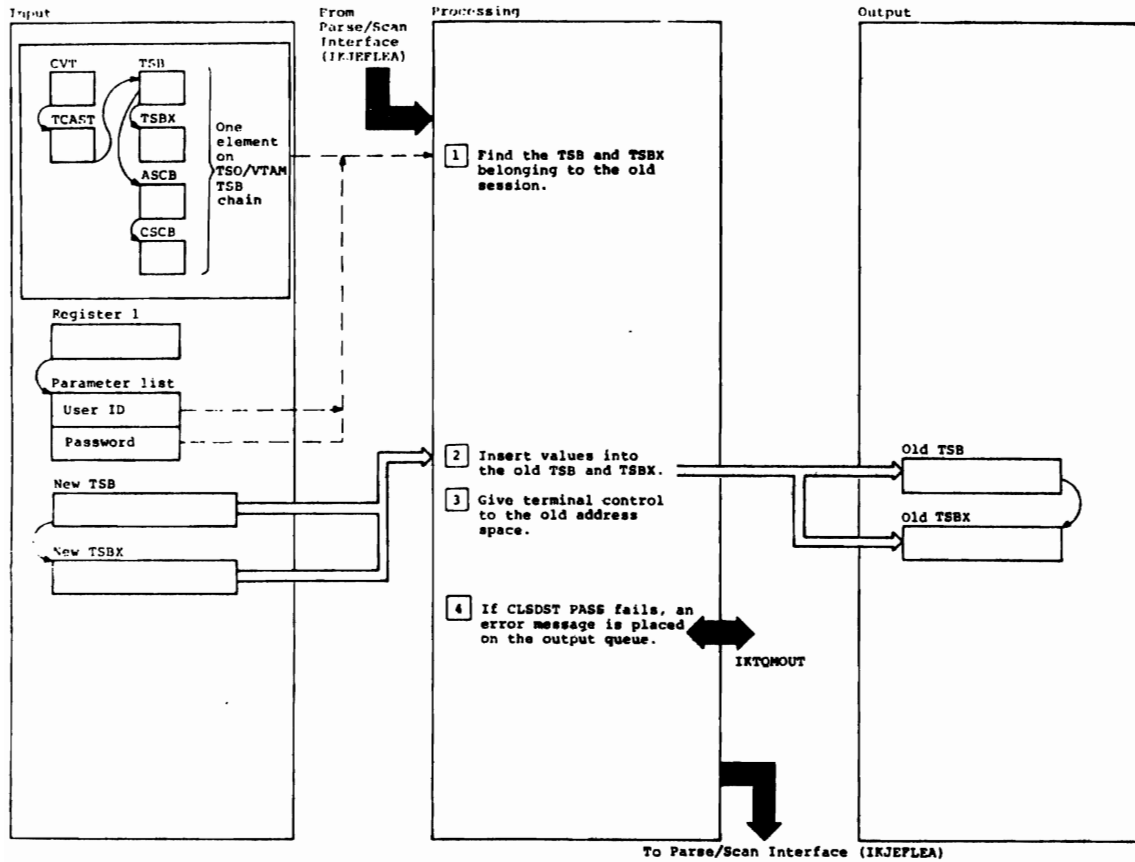
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine checks the status of the OPNDST RPL request issued in the VTIOC initialization routine (IKTXINIT).</p> <p>1 The protection key is set to 0 (MODESET macro) and CHECK RPL is issued.</p> <ul style="list-style-type: none"> After posting (POST macro) the ECB, the protection key is reset (MODESET macro) to specify the problem state. CHECK RPL is reissued as many times as specified in TWWARTR. If the OPNDST RPL did not complete successfully when the limit is reached, the terminal session is canceled. 	IKTRPLXT						
	IKTRPLXT						
	IKTRPLXT						
	IKTRPLXT						

MO 1.2 Extended Logon Routine (IKTXLOG)



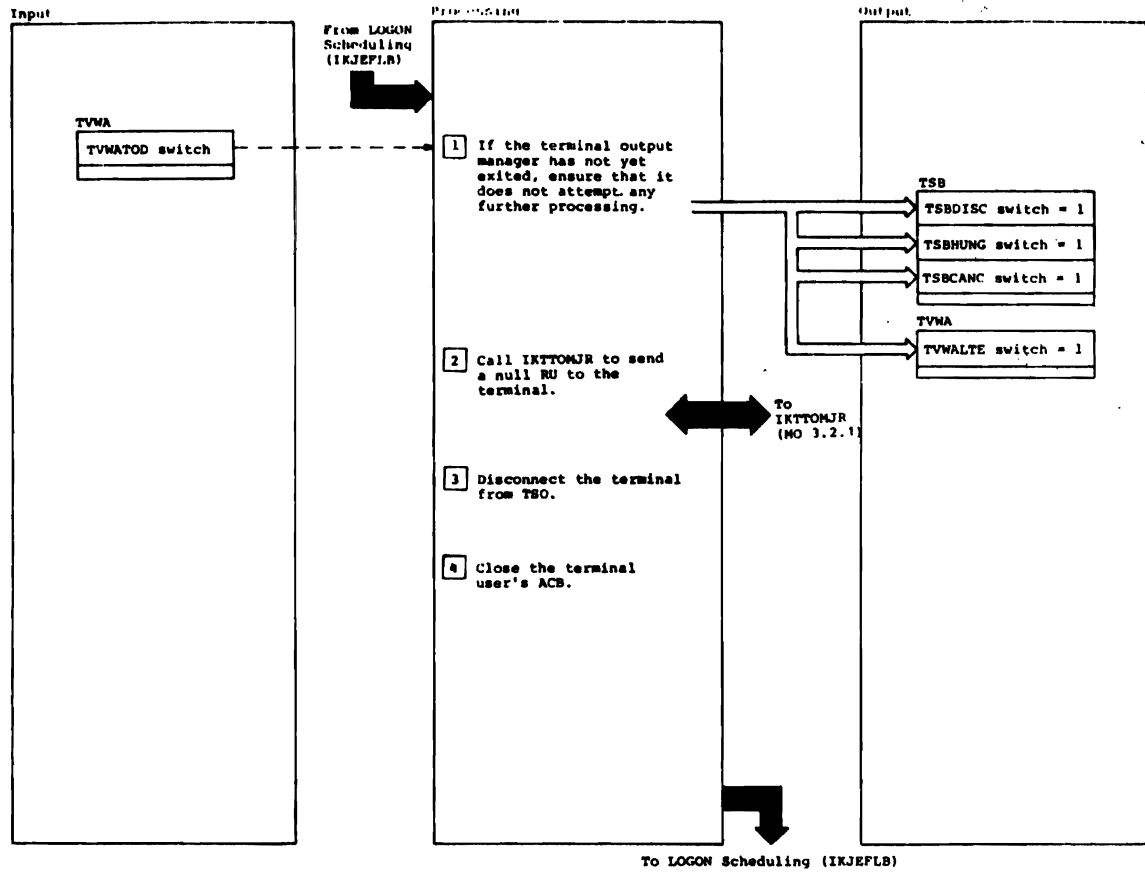
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine sets up the address of the logon command buffer as a parameter for the LOGON Scheduling routines.	IKTXLOG						
<p>1 The address of the logon buffer was placed into TVMAVST by IKTXINIT. The logon buffer is in subpool 1, key 8 storage. It contains the logon command entered by a TSO/VTAM terminal user.</p> <p>This routine is entered for an initial logon only, not for a re-logon or a logoff.</p>	IKTXLOG		1.1				

MO 1.3 Logon Reconnect Routine (IKTLOGR)



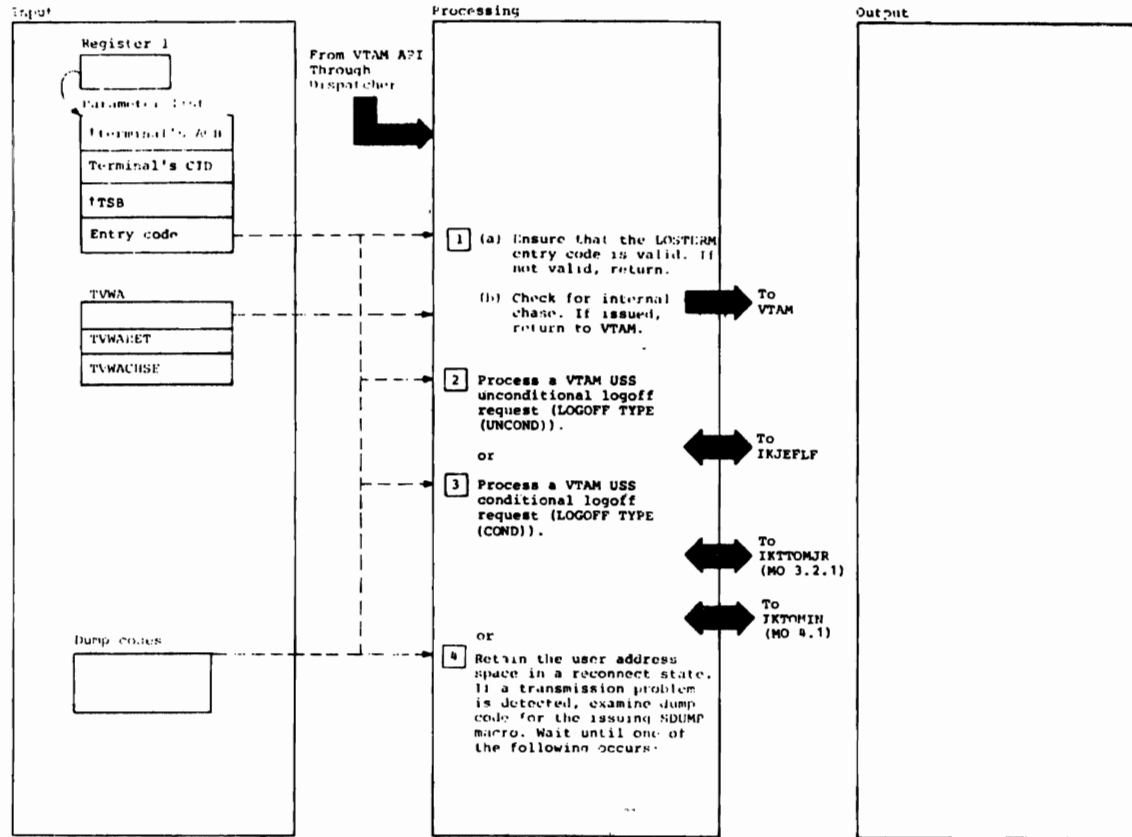
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine establishes reconnection of a TSO/VTAM terminal to the address space from which it was disconnected. It is invoked when a logon command specifies a Reconnect operand.	IKTLOGR						
1 The local and CMS locks are obtained, and the TSB chain is searched for a TSB with the reconnect bit on (TSBXWREC) and with a user ID and password that match those of the user requesting reconnection (pointed to by register:1).	IKTLOGR			4	IKTQMOUT		
2 Unaware that RECONNECT was specified, another LOGON routine created a new TSB (and TSBX). IKTLOGR now transfers the following TSB and TSBX values from the new TSB and TSBX to the existing ones: terminal symbolic name (TSBXSVM), ASCII flag (TSB3270, TSBDSPLY), terminal characteristics (TSBTRMID), and the Bind (TSBXBIND). Later, an RTM routine will free the unneeded new TSB and TSBX.	IKTLOGR						
3 The application ID of the old terminal is given to the new one (CLSDST PASS macro) and the old address space is posted.							
Return codes: 0 Successful reconnection 4 Invalid password 8 Invalid user ID C Unsuccessful reconnection							

MO 1.4 Extended Logoff Routine (IKTLOGFF)



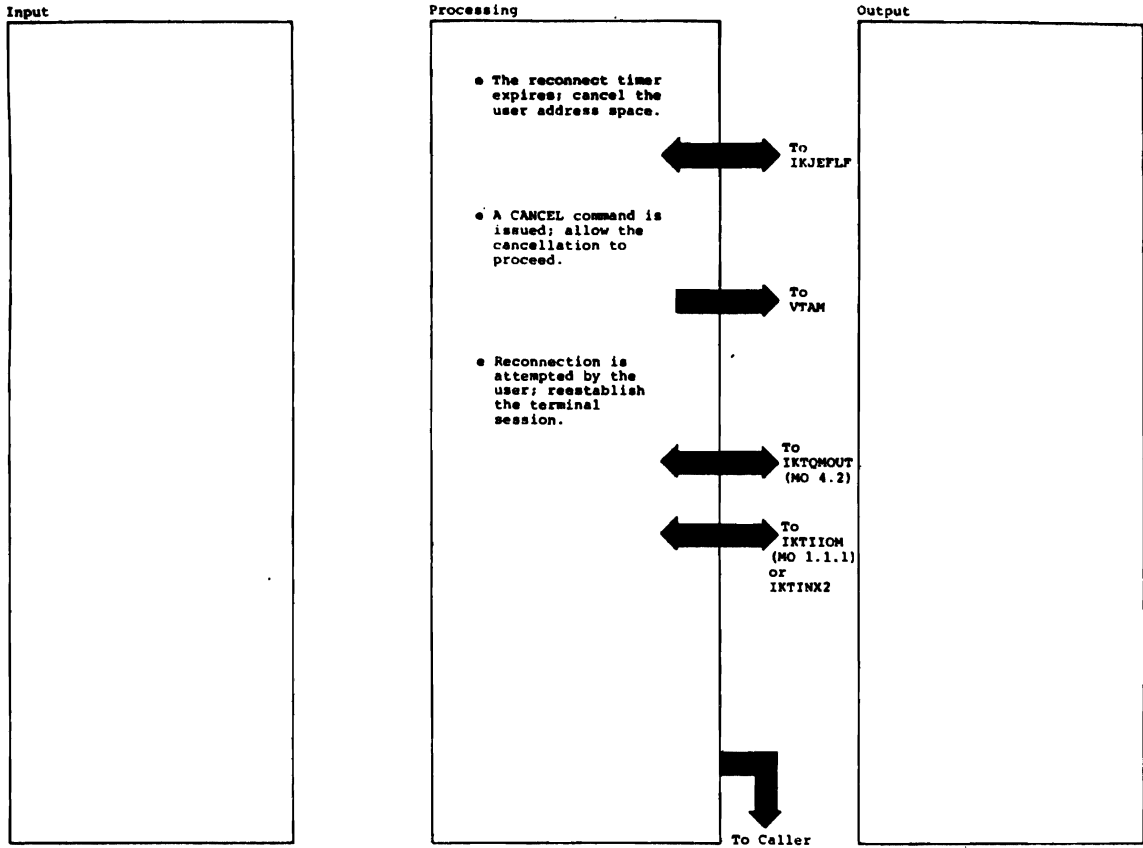
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine disconnects the user's terminal from TSO and closes the user's ACB.	IKTLOGFF						
1 The local lock is obtained, the TSB switches are set, TVWALTE is turned on if the TOM is scheduled (TVMATIS switch = 1), and the local lock is released.	IKTLOGFF						
3 The CLSDST macro disconnects the terminal from TSO.	IKTLOGFF	TERM					
4 SVC 20 is issued to close the ACB. If the close is unsuccessful, abend OAB is issued (ABEND macro) and reason code 203 (hex) is placed into register 15.	IKTLOGFF	TERM					

MO 1.5 (Part 1 of 2) LOSTERM Exit Routine (IKTLTERM)



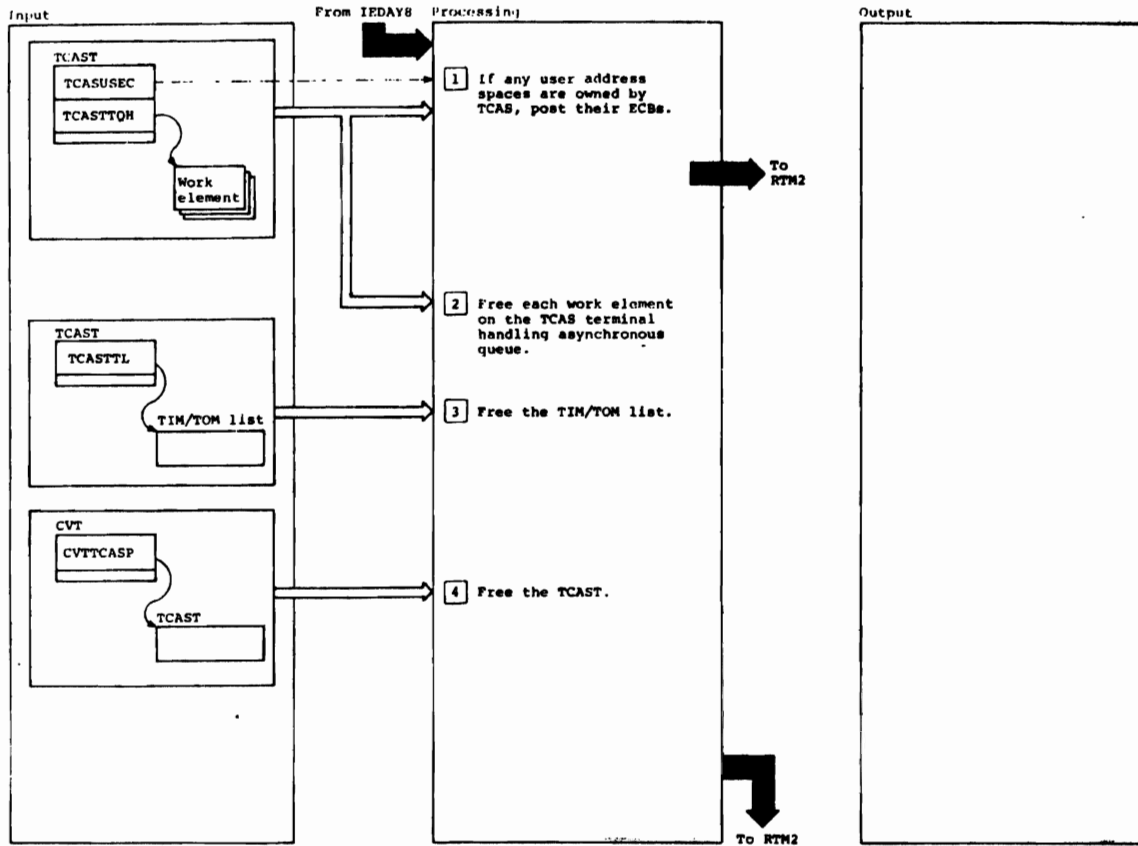
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref	
This routine processes VTAM USS logoff requests, and handles certain VTAM transmission errors for TSO/VTAM time sharing. It is scheduled as an IRB by VTAM.	IKTLTERM			<p>4 A request to retain the user address space in a reconnect state is indicated by entry codes 0, 1, 12, 24, 36, 40. The local lock is released and CLSDST RPL is issued to disconnect the terminal from TSO. (If the CLSDST macro is unsuccessful, ABEND OAB with reason code 0105 is issued to indicate a VTAM resource availability problem.) The local lock is released and a reconnect environment is established, if a time interval is specified (RECONLIM value in parmlib member TSOKEY00), by starting the timer (STIMER macro). (If a reconnect environment is not established, SIC routine IKJEFLF is branch-entered to cancel the user address space.) A WAIT macro is issued to wait for the timer to expire, a CANCEL command to be issued, or reconnect to be attempted.</p> <ul style="list-style-type: none"> • If the timer expires, SIC routine IKJEFLF is branch-entered to cancel the user address space. • If a CANCEL command is issued, control returns to VTAM to allow the cancellation to proceed. • If reconnection is attempted by the terminal user, reestablish the terminal session by doing the following: 	IKTLTERM	DECONN		
1 The LOSTERM entry codes are listed in VTAM Macro Language Reference, GC27-6995. If code 16, 28, or an unlisted code is received, message IKT1031 is issued and control returns to VTAM.	IKTLTERM	CONROUT						
2 An unconditional logoff request is indicated by entry code 20. The local lock is released and CLSDST RPL is issued to disconnect the terminal from TSO. (If the CLSDST macro is unsuccessful, ABEND OAB with reason code 0105 is issued to indicate a VTAM resource availability problem.) The local and CMS locks are obtained, and the system-initiated cancel (SIC) routine IKJEFLF is branch-entered to cancel the user address space.	IKTLTERM	LOGPUNCH					RECONNECT	
3 A conditional logoff request is indicated by entry code 32. IKTTOMJR is called to send a null RU to the terminal. Then the queue manager is asked to put the logoff command on the input queue. Parameters passed in IPARMS are: IPREQ - Code 1: Add element to bottom of queue IPTRMTYP - Terminal type IPBFSZ - Length of logon command IPAVBL - Message available indication	IKTLTERM	TERMAS					TERMAS	
	IKTLTERM	LOGCOND	4.1			TERMAS		
						ASSESS		

MO 1.5 (Part 2 of 2) LOSTERM Exit Routine (IKTLTERM)



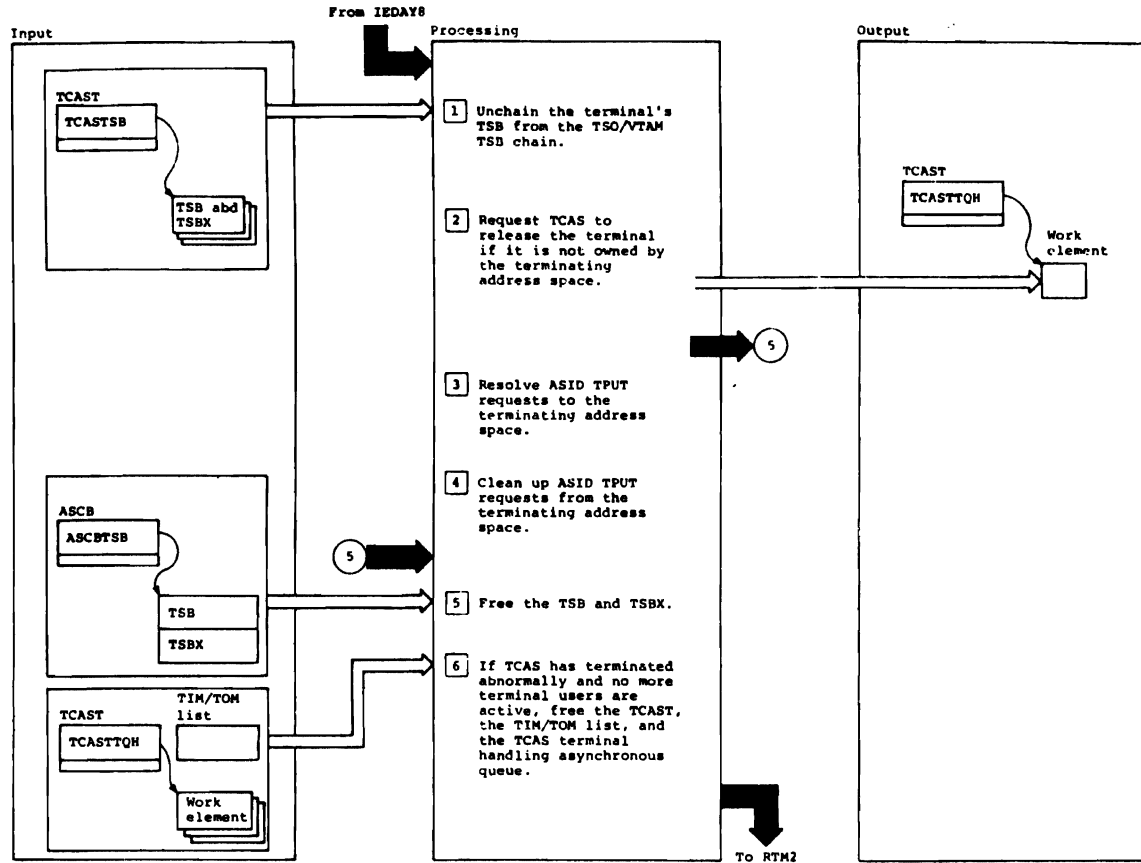
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
(a) Connect the terminal to TSO (OPNDST RPL); if unsuccessful, issue ABEND OAB with reason code 0105.							
(b) Request the output queue manager to restore the output queue to its condition prior to the transmission problem (line drop). The parameter passed in OPARMS is OPREQ (reset request, code 7).			4.2				
(c) Request the output queue manager to insert the logon reconnect command on top of the output queue. The parameters passed in OPARMS are OPREQ (code 8 - add element to top of queue) and OPBFSE (length of command).			4.2				
(d) Reinitialize the terminal input manager and terminal output manager by calling IBM-supplied routine IKTIOM or installation-written routine IKTINX2.			1.1.1				
(e) Schedule the appropriate terminal output manager to send a message to the terminal indicating successful reconnection (IKT00300I).			3.2 3.4				

MO 1.6 TCAS Termination Routine (IKTAY81)



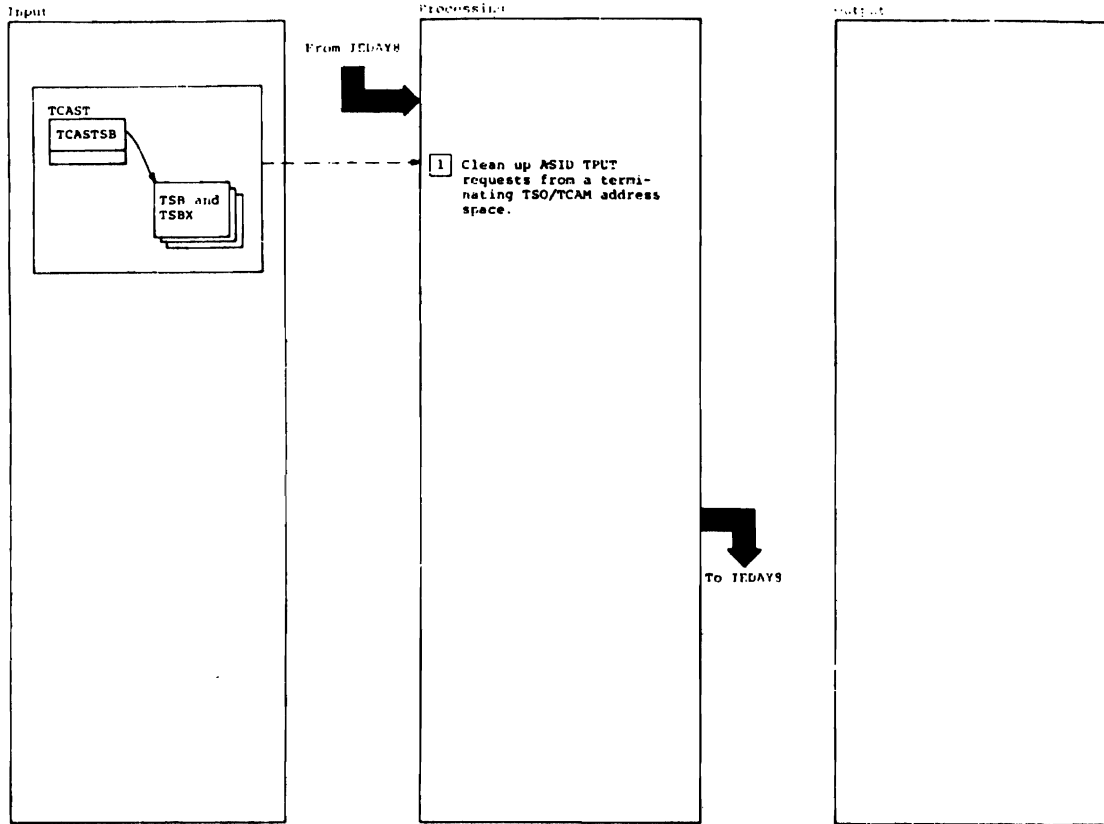
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine handles TCAS termination.	IKTAY81						
1 If TCAS terminates before it could transfer ownership of one or more terminals to user address spaces, the ECB of each unowned address space is posted, each work element on the TCAS terminal handling asynchronous queue is freed, and RTM2 receives control.	IKTAY81	POSTWECB FRNESTD					
2 FREEMAIN is issued to free each queue element.	IKTAY81	FRWESTD					
3 FREEMAIN is issued to free the TIM/TOM list, and the pointer in the TCAST is cleared. (The TIM/TOM list contains the addresses of the four TIM/TOM routines.)	IKTAY81	FRTTLIST					
4 FREEMAIN is issued to free the TCAST, and the pointer in the CVT is cleared.	IKTAY81	FRTTCAST					

MO 1.7 Terminal User Address Space Termination Routine (IKTAY82)



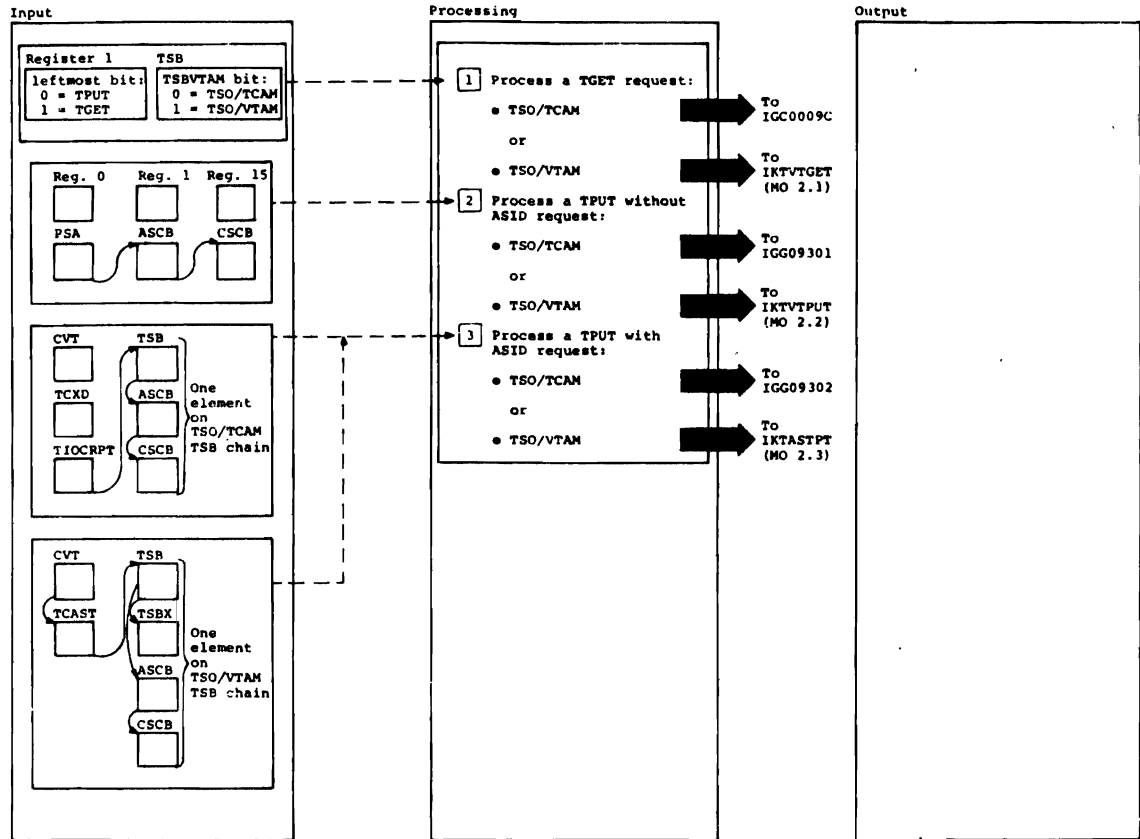
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine handles TSO/VTAM terminal user address space termination.	IKTAY82			terminating TSO/VTAM address space. When such a TSB is found, its ASID TPUT switches are cleared, IEDAYTPQ is called to inform TSO/TCAM not to expect any more ASID output, and tasks associated with this TSB are marked dispatchable (STATUS macro routine).	IEDAYTPQ		
1 The forward and backward chain pointers are manipulated to unchain the terminal's TSB and TSBX.	IKTAY82			The TSO/VTAM TSB chain is searched for TSBs that indicate an ASID TPUT request to a TSO/VTAM address space from the terminating TSO/VTAM address space. When such TSB is found, if the target address space is no longer active, outstanding posts are allowed to complete (SPOST macro), the cross-memory TPUT SRB is freed, and the TVCS is freed.	SRCHVTAM		
2 The terminal will not be owned by the terminating address space if an error occurred before TCAS could transfer control of the terminal to the address space. REQREL places a work element with function code X'22' on the terminal handling asynchronous queue of TCAS and posts TCAS.	IKTAY82	REQREL		When such a TSB is found and the target address space is still active, TVCSRCRC is set to indicate that the source address space is no longer active (leaving the target with the responsibility of freeing the cross-memory TPUT SRB and the TVCS).			
3 If the source address space has not relinquished control of the TVCS and cross-memory TPUT SRB, TVCSRCRC is set to indicate that the target (terminating address space) relinquishes control of the TVCS and cross-memory TPUT SRB, and the source is informed that the target has terminated by means of IKTAY8 posting (POST macro routine) the TVCS ECB with code X'14' (ASID TPUT complete). If the source address space has relinquished control of the TVCS and cross-memory TPUT SRB, the SRB is freed (FREEMAIN macro) and the TVCS is freed (FREEMAIN macro). Note that PURGEDQ is issued to ensure that the SRB is dequeued before it is freed; if it has been dispatched it is allowed to execute.	IKTAY82	FREESRB FREETVCS		The terminal input manager SRB and the terminal output manager SRB are freed, and the logon buffer is freed.			
4 The TSO/TCAM TSB chain is searched for TSBs that indicate an ASID TPUT request to a TSO/TCAM address space from the	IKTAY82	SRCHTCAM		5 FREEMAIN is issued to free the TSB and the TSBX. The TSB pointer in the ASCB is cleared.	IKTAY82	FREETSB	
				6 FREEMAIN is issued to free the TCAST, the TIM/TOM list, and each work element. The TCAST pointer in the CVT is cleared, and the TIM/TOM list pointer in the TCAST is cleared.	IKTAY82	FRITCAST FRFTLIST FRWFSTD	

MO 1.8 TSO/VTAM QTIP 29 Equivalent (IKTAY83)



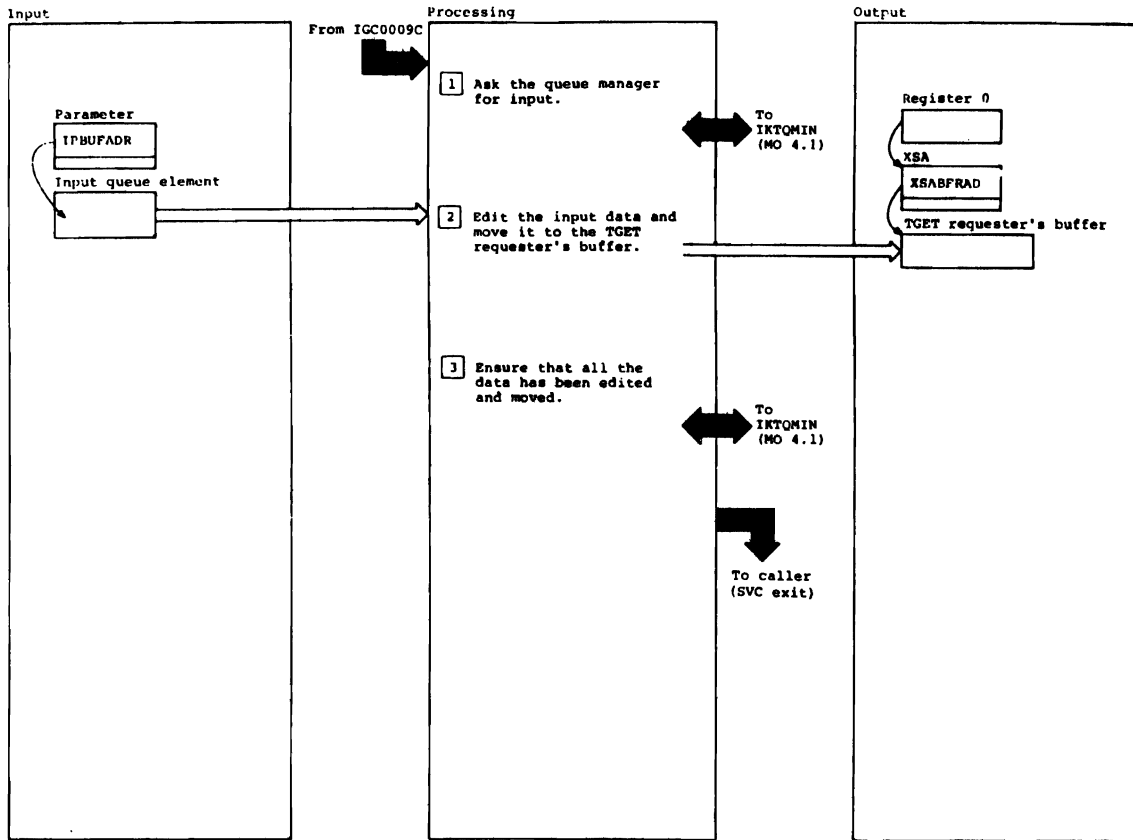
Notes	Program	Label	Ref	Notes	Program	Label	Ref
This routine cleans up ASID TPUT requests to a TSO/VTAM address space from a terminating TSO/TCAM address space.	IKTAY83						
1 The TSO/VTAM TSB chain is searched for TSBs that indicate an ASID TPUT request to a TSO/VTAM address space from a terminating TSO/TCAM address space. When a TSB is found, if the target address space is no longer active, outstanding posts are allowed to complete. (SPOST macro), the cross-memory TPUT SRB is freed, and the TVCS is freed. When a TSB is found and the target address space is still active, TVCSRRC is set to indicate that the source address space is no longer active.	IKTAY83	SRCHVTAM					
		FREESRB					
		FREEVCBS					

MO 2.0 Overview of TGET/TPUT Routines



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>The TGET and TPUT routines handle the movement of data between TSO and a terminal. They are used by the TSO command processors, the TSO terminal monitor program, the TSO I/O service routines, and user-written command processors. They receive control after SVC 93, a TGET macro instruction, or a TPUT macro instruction is issued. (Note that specifying TGET or TPUT invokes SVC 93.)</p> <p>The function described in this diagram is contained in object module IGC0009C, which is the first TGET/TPUT module to receive control in load module IGC0009C. It is entered from the SVC second level interrupt handler, and it transfers control to the appropriate TIOC and VTIOC TGET or TPUT routine. IGC0009C is described here from VTIOC's viewpoint; it is described from TIOC's viewpoint in OS/VS2 TCAM Logic, SY30-2040.</p> <p>2 If an address space ID (Reg. 0) or user ID (Regs. 1, 15) was not specified, or if the specified address space ID (Reg. 0) matches the active address space ID (ASCBASID), or if the specified user ID (Reg. 15) matches the active user ID (CHKEY), TPUT without ASID was issued.</p>				<p>3 The appropriate TSB chain is searched to find the target of the TPUT ASID. The search looks for a match for the address space ID (ASCBASID) or the user ID (CHKEY). When found, the appropriate ESTAE routine is set up (ESTAE macro) and the TPUT routine is called. If a match is not found, code 14 (hex) is returned in register 15.</p>			

MO 2.1 (Part 1 of 2) TGET Routine (IKTVTGET)



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine handles TSO/VTAM TGET requests. It takes data from the input queue, edits it, and places it in the TGET requester's buffer.</p> <p>The local lock is held on entry to IKTVTGET.</p> <p>1 If the terminal session is no longer active or if ATTN was pressed, control returns immediately to IGC0009C. Otherwise, the queue manager is called to get an element from the input queue. The following parameter is passed to the queue manager in IPARMS:</p> <p>IPREQ - Code 2: Obtain address of next element.</p> <p>If the queue manager is unable to obtain the data (IPRC ≠ 0), schedule the terminal output manager to unlock a 3270 keyboard, suspend the TGET requester (by calling the STATUS macro routine) until data is available or ATTN is pressed if TGET WAIT was specified, inform the SRM of the wait (SYSEVENT macro), and obtain the local lock upon being redispatched.</p>	IKTVTGET			<p>2 VERIFY determines if the TGET requester's buffer address is in the TGET requester's key. IKTVTGET uses one or more of the following routines for editing:</p> <p>IKTGETXT - Edits data from terminals other than 3270, 3767, 3770</p> <p>IKTIDSK4 - Can be written by an installation to replace EDIT3270</p> <p>EDIT3270 - Edits and moves data from 3270 terminals</p> <p>IKTRTX4 - Can be written by an installation to replace EDIT3767</p> <p>EDIT3767 - Edits and moves data from 3767 and 3770 terminals</p> <p>The edit routines perform device independent editing. They remove invalid characters, remove 3270 control characters, pad with blanks if the EDIT operand was specified, and move the data from the input queue to the TGET requester's buffer.</p>	IKTVTGET	VERIFY	
	IKTVTGET	TERMSTAT	4.1				
			3.2 3.4			EDIT3270 EDIT3767	

MO 2.1 (Part 2 of 2) TGET Routine (IKTVTGET)

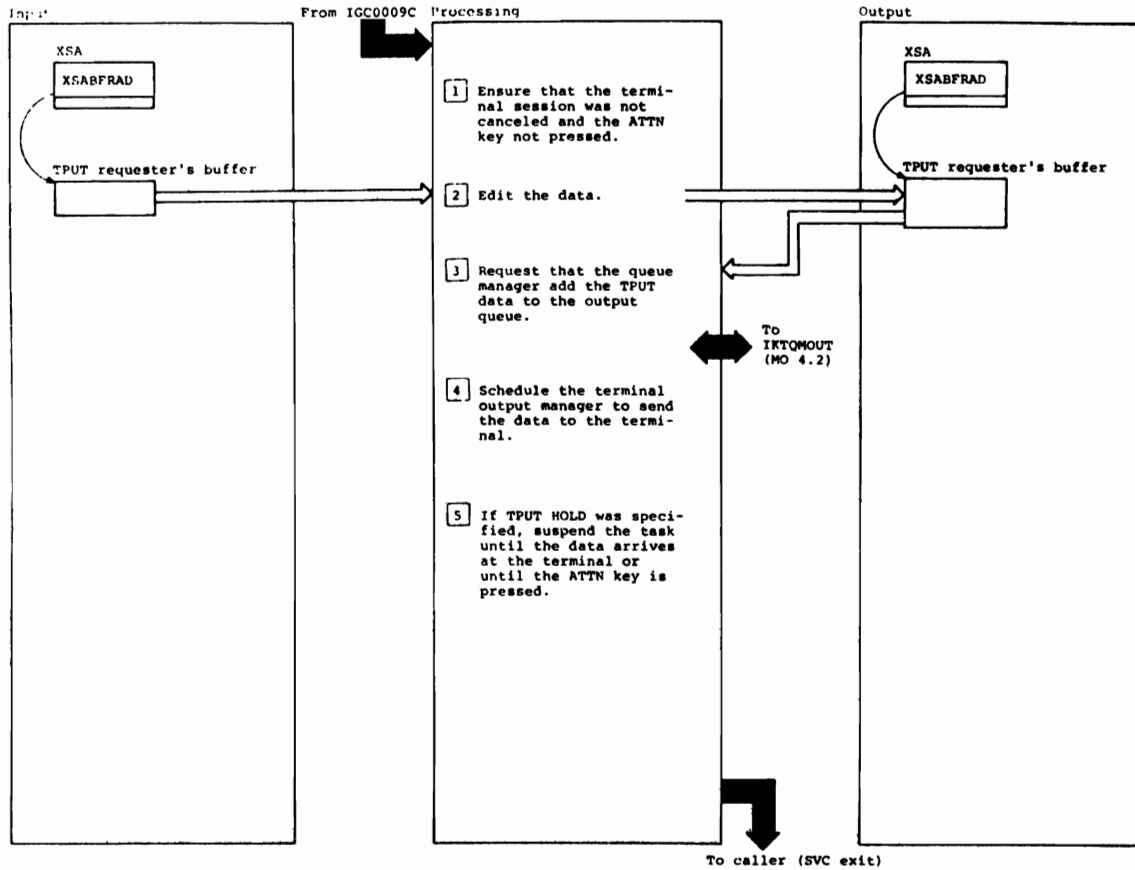
Input

Processing

Output

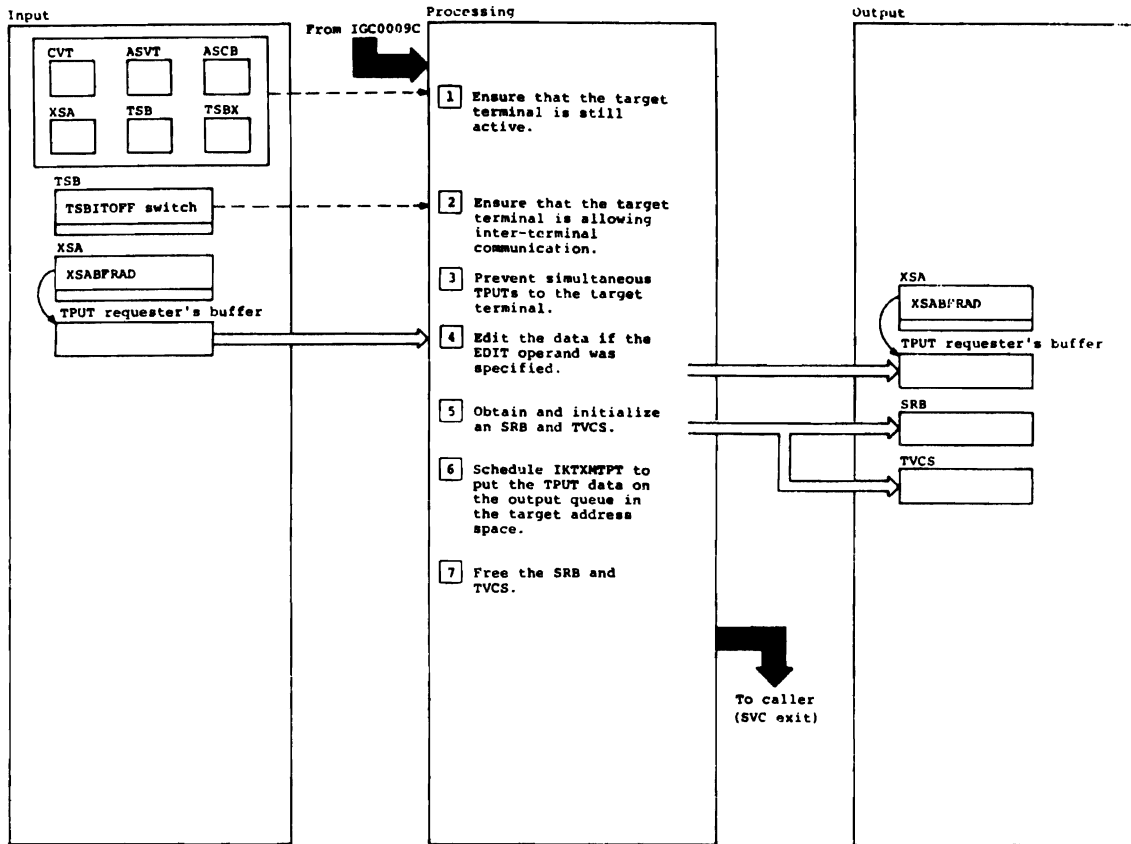
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>3 The queue manager is called to unchain the edited element from the input queue and to free associated storage. The following parameter is passed in IPARMS:</p> <p>IPREQ - Code 3: Remove element from queue. SRM is informed (SYSEVENT macro) of TGET completion. However, if the TGET requester's buffer is smaller than input line size, call the queue manager to update the pointer to the first un-moved character. The following parameter is passed in IPARMS:</p> <p>IPREQ - Code 4: Update element.</p> <p>Return codes (in hex):</p> <p>0 Successful 4 No input available and TPUT NOWAIT was specified 8 ATTN key was pressed; message was not received C TGET requester's buffer too small to hold input line 14 Terminal could not be reached</p>	IKTVTGET						

MO 2.2 TPUT without ASID Routine (IKTVTPUT)



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine handles TSO/VTAM TPUT without ASID requests. It takes data from the TPUT requester's buffer, edits it, and places it on the output queue.	IKTVTPUT			5 The SRM is notified (SYSEVENT macro) that IKTVTPUT is suspended. When the data arrives at the terminal, the SRM is notified (SYSEVENT macro) that the TPUT is complete.	IKTVTPUT	WAITCHK	
1 TERMSTAT checks the terminal's status and WIFOWAIT determines if another task owns the TSB. If another task does own the TSB (is processing a TPUT), WAITCHK suspends IKTVTPUT (STATUS macro routine) until the TSB is available.	IKTVTPUT	TERMSTAT WIFOWAIT WAITCHK		Any tasks waiting on IKTVTPUT are marked dispatchable (STATUS macro routine), and the local lock is released.	IKTVTPUT	RESWAITS	
2 Device independent editing is performed. It consists of removing trailing blanks.	IKTVTPUT	DATAEDIT		Return codes (in hex):			
3 The following parameters are passed to the queue manager: OPBFSZ - Size of TPUT data OPFLAGS - Output options from XSAOPTNS OPREQ - Code 1: Add element to bottom; or, if BREAKIN specified, Code 8: Add element to top. If the data cannot be added to the queue because high buffer threshold is reached, IKTVTPUT is suspended (STATUS macro routine) unless TPUT NOWAIT was specified, until the queue manager is able to add the data.	IKTVTPUT	TRYTOQUE	4.2	0 Successful			
4 The appropriate terminal output manager is scheduled (SCHEDULE SRB) if not already scheduled (TVWATIS) and if available to be scheduled (TVWATAS).	IKTVTPUT	SCHEDTOM	3.2 3.4	4 No output buffer available and TPUT NOWAIT specified			
				8 ATTN key was pressed; message was not sent			
				14 Terminal could not be reached.			

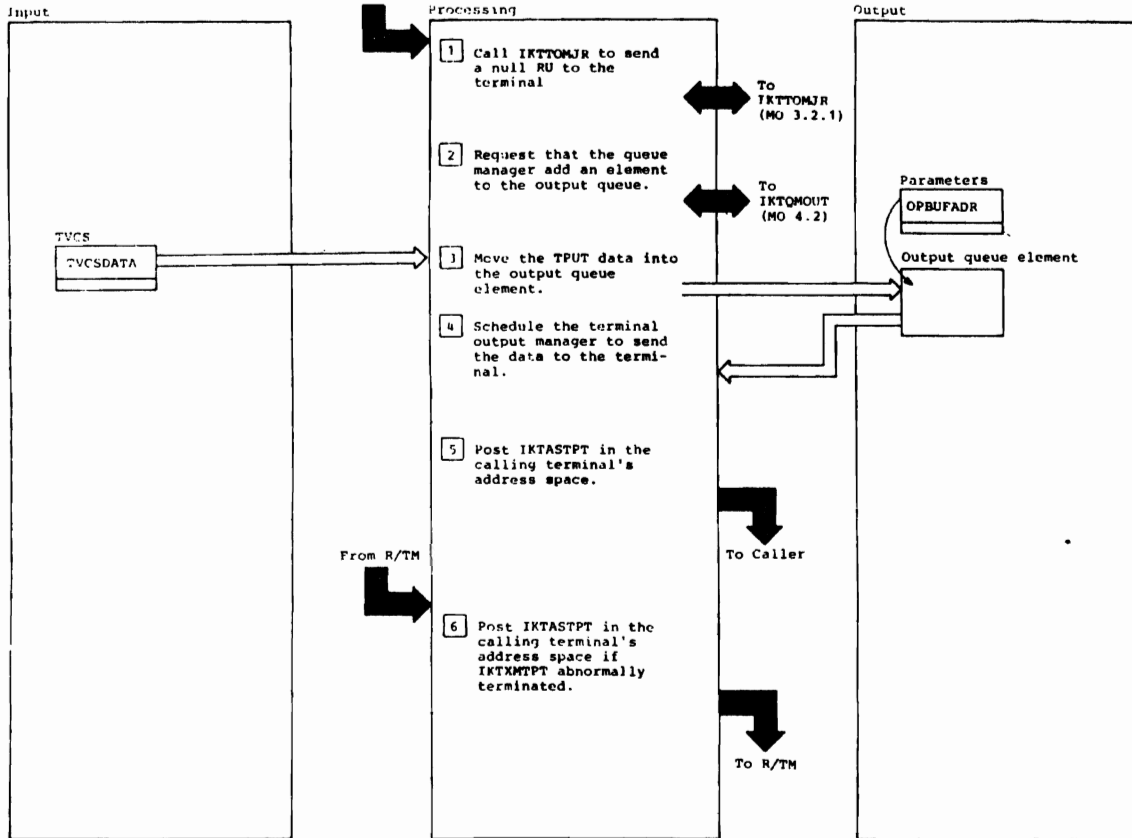
MO 2.3 Calling Address Space TPUT with ASID Routine (IKTASTPT)



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine handles TSO/VTAM TPUT with ASID requests for the calling address space. It takes data from the TPUT requester's buffer, edits it, and places it in the TVCS for handling by IKTXMPT.</p> <p>The local lock is held on entry to IKTASTPT.</p>	IKTASTPT			<p>4 DATAEDIT scans the calling terminal's data and performs device independent editing, that is, it removes trailing blanks.</p>	IKTASTPT	DATAEDIT	
<p>1 TERMSTAT determines if TSO/VTAM is still running (CVTTCASP), if the target terminal is still in session (ASVTAVAL, ASCBTSE, XSAASID), and if the target terminal had its session canceled (TSBANC), was disconnected (TSBDISC), or was not completely initialized (TSBACTV). Register 15 is set to hex 14 if any of these conditions is negative.</p>	IKTASTPT	TERMSTAT		<p>5 The SALLOC locks are obtained before storage is allocated (GETMAIN macros) in the CSA for an SRB (for scheduling IKTXMPT), and TVCS. The TVCS holds the data until IKTXMPT in the target address space can move it to the output queue. TERMSTAT ensures that the target terminal did not go down while the CMS lock was released. INITTVCS initializes the TVCS, VERIFY ensures that the caller's area is in the caller's key, and INITSRB initializes the SRB.</p>	IKTASTPT	TERMSTAT	
<p>Whenever the TSB of the target terminal is referred to, the CMS lock is held to prevent the target address space from freeing the TSB should the target terminate.</p>	IKTASTPT			<p>6 IKTXMPT is scheduled (SCHEDULE SRB) in the target address space. IKTASTPT waits (WAIT macro) until posted (POST macro) by IKTXMPT. If IKTXMPT abends, IKTASTPT reschedules it once.</p>	IKTASTPT		2.3.1
<p>2 If the target terminal specified no inter-terminal communication, (TSBITOFF), the TPUT is not honored unless the calling terminal is appropriately authorized (TESTAUTH macro) and specified the HIGHP operand.</p>	IKTASTPT			<p>7 TERMSTAT ensures that the target terminal did not go down while the CMS lock was released.</p>	IKTASTPT	TERMSTAT	
<p>3 The ENQ macro is issued to enqueue on the target terminal's address space ID. The ENQ qname is SYSZIGGI; the rname is the ID of the target address space.</p>	IKTASTPT			<p>Return codes (in hex):</p> <ul style="list-style-type: none"> 0 Successful 4 Buffer not available or ENQ resources not available, and TPUT NOWAIT specified C PROFILE NOINTERCOM specified by terminal operator 14 Terminal could not be reached 			

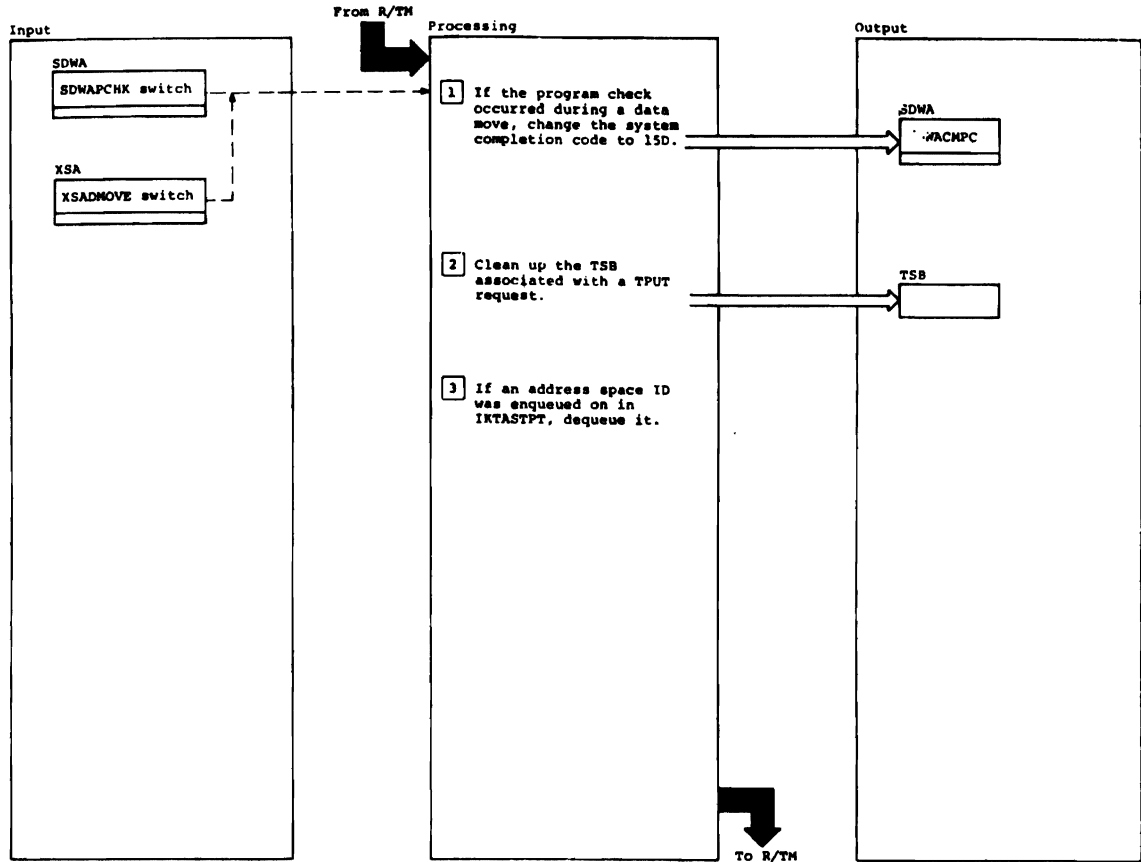
MO 2.3.1 Target Address Space TPUT with ASID Routine (IKTXMTPT)

From IKTASTPT (MO 2.3) through Dispatcher



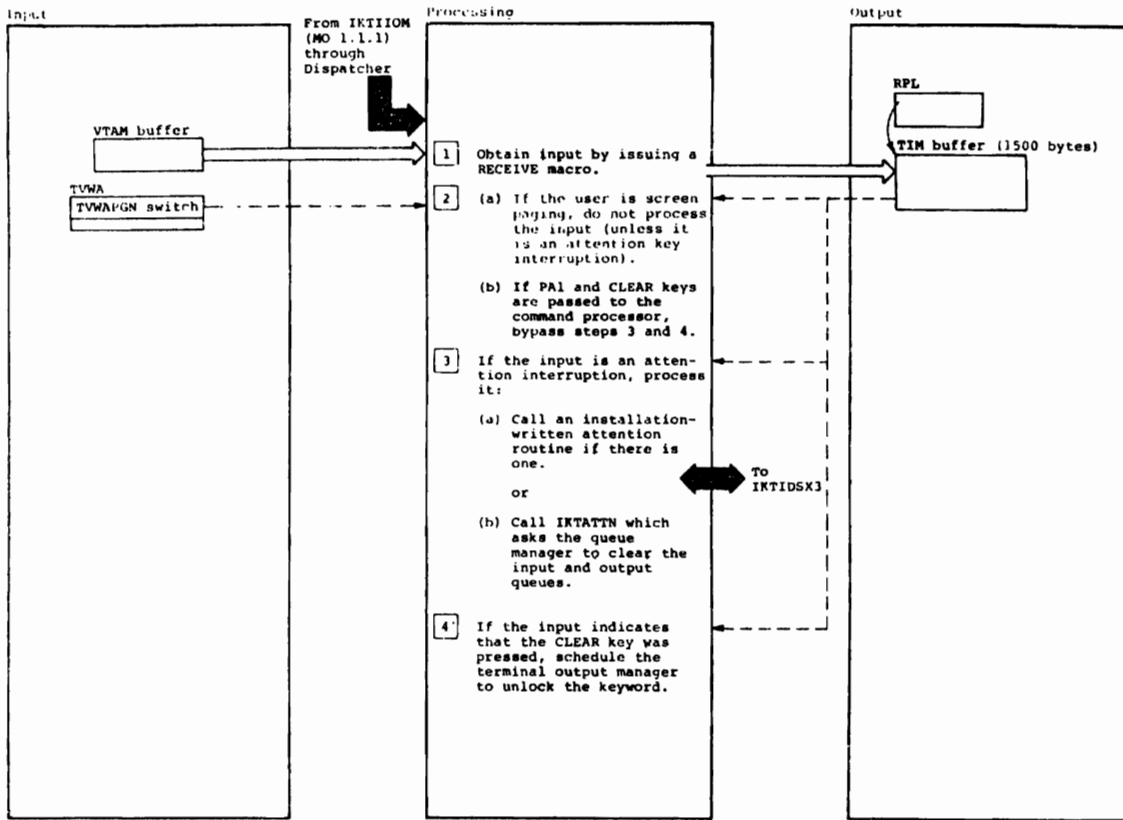
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine handles TSO/VTAM TPUT with ASID requests for the target address space. It takes data from the TVCS and places it on the output queue. It is scheduled by IKTASTPT as an SRB.	IKTXMTPT						
<p>2 Set up FRR IKTXMFRR by issuing the SETFRR macro, obtain the local lock, ensure that TSO/VTAM and the terminal are still active, and pass the following parameters to the queue manager in OPARMS:</p> <p>OPREQ - Code 8: Add element to top</p> <p>OPBFSZ - Size of TPUT data</p> <p>OPOPTNS - Output options from TVCSOPTN</p> <p>OPASID - TPUT ASID specified</p> <p>If the queue manager is unsuccessful (OPRC ≠ 0), TSBTJBF is set so that the queue manager reschedules IKTXMTPT when buffer space is available. However, if NOWAIT is specified (TVCSNOWT), IKTASTPT is posted with an indication that the TPUT could not be done. (POST code = 4.)</p>	IKTXMTPT		4.2				
4 The appropriate terminal output manager is scheduled (SCHEDULE SRB) if not already scheduled (TVWATIS) and if available for scheduling (TVWATAS).	IKTXMTPT		3.2 3.4				
5 IKTASTPT in the calling terminal's address space is posted and SRM is notified (SYSEVENT macro).	IKTXMTPT	POSTEM					
6 The calling terminal's address space is posted, indicating that the FRR was entered.	IKTXMFRR						

MO 2.4 ESTAE Routine for SVC 93 (IKT93EST)



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine is entered when one of the VTIOC TGET/TPUT routines (IKTGTGET, IKTGTPUT, or IKTASTPT) abends, or, through percolation, when the queue manager abends. It changes the abend code, cleans up the TSB, and dequeues the target address space.	IKT93EST						
1 If the buffer address specified on the TGET or TPUT macro is not in the caller's key, the program check that occurs (usually 0C4) is changed to system completion code 15D. The TGET or TPUT is not retried; recovery is left to the next higher ESTAE routine.	IKT93EST						
2 There is no cleanup required with a TGET request. For a TPUT request tasks waiting for the TPUT processing to complete are dispatched (STATUS macro routine), and TSB fields TSBOWIP and TSBCTCB are cleared. If the TPUT request specified ASID, and the target address space relinquished control or IKTXTPT was not scheduled, clear TSB fields TSBWJID and TSBWTCB and TSBX field TSBXCSAP if the target terminal is still connected, and free (FREEMAIN macro routine) the SRB (used for IKTXTPT) and the TVCS after ensuring that the post from the target address space has completed.	IKT93EST	TSBCLEAN					
3 The enqueue in IKTASTPT prevents simultaneous TPUTs to the target terminal. The DEQ qname is SYBZIGGI; the rname is the ID of the target address space.	IKT93EST	TERMSTAT FREESTOR					

MO 3.1 (Part 1 of 2) 3270 LU0 Terminal Input Manager (IKTIMIDS)



Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
This routine obtains IBM 3270 terminal input from a VTAM buffer and places it on the input queue for processing by the 1GET routine.	IKTIMIDS						
1 The RPL for the RECEIVE macro is initialized at first entry to IKTIMIDS. Upon initial entry and after each RECEIVE is completed, issue SETFRR to establish recovery function. Up to two RECEIVE macros are issued to obtain all the input from the VTAM buffer. If the RECEIVE is unsuccessful, retry is attempted. If retry is unsuccessful three times, ABEND OAB is issued. If the LOSTERM exit routine was entered, IKTIMIDS exits to INTEXIT.	IKTIMIDS	PRIMPROC INITFRR					
		REC					
		VTAMERR EXITFRR					
		FINPROC					
	INTEXIT						
After the input is processed, RECEIVE is issued for more input. In this way a RECEIVE request is always outstanding.							
2 If the user is paging, only an attention interruption is recognized. The line count is reset to one (TVWALNCT), the indicator is turned on to format the screen (TVWAFMSC) and the terminal output manager is scheduled if requested.	INTIMIDS	DATARTN					
		SCHTOM					
				3 If the user entered an attention interruption, an installation-written attention routine is called, if there is one, and the IBM-supplied routine is bypassed.	IKTIMIDS	DATARTN	
				(b) Code 5 is passed in IPARMS. IKTATTN is called to remove all elements from the input queue. Code 5 is passed in OPARMS to ask the output queue manager to remove all elements from the output queue.	IKTIMIDS	ATTN	4.1
							4.2
				4 If the user pressed the CLEAR key, the line count is reset to one (TVWALNCT), the indicator is turned on to format the screen (TVWAFMSC), and the terminal output manager is scheduled to unlock the keyword.	IKTIMIDS	DATARTN	
						SCHTOM	

MO 3.1 (Part 2 of 2) 3270 LU0 Terminal Input Manager (IKTIMIDS)

Input

Processing

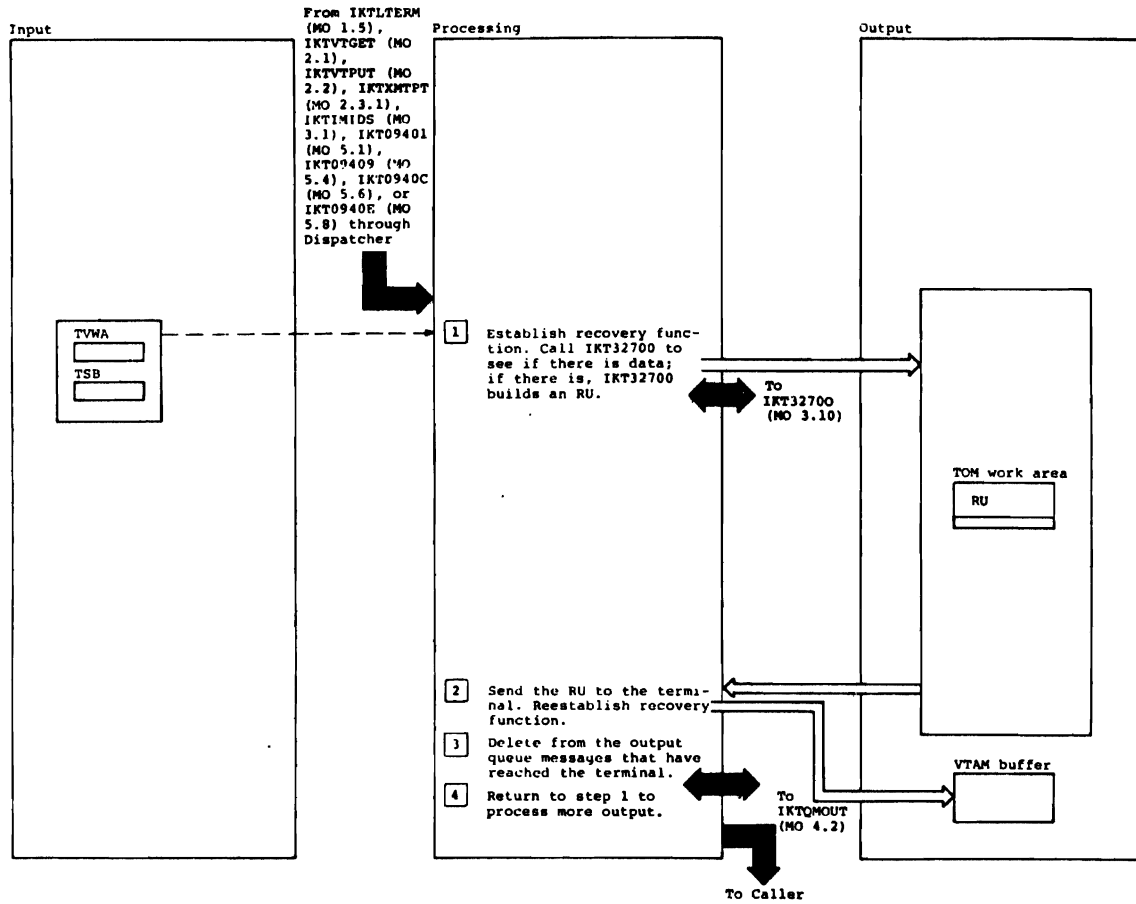
5 If the input is data, process it.

6 If there is an error, handle it.

Output

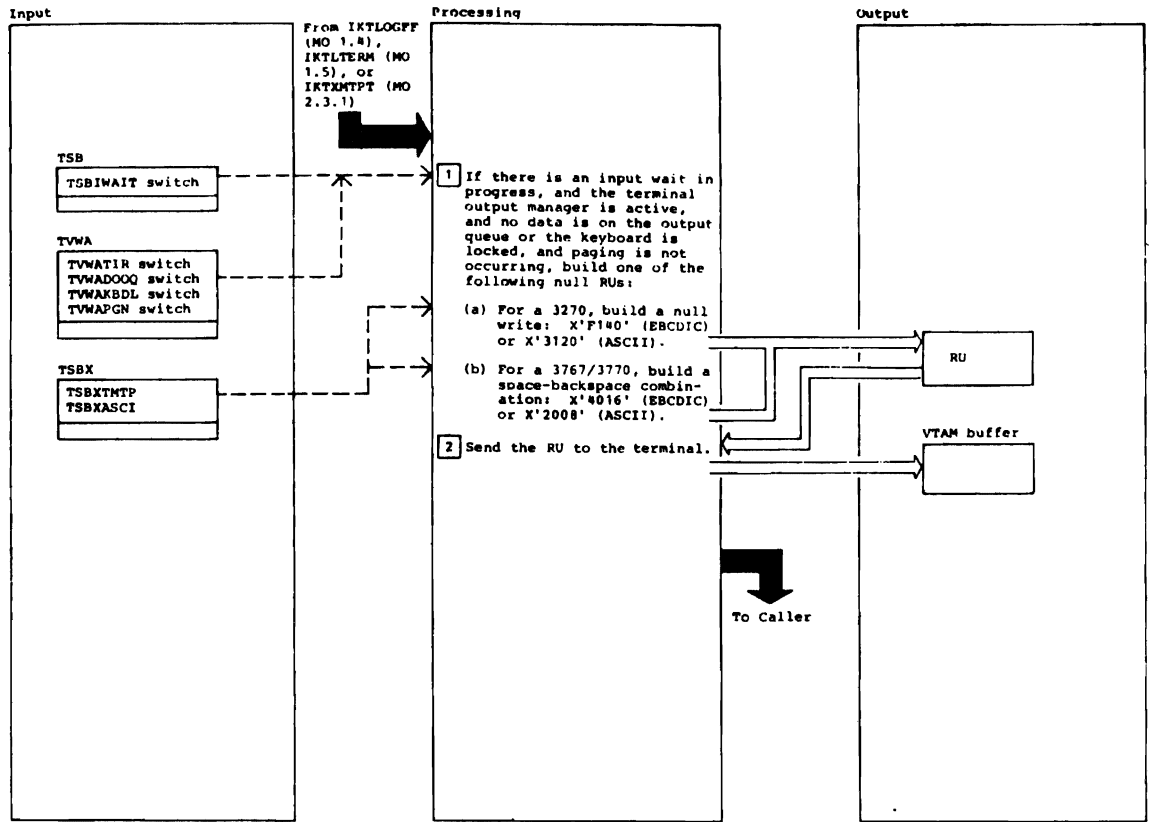
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
5 Data is processed and the loop resumes processing at step 1.	IKT3270I						
6 Up to three retries are attempted, then IKTIMIDS either calls IKTEXIT or abnormally terminates.	IKTEXIT	EXITFRR					

M.O 3.2 3270 LU0 Terminal Output Manager (IKTOMIDS)



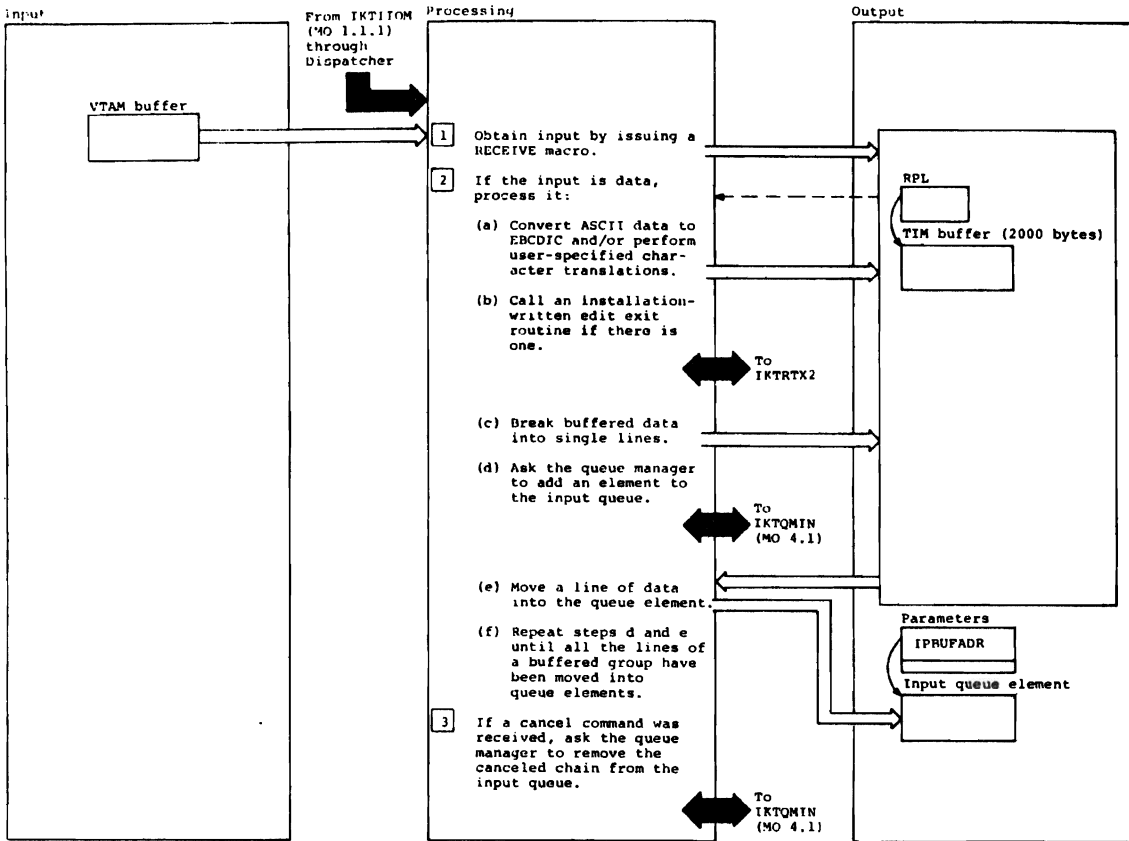
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
1	IKT32700	INITFRR					
2 A SEND macro is issued. If an error occurs attempting to send, the macro may be reissued three times before termination is required.	IKTOMIDS	SENDRU CKVTAMRC INITFRR EXITFRR					
3 Code 3 is passed in OPARMS to the output queue manager if positive function management end (FME) was received.	IKTOMIDS		4.2				
4 If the terminal input manager is no longer active, permanent cleanup is required: the TOM work area and SRB are freed (FREEMAIN macro) and control returns to the dispatcher. Otherwise, processing returns to step 1 until there are no more RUs to send.	IKTOMIDS	ERREXIT					

MO 3.2.1 Terminal Output Manager Null RU Routine (IKTTOMJR)



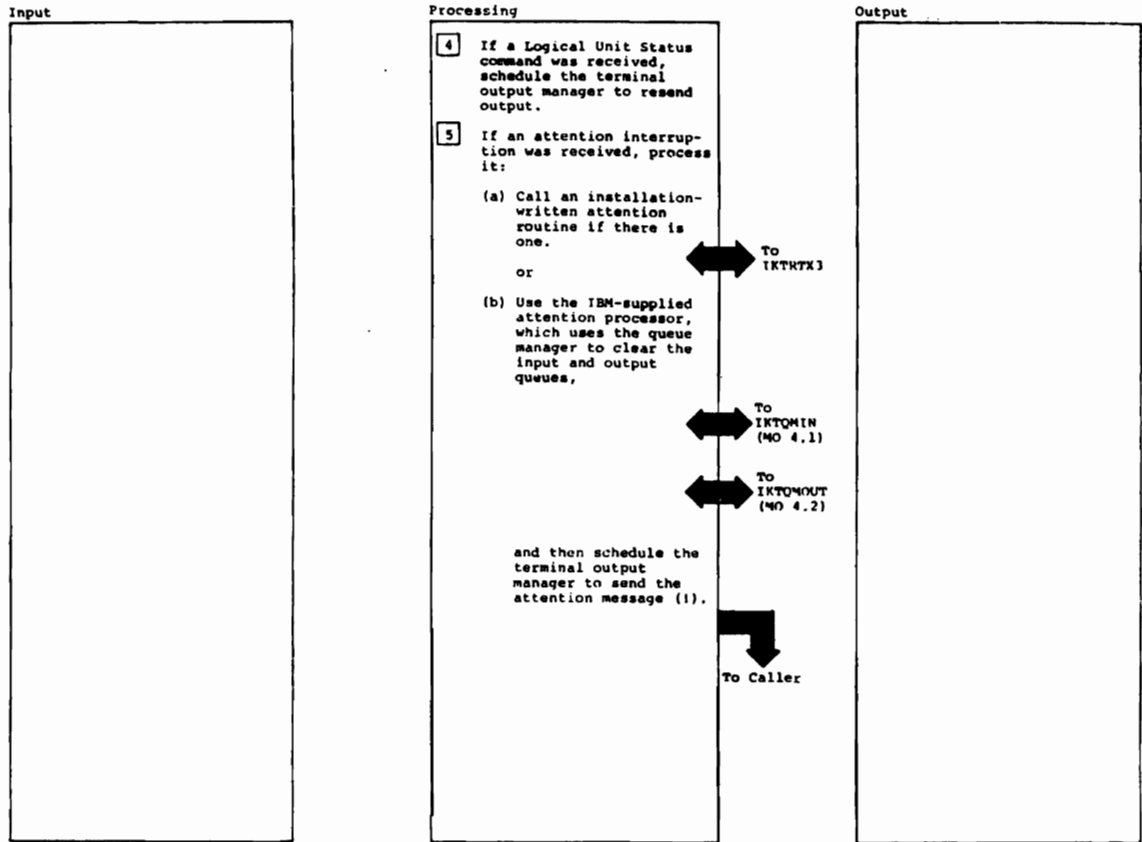
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine sends (SEND macro) a null RU when one of the terminal output managers (IKTTOMIDS or IKTTOMRT) cannot send an RU because it is suspended. This happens after a TSO/VTAM user has been canceled via the system-initiated cancellation routine, a conditional logoff has occurred, or a cross-memory TPUT has been sent to a 3767 or 3770 terminal. Sending a null RU causes VTAM to be dispatched, thus notifying VTAM that the user's address space is swapped in again so that the terminal output manager can be dispatched.</p>	IKTTOMJR						
<p>1 A GETMAIN macro is issued to obtain storage for the null RU. The storage is freed (FREEMAIN macro) after the RU is sent.</p>	IKTTOMJR						

MO 3.3 (Part 1 of 2) 3767/3770 Terminal Input Manager (IKTTIMRT)



Notes	Routine	Lab-1	Ref	Notes	Routine	Label	Ref
This routine obtains IBM 3767 and IBM 3770 terminal input from a VTAM buffer and places it on the input queue for processing by the TGET routine.	IKTTIMRT			(d) Code 1 is passed in IPARMS to ask the queue manager to add an element to the bottom of the input queue.		ADDELRTM	4.1
1 The RPL for the RECEIVE macro is initialized at first entry to IKTTIMRT. The recovery function is established and RECEIVE is issued to obtain input. The recovery function is reestablished. If the RECEIVE is successful, the input is processed. If the RECEIVE is unsuccessful, retry is attempted. If unsuccessful three times, ABEND OAB 18 issued. If the LOS-TERM exit routine was entered, IKTTIMRT exits. After the input is processed, RECEIVE is issued for more input. In this way a RECEIVE request is always outstanding.	IKTTIMRT	PRIMPROC INITFRR REC INITFRR VTAMERR EXITFRR FINPROC		3 A cancel command is received when the CANCEL key is pressed. Code 6 is passed in IPARMS to ask the queue manager to remove from the queue elements that are not marked available for TGET.	IKTTIMRT	CANCELR	4.1
2	IKTTIMRT	DATARTN DATA2 DATA2 DATA2					
(a) If the data is in ASCII code, a table is used to convert it to EBCDIC. If requested by the user (TERMINAL command), character translation is performed.							
(b) IKTRTX2 may be used to do scanning and editing in addition to that provided by IKTTIMRT.							
(c) Buffered data is scanned for Return indicators so that the data can be broken up into single lines.							

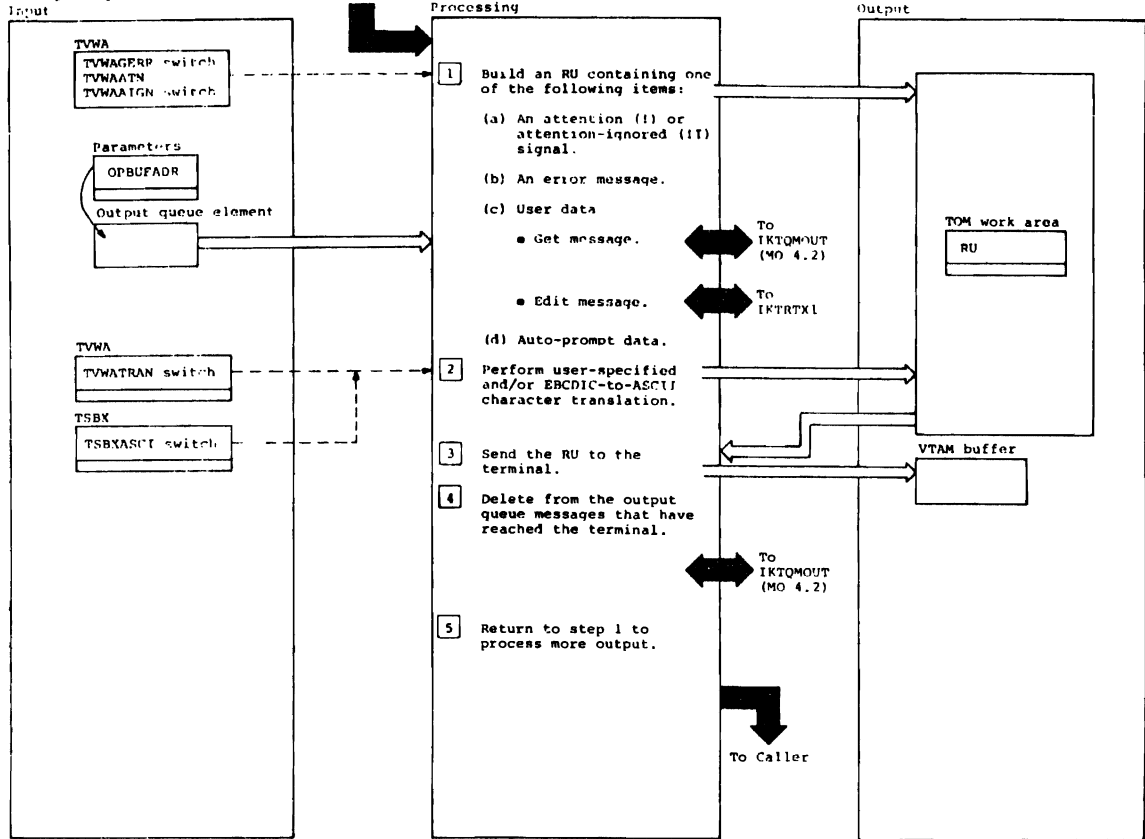
MO 3.3 (Part 2 of 2) 3767/3770 Terminal Input Manager (IKTTIMRT)



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>4 A Logical Unit Status command is received when the CANCEL key is pressed between chains after a negative function management end (FNE), when buffer contention is indicated to the host, or when an intervention required condition is satisfied.</p>	IKTTIMRT	LUSTAT					
<p>5 If the user entered an attention interruption, an installation-written attention routine is called, if there is one, and the IBM-supplied routine is bypassed.</p> <p>(b) Code 5 is passed in IPARMS to ask the input queue manager to remove all elements from the input queue. Code 5 is passed in OPARMS to ask the output queue manager to remove all elements from the output queue.</p>	IKTTIMRT	REC					
		ATTN	4.1				
			4.2				

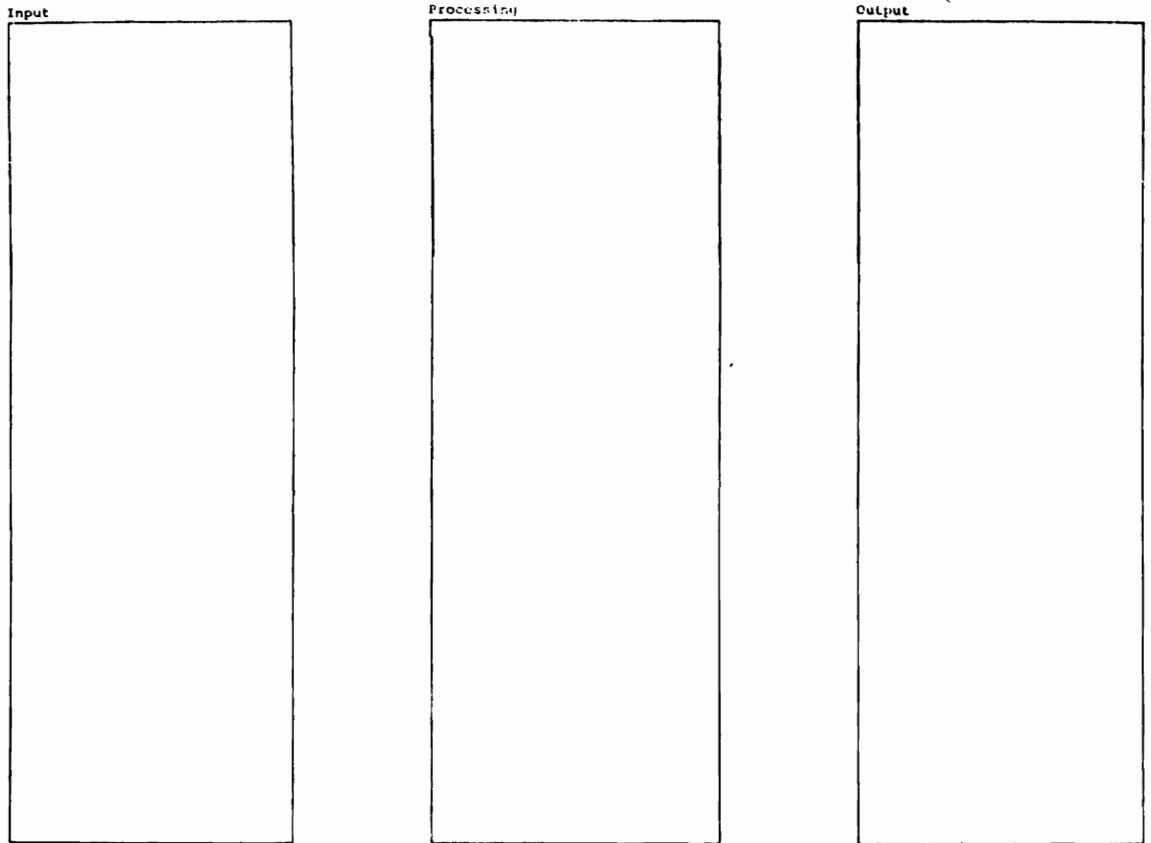
MO 3.4 (Part 1 of 2) 3767/3770 Terminal Output Manager (IKTTOMRT)

From IKTLTERM (MO 1.5), IKTVTPUT (MO 2.2), IKTXMTP (MO 2.3.1), IKTTIMRT (MO 3.3), IKTO9401 (MO 5.1), IKTO9409 (MO 5.4), IKTO940C (MO 5.6), or IKTO940E (MO 5.8) through Dispatcher



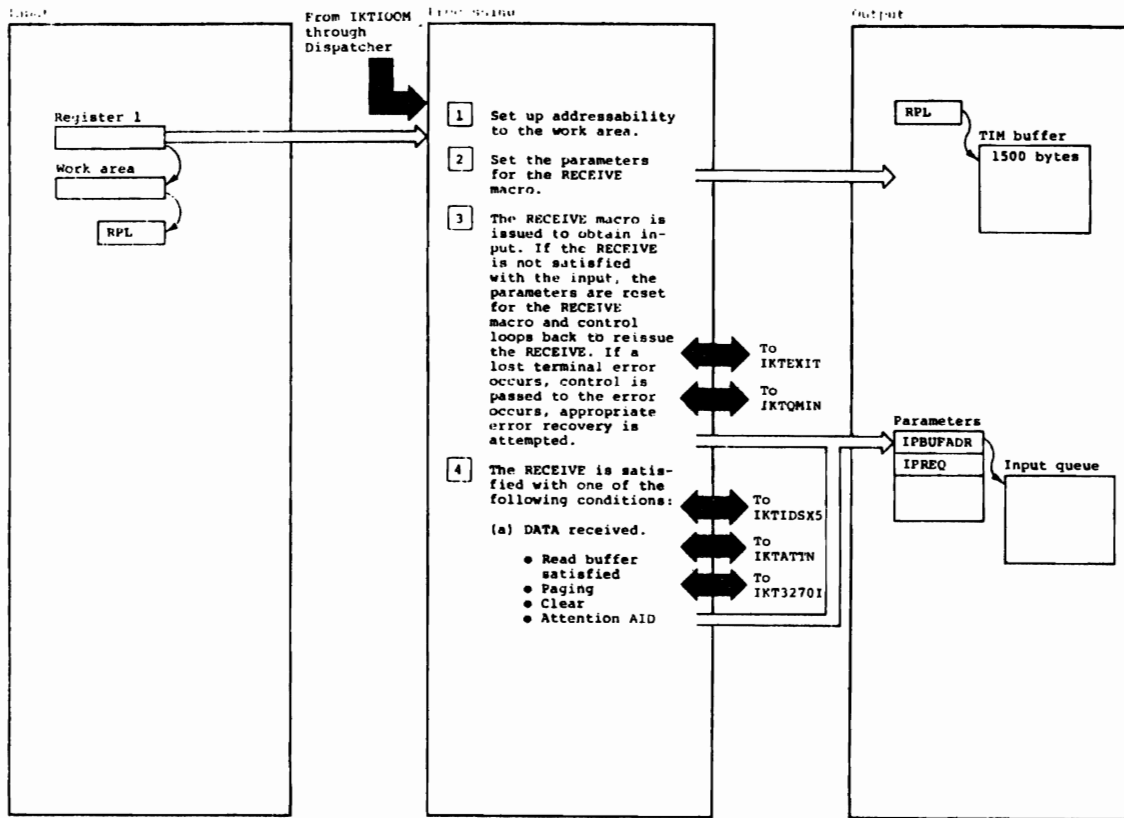
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine takes, from the output queue, output processed by the TPUT routine, and sends it to a VTAM buffer for transmission to an IBM 3767 or 3770 terminal.	IKTTOMRT			TPUT macro (EDIT, ASIS, or CONTROL), and placing the edited message into the RU.		EDITASIS CNTRLOPT	
1 The RPL for the SEND macro is initialized at first entry to IKTTOMRT. Functional recovery is established.	IKTTOMRT	INITFRR		Messages from one or more elements are put into a single RU. If a queue element contains a cross-memory message, a breakin message, or a control message, the message is the only one put into the RU. If a queue element contains any other kind of message, the message is put into the RU with messages of the same type from other elements, to fill the RU.		BLDMID	
Switches indicate which type of RU should be built.				(d) If prompting is specified and an RU is empty, a line number or character-prompt character is put into the RU.		PRMPTCHK	
(a) The SESSIONC macro is issued to send the Clear command and Start Data Traffic command. The character "I" is put into the RU for each accepted attention interruption. The characters "II" are put into the RU when an ignored attention interruption is detected.		CLRSDD		2 If requested (TERMINAL command), the edited output data is translated by using a user-specified translation table. If specified, the data is translated from EBCDIC to ASCII.	IKTTOMRT	TRANSL8	
(b) The SESSIONC macro is issued to send the Clear command and Start Data Traffic command. An error message is put into the RU.		ATTNROUT		3 A SEND macro is issued and the RU is sent as first, middle, last, or only RU in a chain. (Chain length is determined by the CHNLLEN value in parmib member TSOKEY00.) Functional recovery is reestablished. If the RPL indicates that buffer contention has prohibited the message from being sent, a signal is sent (SEND CONTROL= SIGNAL) to the message through if it is an	IKTTOMRT	SENDRU	
(c) Code 2 is passed in OPARMS to ask the queue manager to obtain the address of the message in the next available queue element. Installation-written exit routine IKTRTX1 is called, if present. Then normal editing may be performed, depending on the return code from IKTRTX1, by editing the message according to the operand specified on the		BLDMSG	4.2			INITFRR	
		EDITMSG					

MO 3.4 (Part 2 of 2) 3767/3770 Terminal Output Manager (IKTTOMRT)



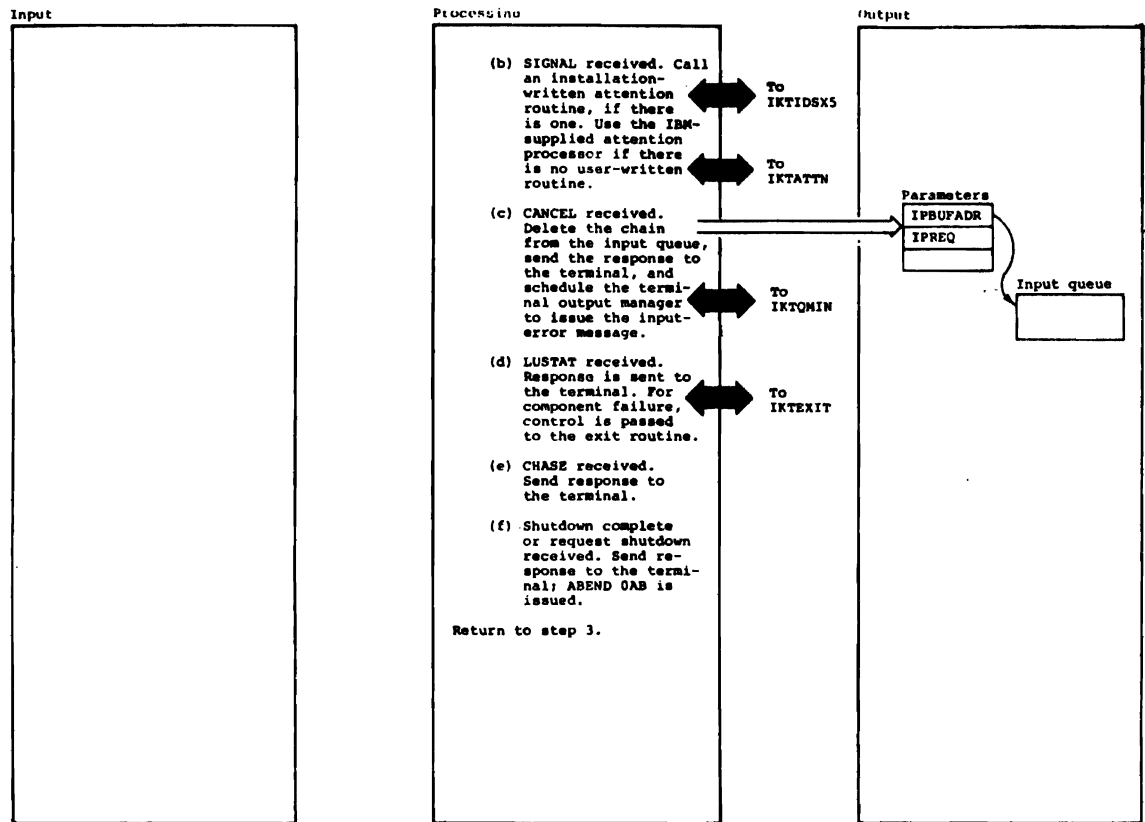
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
attention or attention ignored indication, an error message, or a breakin message. If the RPL indicates that the CANCEL key was pressed, the chain is canceled. If the RPL indicates an error condition, the RPL is rebuilt and the operation is retried three times. If the problem continues, ABEND 0AB is issued. 4 Code 3 is passed in OPARMS. 5 If the terminal input manager is no longer active, permanent cleanup is required: the TOM work area and SRB are freed (FREEMAIN macro) and control returns to the dispatcher. Otherwise, processing returns to step 1 until there are no more RUs to send.		CHKRESP					
		EXITFR					
		IKTTOMRT	BLDMSG	4.2			
		IKTTOMRT	CLEANXIT				

MO 3.5 (Part 1 of 2) 3270 LU2 Terminal Input Manager (IKTIMLU2)



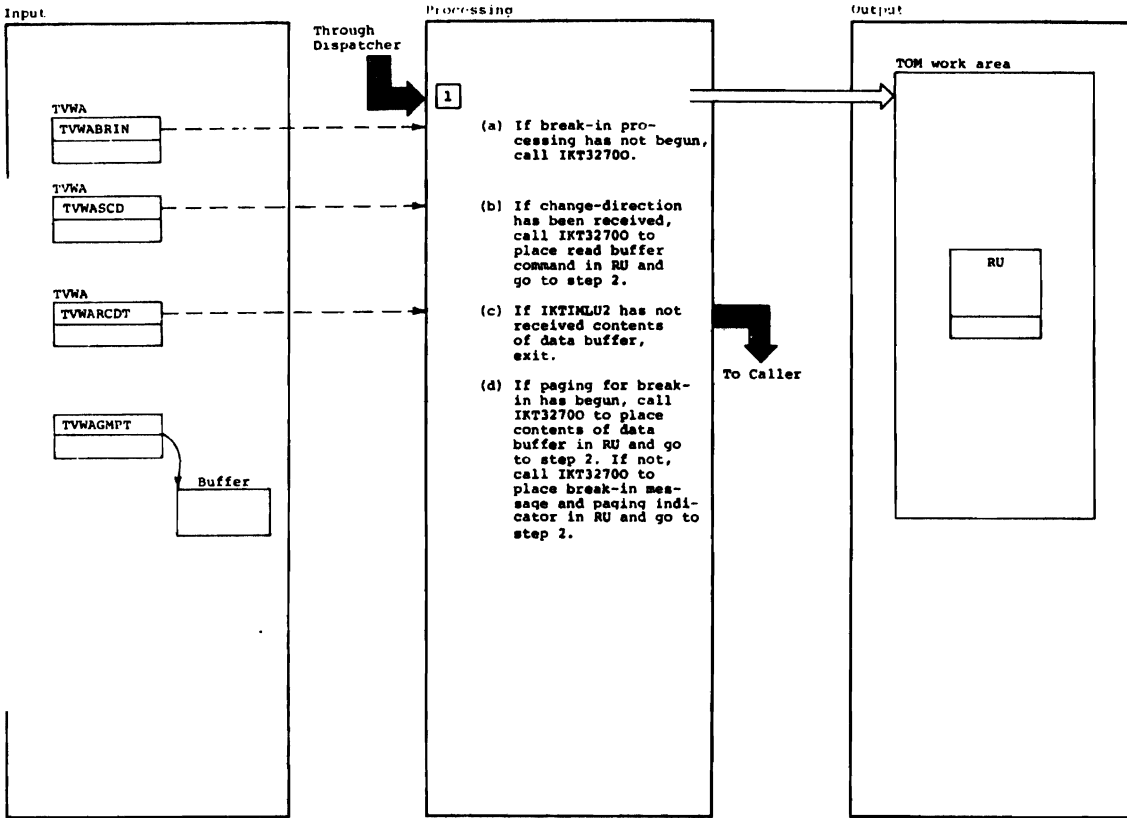
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
This routine issues the VTAM RECEIVE macro, determines what the input is when the RECEIVE is satisfied, calls the appropriate routine to process the input and loops back to reissue the RECEIVE. The RECEIVE may be satisfied with DATA, CANCEL, LUSTAT, SIGNAL, CCLEAR or some kind of error condition.				If the terminal output manager has not issued a Read-Buffer, a call to IKT3270I is issued to validate the data received. If the user is paging, no input is processed and only an attention AID is recognized. If the user pressed the CLEAR key, the line count (TVWALNCT) is reset to one, the indicator (TVWAFMSC) is turned on to format the screen, and the terminal output manager (IKTOMLU2) is scheduled to unlock the keyboard. If there is an attention interruption, a user-written attention routine is called if there is one. If there is no user supplied attention routine, the IBM-supplied attention routine is used. In support of full-screen mode during paging, a call is issued to IKT3270I to force an input reshow.	IKT3270I		
1 Upon initial entry to IKTIMLU2 the address of a work area is stored, the SRB is freed, and the recovery function is established.	IKTIMLU2	INITFRR					
2 This is the primary restarting point in case of failure due to an error condition. The RPL is initialized.	IKTIMLU2	PRIMPROC					
3 The recovery environment is reestablished. If the RECEIVE is unsuccessful, retry is attempted. If LOSTERM was encountered, IKTIMLU2 exits. If the retry is unsuccessful three times or if an unrecoverable error has occurred, ABEND OAB is issued.	IKTIMLU2	VTAMERR INITFRR				IKTIDSX5	SCHTOM
		EXITFRR				IKTATTN	
4 Check the RPL control fields for type of input.	IKTIMLU2	POSRESPR					
(a) Up to two RECEIVES are issued to complete the transfer of all the data from the VTAM buffer. If the terminal output manager has issued a Read-Buffer, the contents of the terminal buffer are retrieved into an area and translated from ASCII to EBCDIC if necessary.	IKTIMLU2	DATARTN					
		DATARBPF GETAREAR					
	IKT3270I	ASCITRAN					

MO 3.5 (Part 2 of 2) 3270 LU2 Terminal Input Manager (IKTIMLU2)



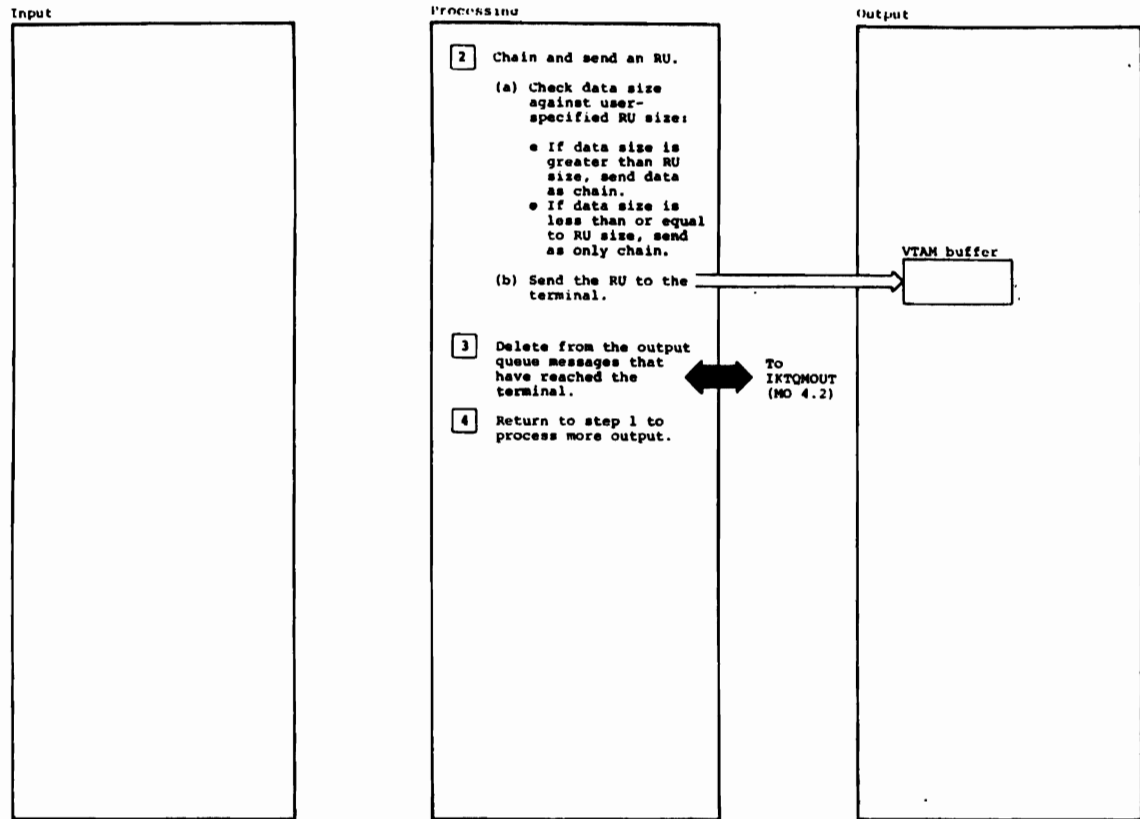
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
(c) The input queue manager (IKTQMIN) is called to delete the chain from the input queue, the response is sent to the terminal, the input-error flag (TVNAERMG) is set, and the terminal output manager is scheduled.	IKTIMLU2	CANCELR					
		SENDRESP					
		SCHTOM					
	IKTIMLU2	SCHTOM					
		SENDRESP					
		SENDRESP					
(d) If the component is available, schedule the terminal output manager and send the response to the terminal. If there is a component failure, send the response and exit. If integrity is lost, turn on input-error message (TVNAERMG) and formatting-erase-write flag (TVNAFNEW), schedule the terminal output manager, and send the response to the terminal. If none of the above has occurred, increase the error-receive count by 1 and send the response.		SCHTOM					
		SENDRESP					
		SENDRESP					

MO 3.6 (Part 1 of 2) 3270 LU2 Terminal Output Manager (IKTOMLU2)



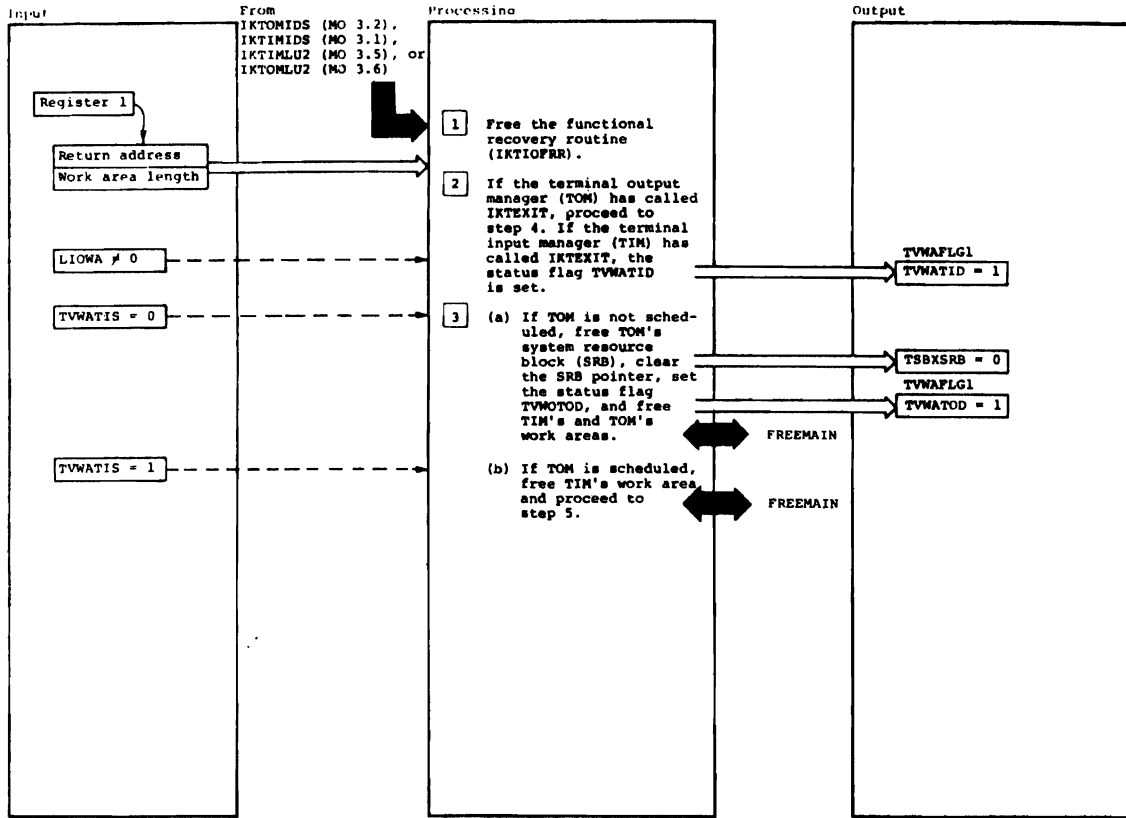
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine takes from the output queue, output processed by the TPUT routine, and sends it to a VTAM buffer for transmission to an IBM 3274, 3276, or 3278 terminal. Entry is through the Dispatcher from one of the following:</p> <p>IKTLTERM (MO 1.5) IKTVGET (MO 2.1) IKTVTPUT (MO 2.2) IKTXMPT (MO 3.3.1) IKTIMLU2 (MO 3.5) IKTO9401 (MO 5.1) IKTO9409 (MO 5.4) IKTO940C (MO 5.6) IKTO940E (MO 5.8)</p>							
<p>1 The RPL for the SEND macro is initialized at first entry to IKTOMLU2. The recovery function is established.</p>	IKTOMLU2	TOMINIT	INITFRR				
<p>(a) TVWABRIN is checked to see if break-in processing has begun.</p>	IKT32700						
<p>(b) The read buffer command is put in the RU.</p>	IKT32700						
<p>(d) If IKTIMLU2 has received the contents of the data buffer, and break-in paging has begun, place the contents in the RU.</p>							

MO 3.6 (Part 2 of 2) 3270 LU2 Terminal Output Manager (IKTOMLU2)



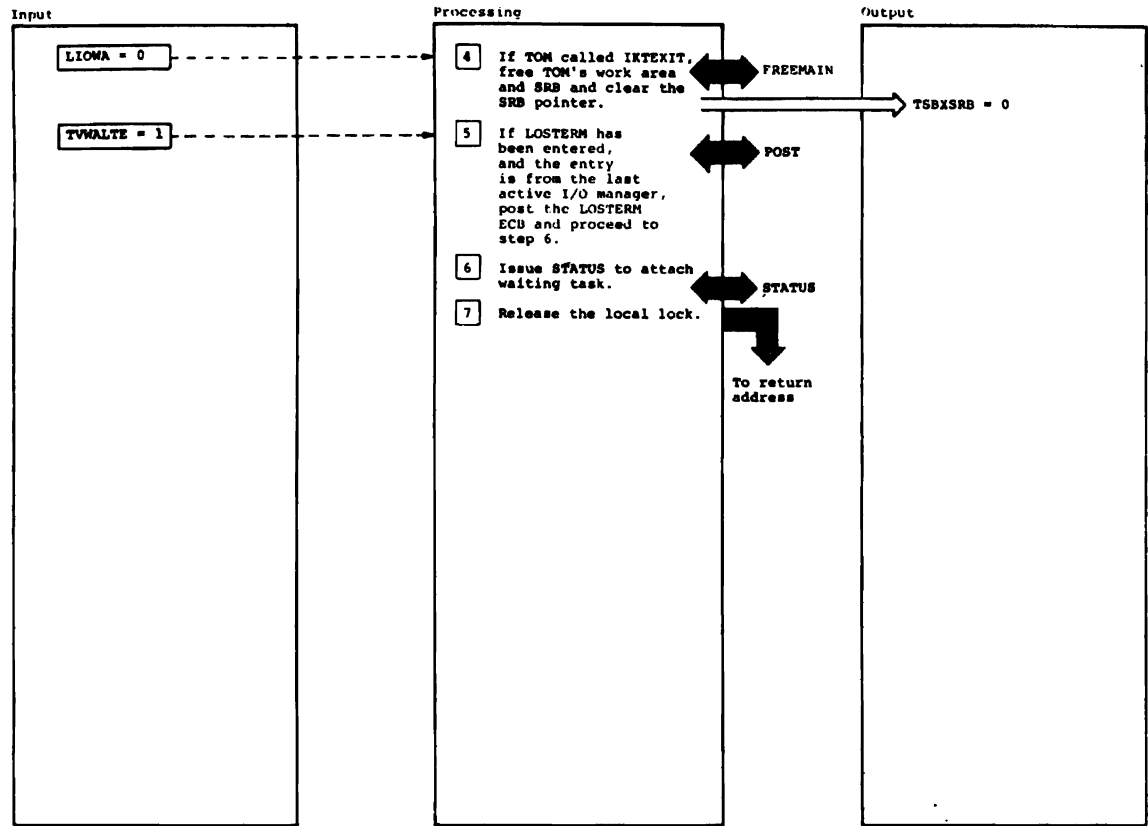
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>2 A SEND macro is issued. After the SEND is completed, the recovery function is established. If an error occurs attempting to send, and the chain element at that point is a beginning or middle element, IKTOMLU2 issues SEND CONTROL=CANCEL to cancel the part of the chain already sent (so that it will not be sent twice). The original SEND may be reissued three times before termination is required. If the RPL indicates that the terminal is in transmit mode, and the message to be sent is a break message, and the terminal is a break terminal, a signal is sent (SEND CONTROL=SIGNAL) to the terminal to try to force the message through.</p>	IKTOMLU2	INITFRR					
		EXITFRR					
<p>3 Code 3 is passed in OPARMS to the output queue manager, if positive function management end (PME) was received.</p>	IKTOMLU2						
<p>4 If the terminal input manager is no longer active, permanent cleanup is required. IKTEXIT is called to free the TOM's work area and SRB, and control returns to the dispatcher. Otherwise, processing returns to step 2 until there are no more RUs to send.</p>	IKTEXIT						

MO 3.7 (Part 1 of 2) Exit Routine (IKTEXT)



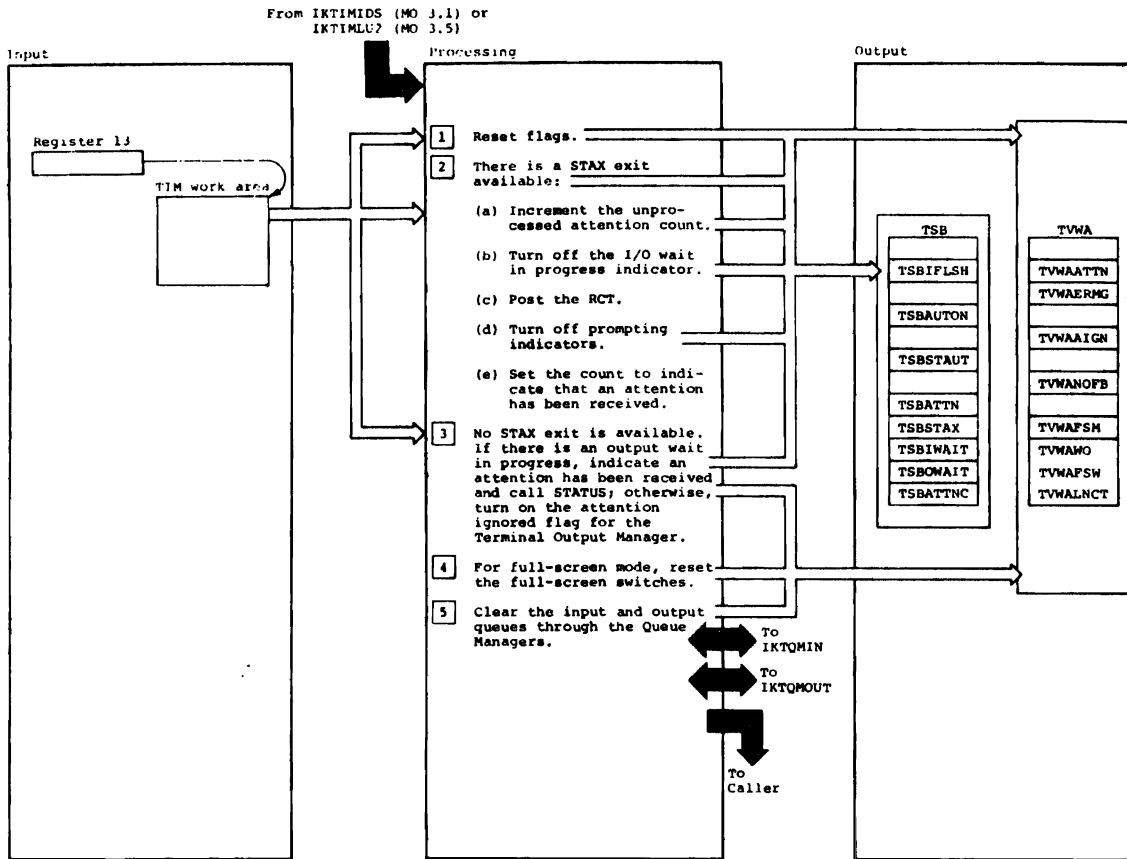
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine does cleanup. If it has been called by TIM when TOM is not scheduled, it:</p> <ul style="list-style-type: none"> • Frees TOM's SRB • Frees TIM's & TOM's work areas • Calls IKTLTERM to post the ECB <p>If it has been called by TOM when TOM is scheduled, it frees TIM's work area. If this routine is called by TOM, it:</p> <ul style="list-style-type: none"> • Frees TOM's work area • Frees TOM's SRB • Calls IKTLTERM to post the ECB <p>Note: TOM can only call IKTEXT if TVWATID is on (TVWATID = 1).</p> <p>1 This stack entry is used by the functional recovery routine (FRR) if an error condition is encountered.</p> <p>2 If the third parameter field (LIOWA) is not zero, TIM has called the exit routine. IKTEXT sets the status bit (TVWATID) indicating that TIM has exited normally and freed the work area.</p>				<p>3 If TOM is not scheduled, the status bit (TVWATIS) is equal to zero. In this case, TIM's and TOM's work areas and TOM's SRB are released. The status bit (TVWATOD) indicating that TOM has exited normally and freed the work area is set (TVWATOD).</p>			

MO 3.7 (Part 2 of 2) Exit Routine (IKTEXT)



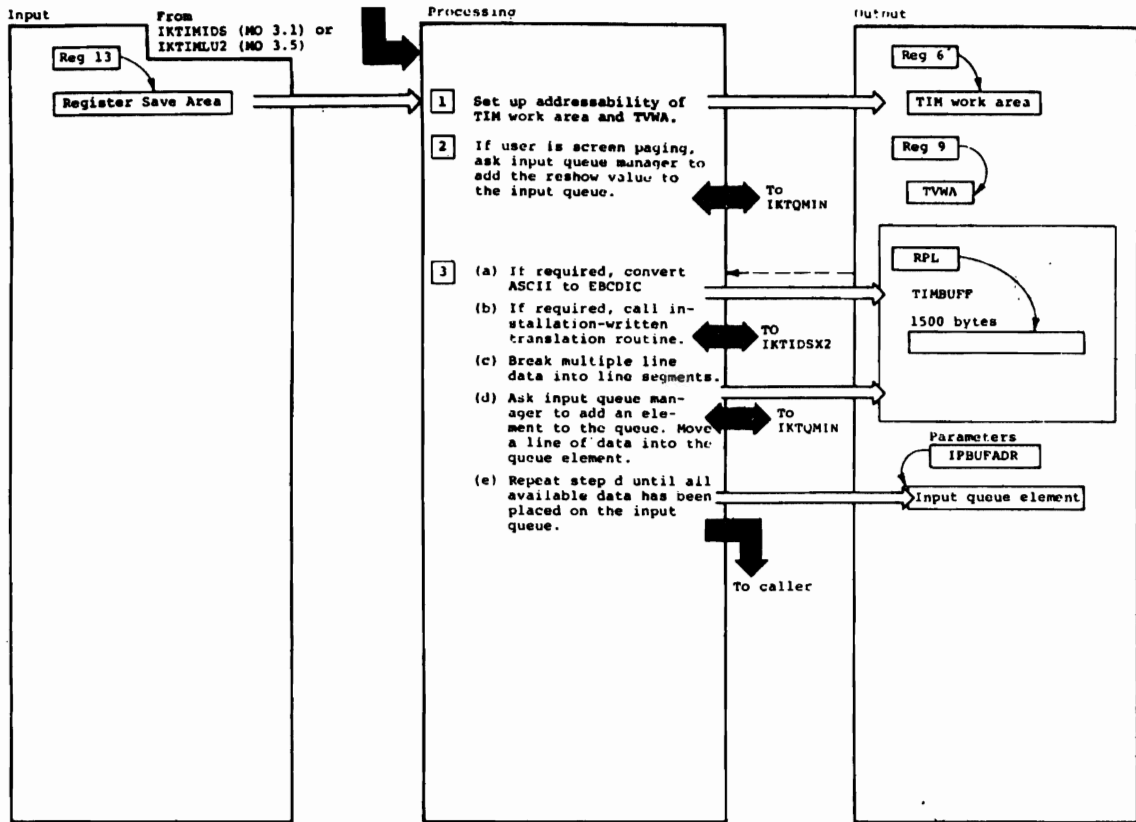
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>5 If LOSTERM was entered, the status flag TVWALTE is set to one.</p>							

MO3.8 Attention Routine (IKTATTN)



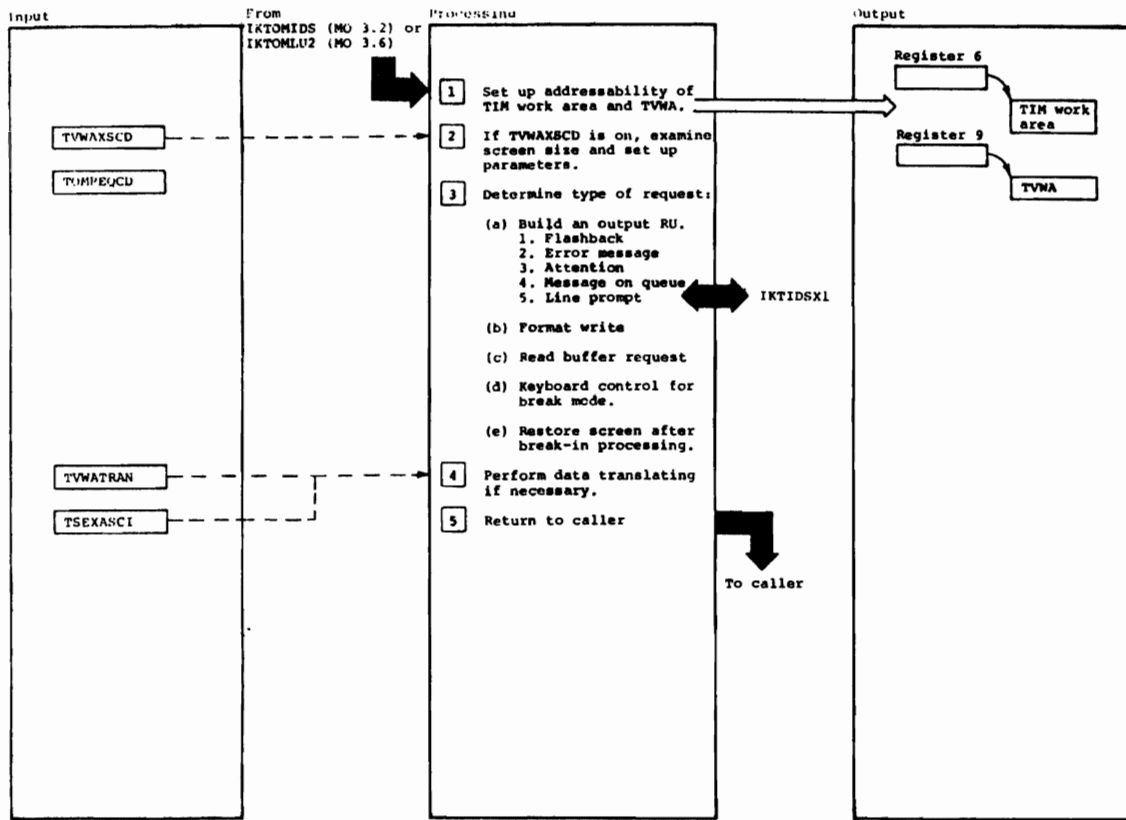
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine receives control when the VTAM input buffer contains an attention attribute character and there is no user-written attention routine.</p> <p>1 Turn off the error-on-input flag (TVWAERMG), the flashback flag (TVWANOFB), and the input flush flag (TSBIFLSH) as all elements are to be removed from the queues.</p> <p>2 The number of unscheduled STAX exits (TSBSTAX) is greater than the number of unprocessed attentions (TSBATTNC).</p> <p>(a) Increase the number of STAX taken in the TSB (TSBATTNC).</p> <p>(b) Turn off the TSB wait switches (TSBOWAIT, TSBWAIT).</p> <p>(c) Issue the POST macro to indicate the occurrence of the region control task (RCT).</p> <p>(d) Turn off the auto-prompting-requested flag (TSBAUTON) and the flag to prompt the user with the next line number (TSBSTAUT).</p> <p>(e) Increase the STAX attention count (TVWAATTN) by 1.</p>	IKTATTN			<p>3 There is no STAX exit. If TSBOWAIT shows an output wait in progress, reset TSBOWAIT, set the indicator to show an attention received (TSBATTN), and call STATUS to notify the user of the status of the output request. If there is no output wait in progress, turn on the attention-ignored flag (TVWAIGN) for the terminal output manager.</p> <p>4 For full-screen mode, reset the full-screen switch (TVWAFSM), the written-over switch (TVWAMO), and the full-screen TPUT waiting switch (TVWAFSW); set the line count (TVWALNCT) to zero.</p> <p>5 In non-full-screen mode, increase the line count (TVWALNCT) by 1, and set the flag to schedule the terminal output manager (TIMSCHTM). Use the queue managers to clear the input and output queues of non-cross-memory messages.</p>	IKTATTN		

MO 3.9 Input Data Handling Routine (IKT3270I)



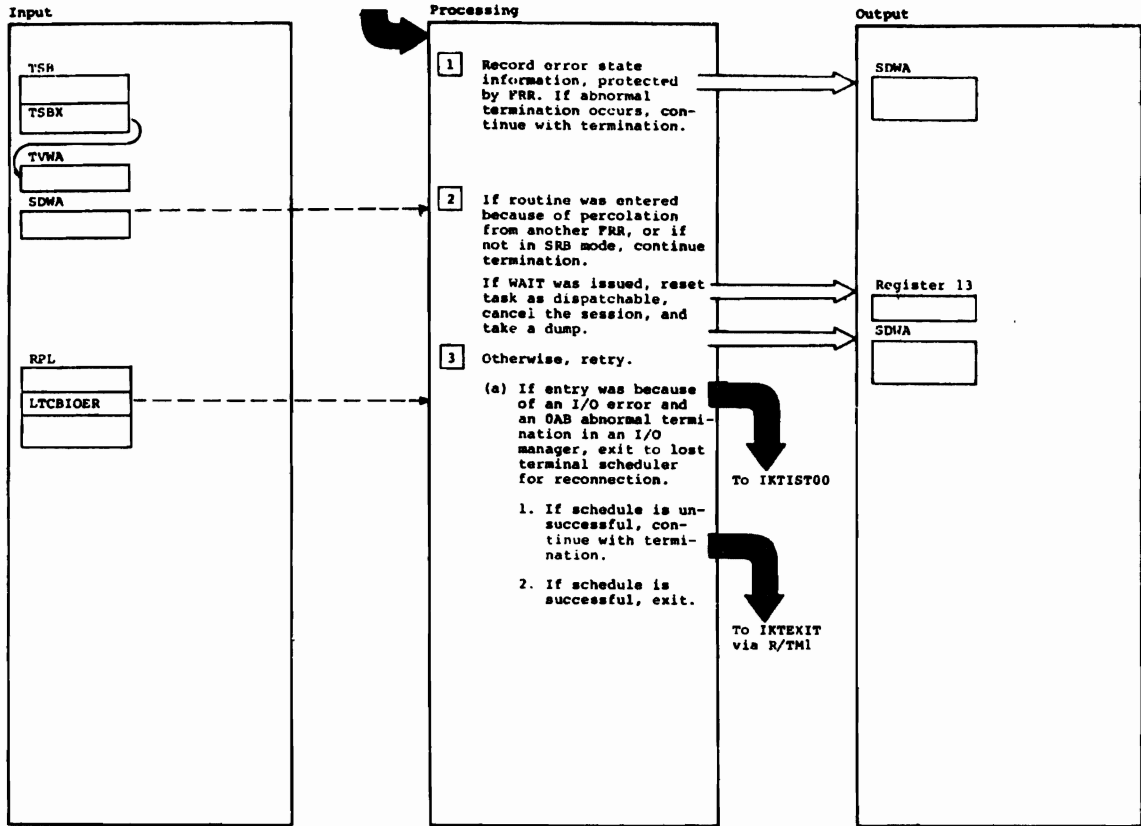
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine receives input data, checks it, and places it on the input queue for processing by the correct TSO command processor.</p> <p>1 Processing begins with the following registers initialized: Reg 6 : TIM work area Reg 9 : TVWA</p> <p>If the input queue is being flushed, a check is made to determine if another RECEIVE is necessary, in which case return is to the caller. If another RECEIVE is not necessary, schedule the output manager.</p> <p>2 If the user is paging, a request is made to the input queue manager to add the reshow value (TVWARSHW) to the input queue.</p> <p>3</p> <p>(a) A check is made for ASCII input, which is translated by calling the ASCITRAN routine.</p> <p>(b) If a user translation table is available (IKTIDSX2 not zero), exit by calling IKTIDSX2. Register 0 contains the address, and register 1 contains the length of the data to be translated.</p>	IKT3270I			<p>(c) Data is scanned for input line delimiters. Multiple line data is broken into single lines. Headers are placed on each line by the HEADPROC routine.</p> <p>(d) Code 1 is passed in IPARMS to ask the queue manager to add an element to the bottom of the input queue. If requested by the user, character translation is performed after the element is obtained. If processing is not done in full screen mode, LINECNT is called to update the current line number and to save the input to be flashed back to the terminal operator.</p>	IKT3270I		
						HEADPROC	
						ADDELRTN	
						USERTRAN	
						LINECNT	
	IKT3270I	ASCITRAN					
	IKTIDSX2						

MO 3.10 Output Data Handling Routine (IKT32700)



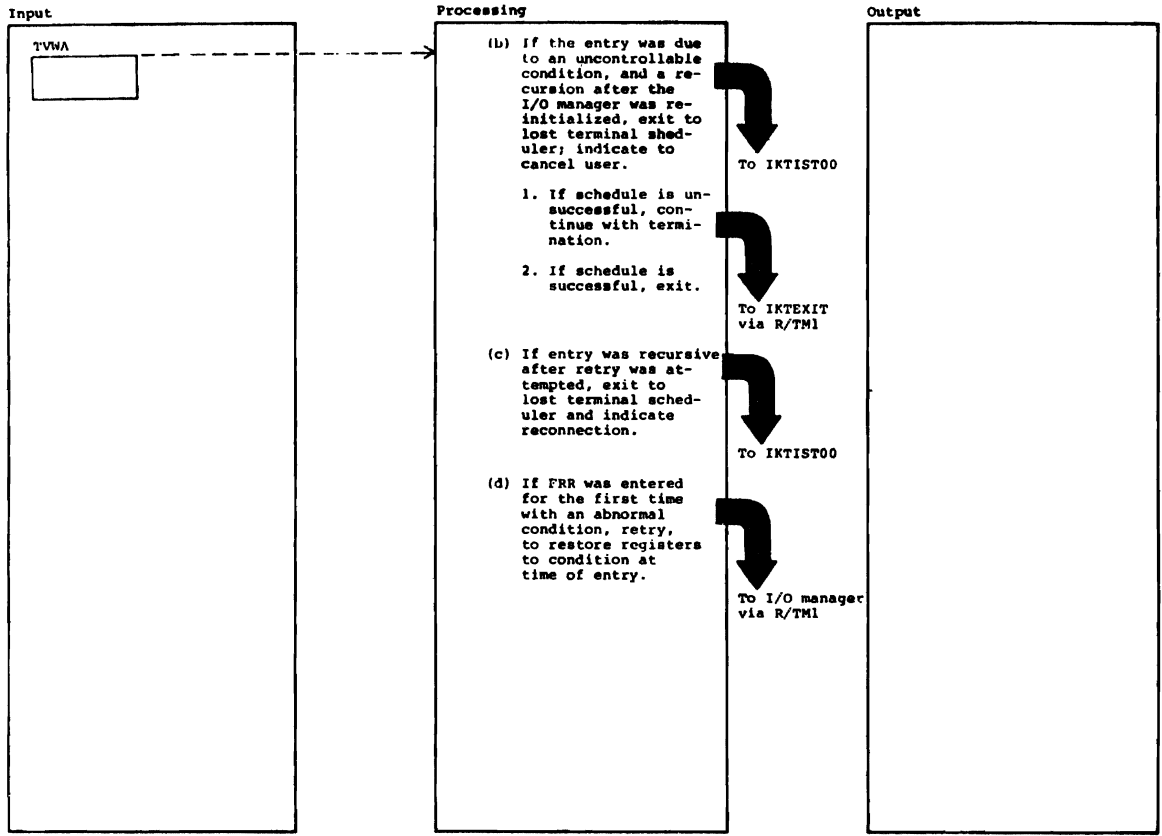
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>1</p> <p>3 (a) 1. The last logical line of screen input is put into the RU for display at the top of the screen.</p> <p>2. An error message is put into the RU for display at the terminal.</p> <p>3. The character "1" is put into the RU for placement on the next available screen line for each accepted attention interruption. The characters "11" are put into the RU for placement on the next available screen line when an ignored attention interruption is detected.</p> <p>4. Code 2 is passed in OPARMS to ask the queue manager to obtain the address of the message in the next available queue element. Installation-written exit routine IKTIDSX1 is called, if present. Then normal editing may be performed, depending on the return code from IKTIDSX1, by editing the message according to the operand specified on the TPUT macro (EDIT, ASIS, CONTROL, or FULSCRN), and placing the edited message into the RU. This cycle of obtaining a message, editing it, and placing it into the RU continues until the output queue is empty.</p>	IKT32700			<p>5. A line number or auto-prompt character is put into the RU, and the keyboard is unlocked.</p> <p>(b) Format write is done if IKTIMLU2 or IKTUMLU2 finds that screen integrity has been lost.</p> <p>(c) During break processing, the terminal buffer must be read to maintain data integrity. A Read Buffer command is sent to the terminal.</p> <p>(d) In break mode, the keyboard is opened after each user input, whether or not a TGET is outstanding. The cursor is repositioned and the keyboard is unlocked.</p> <p>(e) The buffer contents is restored after break processing.</p> <p>4 If the TVWATRAN bit is on, the data is translated according to the user translation table pointed to by TVWATABO. If ASCII translation is required (TSBXASCI = 1), translation is done using the table pointed to by TVWAATBO.</p>			

MO 3.11 (Part 1 of 2) I/O Functional Recovery Routine (IKTIOFFR)



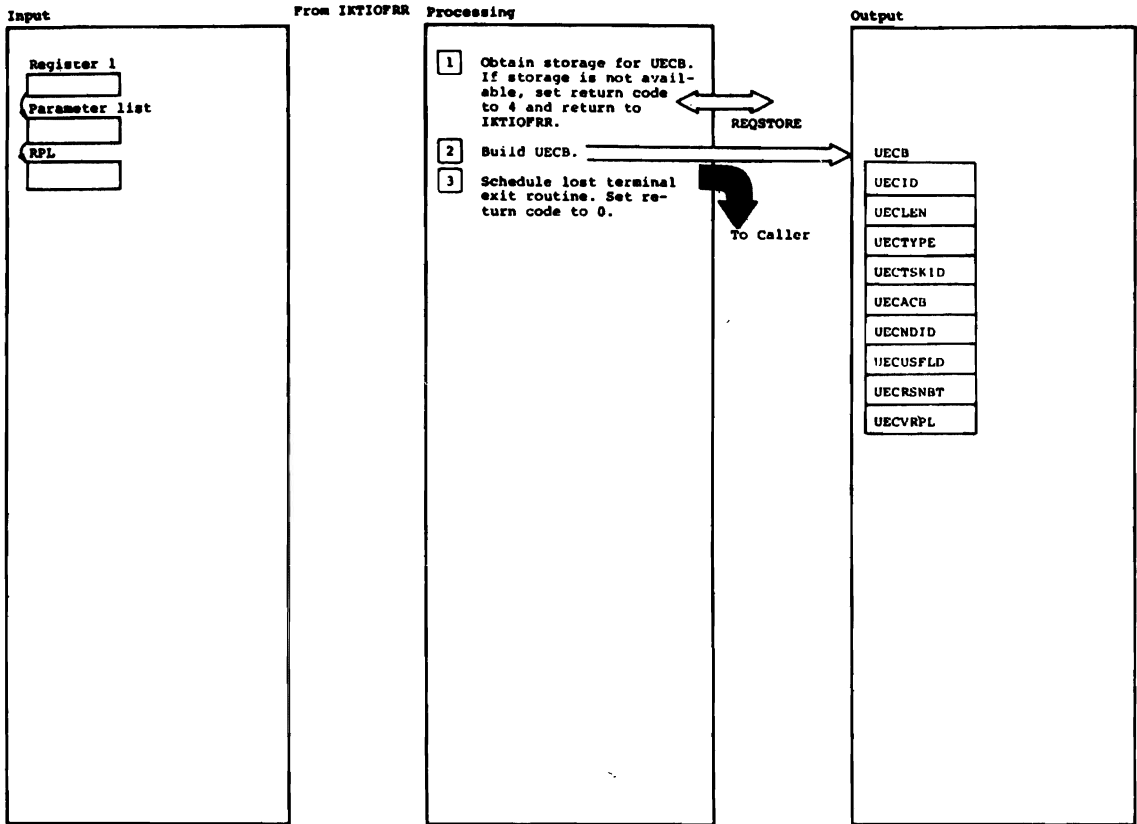
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
In this routine, "retry" means the restoration of registers and passage of control to a retry point. Continuation with termination allows percolation of the termination of a task. Percolation is the process of giving other higher-level error handler routines a chance to recover.							
1	IKTIOFFR	DOCUMENT					
2 If there is no user code, the second digit of the reason code is set to 1. If there is a user code, register 15 is set to 0.							
3 (a)	IKTIOFFR	CALLRTM RETRY					

MO 3.11 (Part 2 of 2) I/O Functional Recovery Routine (IKTIOFFR)



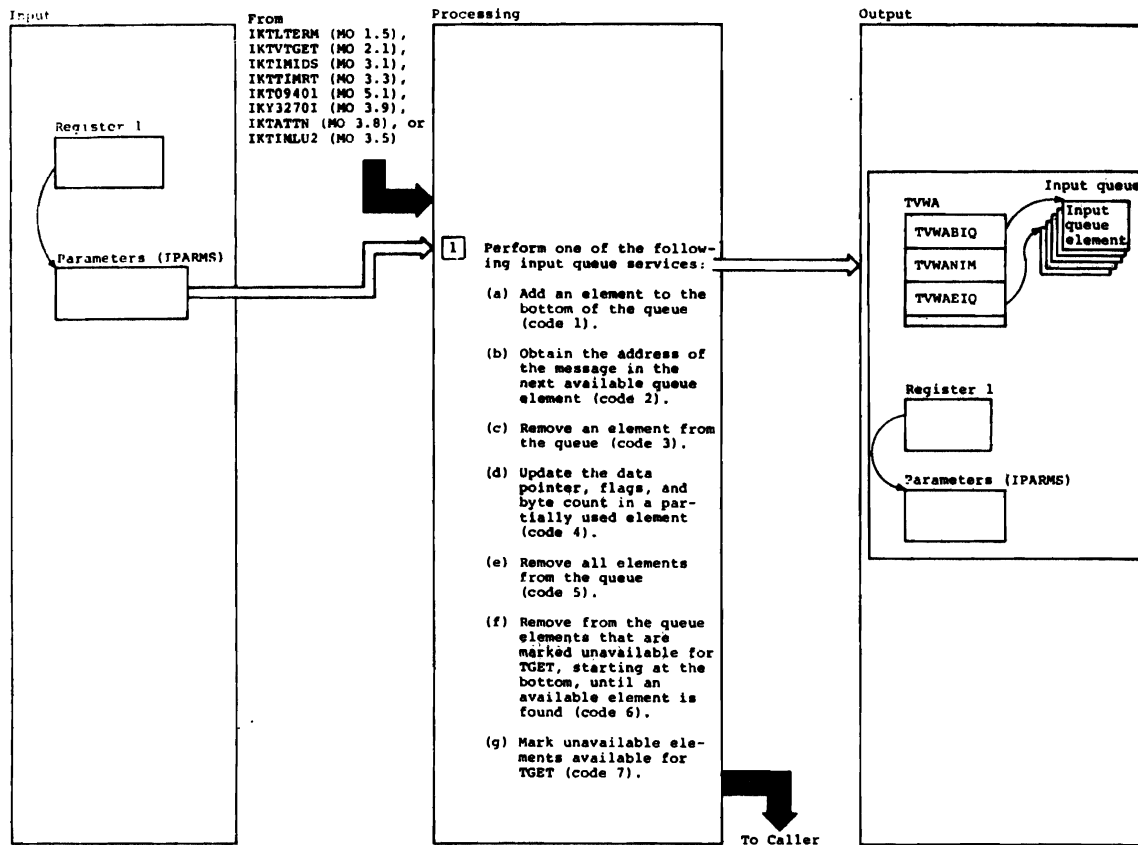
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
(b) The error occurred in SRB mode and was due to a machine check, program check, or other uncontrollable condition.	IKTIOFFR	RETRY					
(c) Logon reconnection completely reconstructs all local control blocks and local work areas for the I/O managers.	IKTIOFFR	EXIT					

MO 3.12 Lost Terminal Exit Scheduler (IKTIST00)



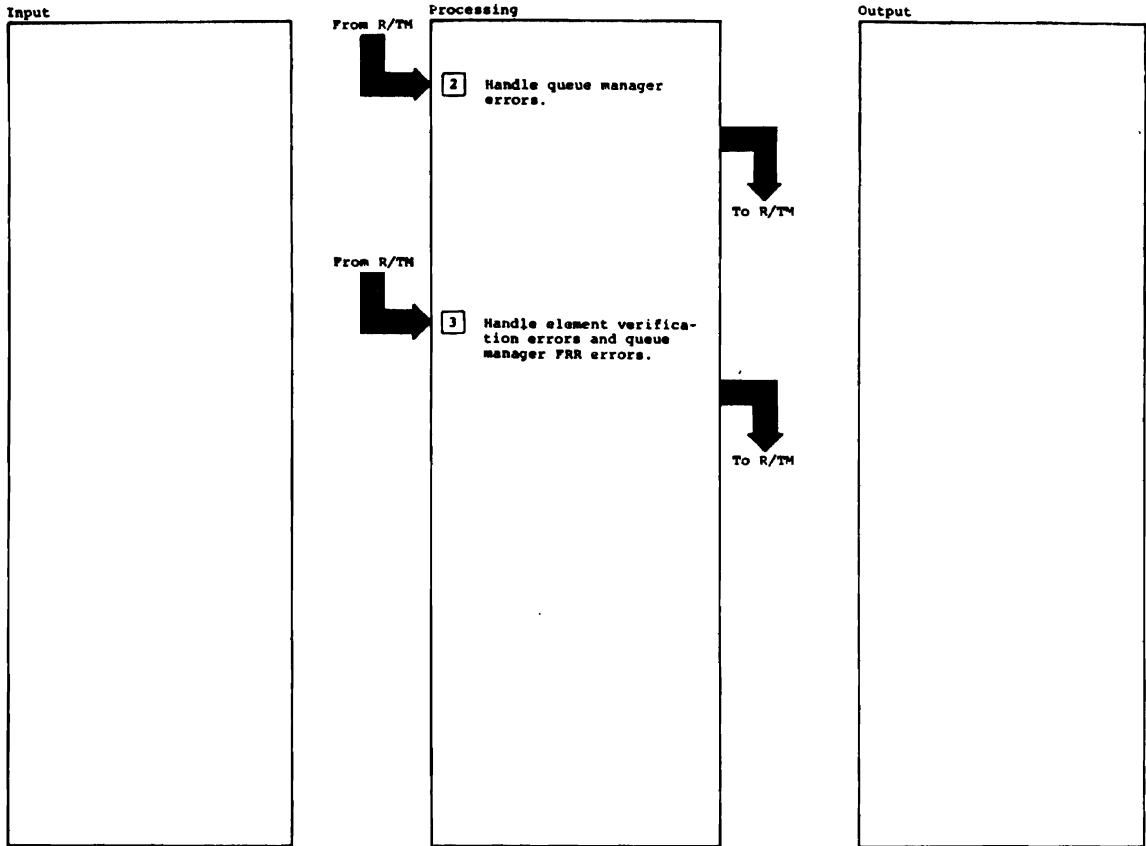
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>2 The UECB is used by the lost terminal exit routine.</p> <p>3 TPQUEUE is issued to queue the UECB to the UECPAB.</p>	IKTIST00						

MO 4.1 (Part 1 of 3) Input Queue Manager (IKTQMIN)



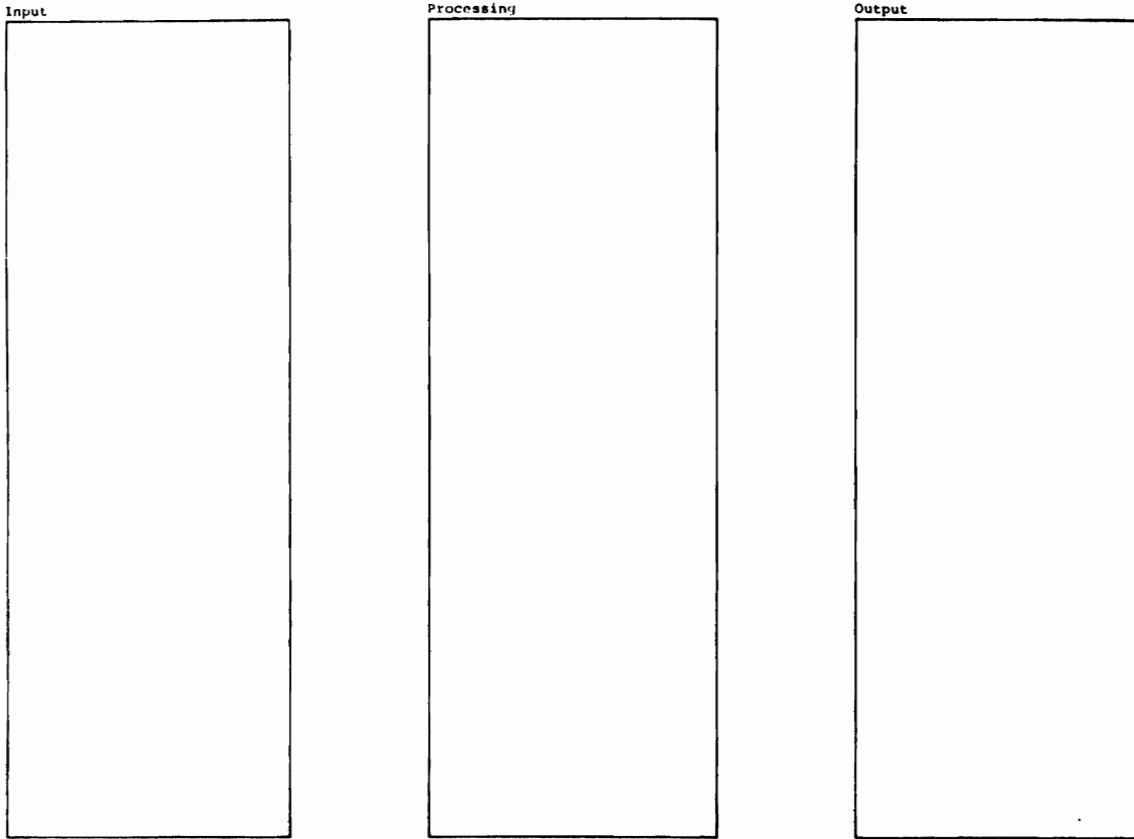
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine adds, updates, and removes input queue elements.</p> <p>1 The request code in IPARMS indicates which service to perform.</p> <p>(a) Storage is allocated (GETCELL for normal size element, GETMAIN for larger than normal size element) for the element and pointers are manipulated to add the element to the queue. If GETCELL or GETMAIN is unsuccessful, ABEND 0AD is issued.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 1 Buffer size (IPBSZ) = Message length Flags (IPFLAGS) = Flag settings requested by caller</p> <p>Output parameters (IPARMS): Return code (IPRC): 1 = Successful 2 = Unsuccessful; no room for element Buffer address (IPBUFADR) = Location of message area of queue element</p> <p>(b) Input parameters (IPARMS): Request code (IPREQ) = 2</p> <p>Output parameters (IPARMS): Return code (IPRC): 1 = Successful and there are more elements on the queue</p>	IKTQMIN IKTQMIN	BLDELE ADDEL OBTAIN		<p>2 = Successful and this is the last element on the queue 3 = Unsuccessful; the queue is empty Buffer address (IPBUFADR) = Location of queue element's message area Buffer size (IPBSZ) = Message length Flags (IPFLAGS) = Flag settings requested by caller</p> <p>(c) The element's storage is freed (FREECELL or FREEMAIN, depending on how it was allocated) and pointers are manipulated to remove the element from the queue. If FREECELL or FREEMAIN is unsuccessful, ABEND 0AD is issued. If any tasks are waiting because of buffer overuse they are dispatched (STATUS macro routine). If a task is waiting in another address space, IKTXMPTT is rescheduled (SCHEDULE SRB) in this address space.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 3</p> <p>Output parameters (IPARMS): Return code (IPRC): 1 = Successful and there are more elements on the queue</p>		DELEL DELIN MARKDISP	2.3.1

MO 4.1 (Part 2 of 3) Input Queue Manager (IKTOMIN)



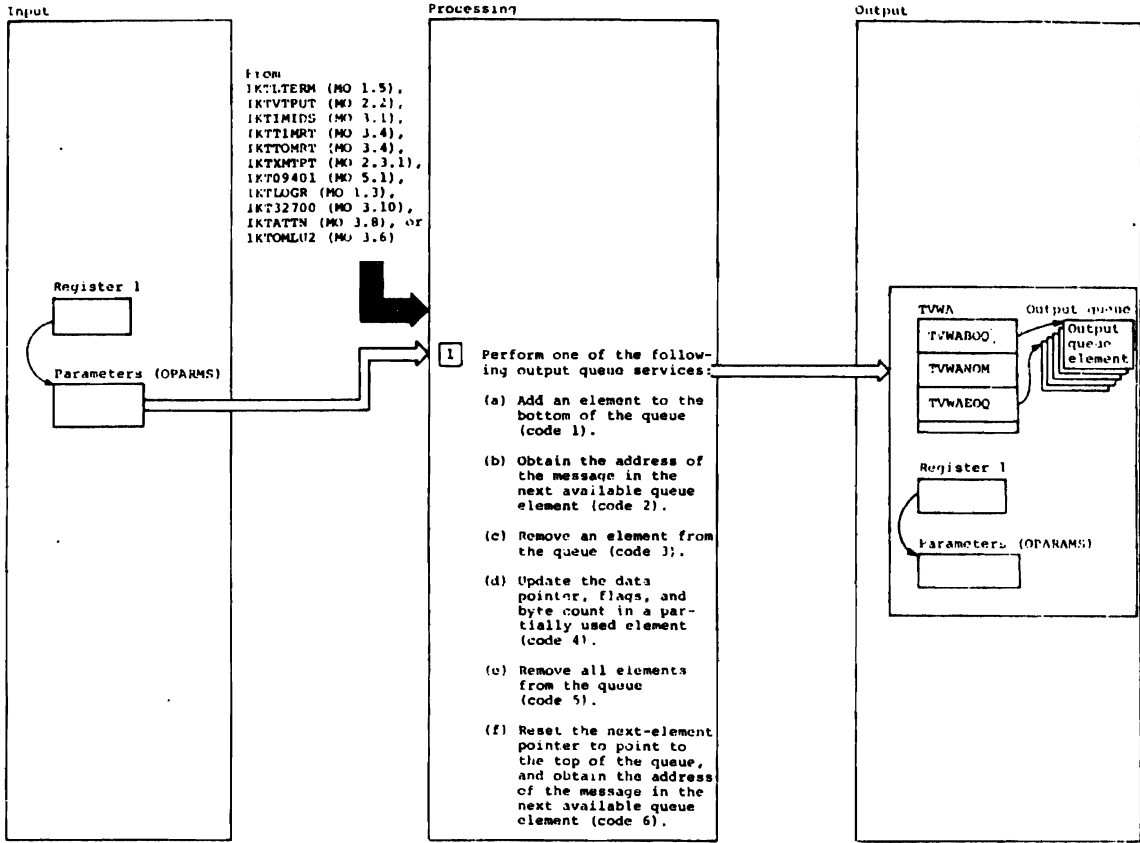
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>2 = Successful and there are no more elements on the queue</p> <p>(d) Updating is required when the TGET requester's buffer is smaller than the message in the input queue.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 4 Buffer address (IPBUFADR) = Address of first unused byte of message</p> <p>Output parameters (IPARMS): Return code (IPRC): 1 = Successful 2 = Unsuccessful; the queue is empty</p> <p>(e) Storage occupied by the elements is freed (FREECELL and/or FREEMAIN), and the current line number is adjusted for line prompting according to the number of elements removed from the queue.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 5</p> <p>Output parameters (IPARMS): None</p>		UPDATE		<p>(f) The storage occupied by the elements marked unavailable is freed (FREECELL or FREEMAIN) and pointers are manipulated to remove the elements from the queue. If FREECELL or FREEMAIN is unsuccessful, ABEND 0AD is issued. If any tasks are waiting because of buffer overflow they are dispatched. (STATUS macro routine). If a task is waiting in another address space, IKTMTPT is rescheduled (SCHEDULE SRB) in this address space.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 6</p> <p>Output parameters (IPARMS): None</p> <p>(g) Elements are marked available when there is no room on the input queue for more elements.</p> <p>Input parameters (IPARMS): Request code (IPREQ) = 7</p> <p>Output parameters (IPARMS): None</p>		DELEL DELBK MARKDISP MARKAVBL	2.3.1
		CLEAR1					

MO 4.1 (Part 3 of 3) Input Queue Manager (IKTQMIN)



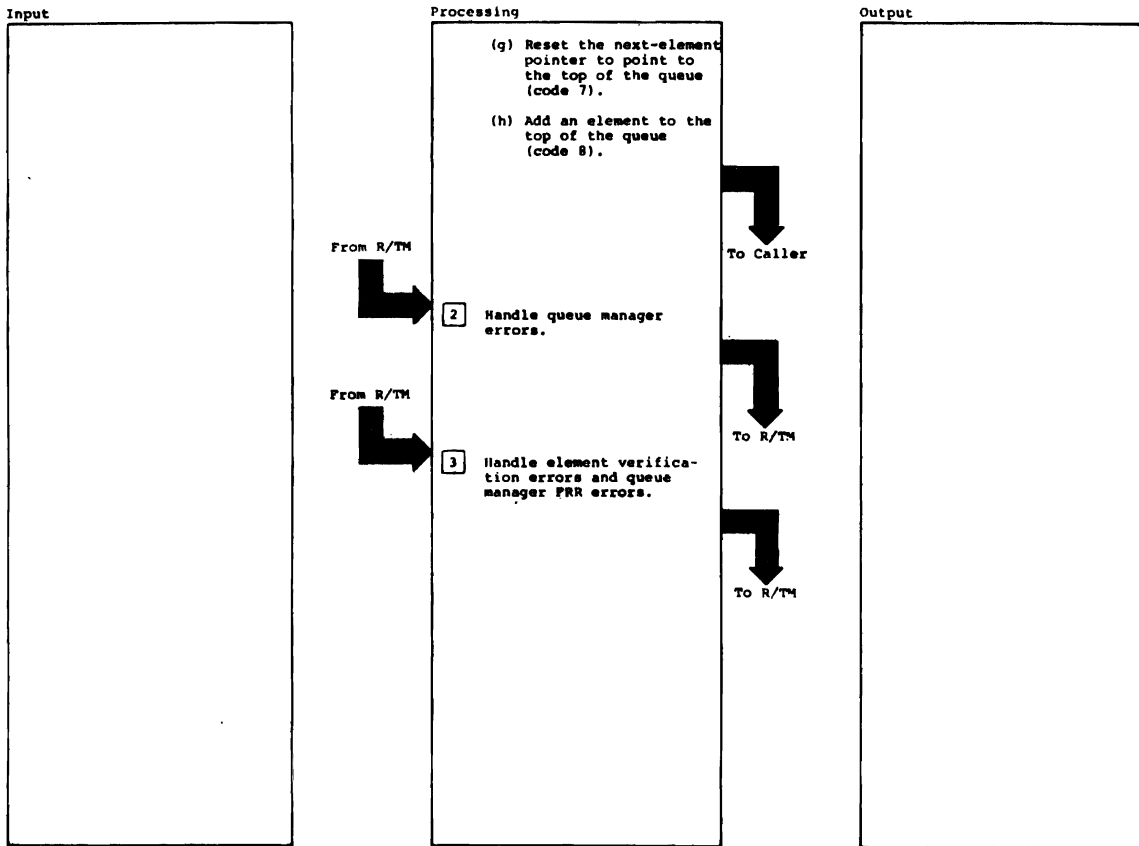
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>2 IKTQMF1 is the first queue manager FRR. It receives control when the queue manager encounters an error. It builds a queue verification parameter list (QVPL), calls IEAVEQVJ to verify and correct the list, logs the TVWA into the SDWA, and retries the queue manager once. (IEAVEQVJ calls IKTQMEV to ensure that queue element size and buffer size are acceptable. If an unacceptable element is found, message IKT00400Y (input data lost) is sent to the terminal.)</p>	IKTQMF1						
<p>3 IKTQMF2 is the second queue manager FRR. It receives control when an error occurs during IKTQMF1 processing. If the element verification routine (IKTQMEV) has failed, the element is marked unacceptable and verification continues. If the queue manager FRR has failed, the TVWA is logged into the SDWA and termination processing continues.</p>	IKTQMF2						

MO 4.2 (Part 1 of 3) Output Queue Manager (IKTQMOUT)



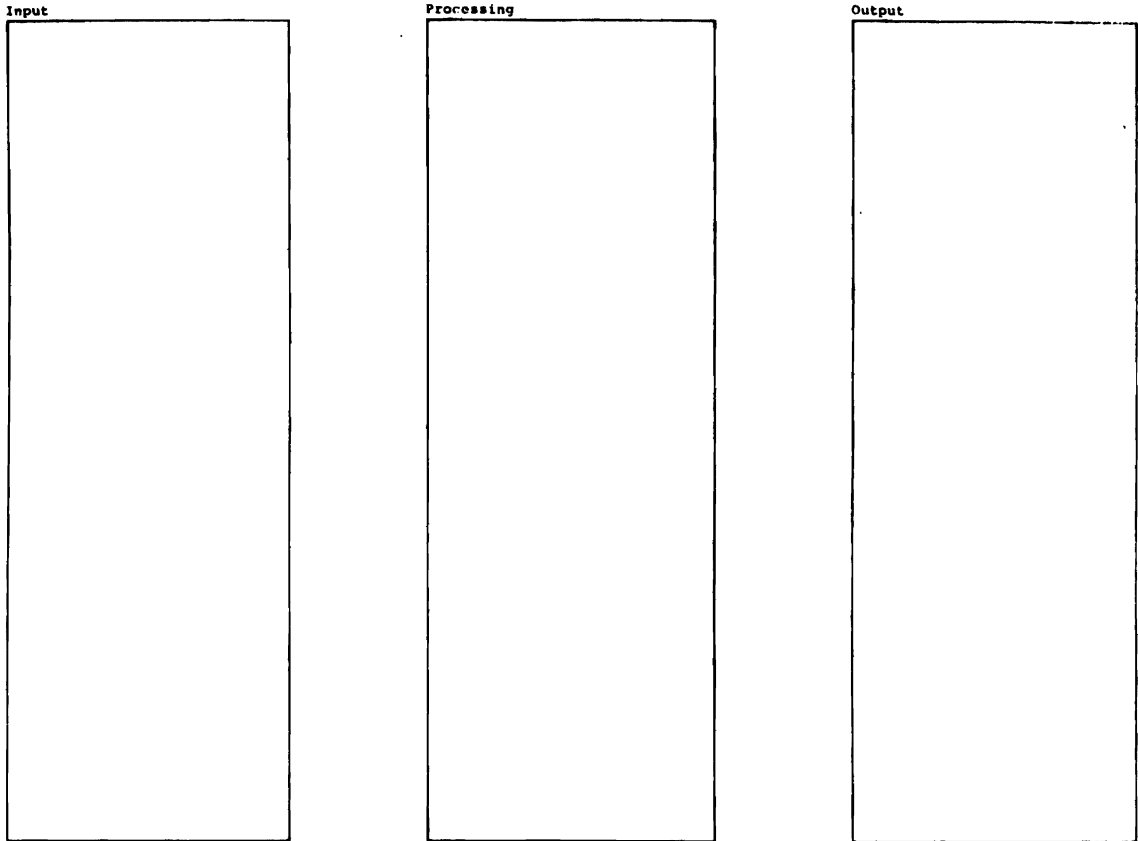
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref	
<p>This routine adds, updates, and removes output queue elements.</p> <p>1 The request code in OPARMS indicates which service to perform.</p> <p>(a) Storage is allocated (GETCELL for normal size element, GETMAIN for larger than normal size element) for the element and pointers are manipulated to add the element to the queue. If GETCELL or GETMAIN is unsuccessful, ABEND OAD is issued.</p> <p>Input parameters (OPARMS): Request code (OPREQ) = 1 Buffer size (OPBFSZ) = Message length Flags (OPFLAGS) = Flag settings requested by caller</p> <p>Output parameters (OPARMS): Return code (OPRC): 1 = Successful 2 = Unsuccessful; no room for element (parmlib limit exceeded)</p> <p>Buffer address (OPBUFADR) = Location of message area of queue element</p> <p>(b) Input parameters (OPARMS): Request code (OPREQ) = 2</p> <p>Output parameters (OPARMS): Return code (OPRC): 1 = Successful and there are more elements on the queue</p>	IKTQMOUT			<p>2 = Successful and this is the last element on the queue 3 = Unsuccessful; the queue is empty Buffer Address (OPBUFADR) = Location of queue element's message area Buffer size (OPBFSZ) = Message length Flags (OPFLAGS) = The element's flags</p> <p>(c) The element's storage is freed (FREECELL or FREEMAIN, depending on how it was allocated) and pointers are manipulated to remove the element from the queue. If FREECELL or FREEMAIN is unsuccessful, ABEND OAD is issued. If any tasks are waiting because of buffer overflow they are dispatched (STATUS macro routine). If a task is waiting in another address space, IKTWMPT is rescheduled (SCHEDULE SRB) in this address space.</p> <p>(d) Updating is required when the amount of data that can be transmitted with a SEND is smaller than the message on the output queue.</p> <p>Input parameters (OPARMS): Request code (OPREQ) = 4 Buffer address (OPBUFADR) = Address of first unused byte of message</p>				
			BLDELE					
			ADDO				DELEL DELETO MKREDISP	
			OBTAINO				UPDATO	

MO 4.2 (Part 2 of 3) Output Queue Manager (IKTQMOUT)



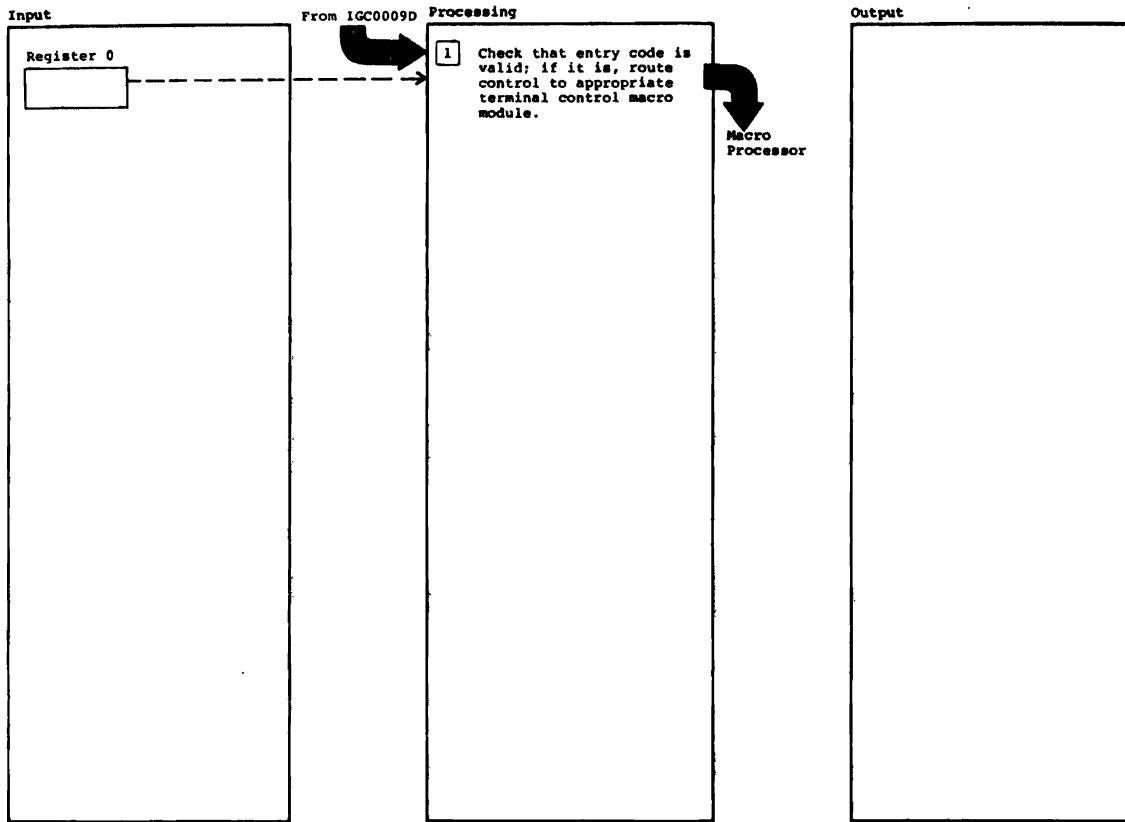
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>Output parameters (OPARMS): Return code (OPRC): 1 = Successful 2 = Unsuccessful; the queue is empty</p> <p>(e) The element's storage is freed (FREECELL or FREEMAIN) and pointers are manipulated to remove all elements not from another address space from the queue. If FREECELL or FREEMAIN is unsuccessful, ABEND OAD is issued. If any tasks are waiting because of buffer overuse they are dispatched (STATUS macro routine). If a task is waiting in another address space, IKTXMTPT is rescheduled (SCHEDULE SRB) in this address space.</p> <p>Input parameters (OPARMS): Request code (OPREQ) = 5</p> <p>Output parameters (OPARMS): None</p> <p>(f) Input parameters (OPARMS): Request code (OPREQ) = 6</p> <p>Output parameters (OPARMS): Return code (OPRC): 1 = Successful and there are more elements on the queue 2 = Successful and this is the last element on the queue 3 = Unsuccessful; the queue is empty</p>		<p>DELEL CLEARO</p> <p>MARKDISP</p> <p>REOBTAIN</p>	2.3.1	<p>Buffer size (OPBUFSE) = Message length Buffer address (OPBFADR) = Address of first byte of message Flags (OPFLAGS) = The element's flags</p> <p>(g) Input parameters (OPARMS): Request Code (OPREQ) = 7</p> <p>Output parameters (OPARMS): None</p> <p>(h) Storage is allocated (GETCELL or GETMAIN) for the element and pointers are manipulated to add the element to the queue. If GETCELL or GETMAIN is unsuccessful, ABEND OAD is issued.</p> <p>Input parameters (OPARMS): Request code (OPREQ) = 8 Buffer size (OPBFSZ) = Message length Flags (OPFLAGS) = Flag settings requested by caller</p> <p>Output parameters (OPARMS): Return code (OPRC): 1 = Successful 2 = Unsuccessful; no room for the element Buffer address (OPBUFADR) = Location of message area of queue element</p>		<p>RESET</p> <p>BLDELE ADDTOP</p>	

MO 4.2 (Part 3 of 3) Output Queue Manager (IKTQMOUT)



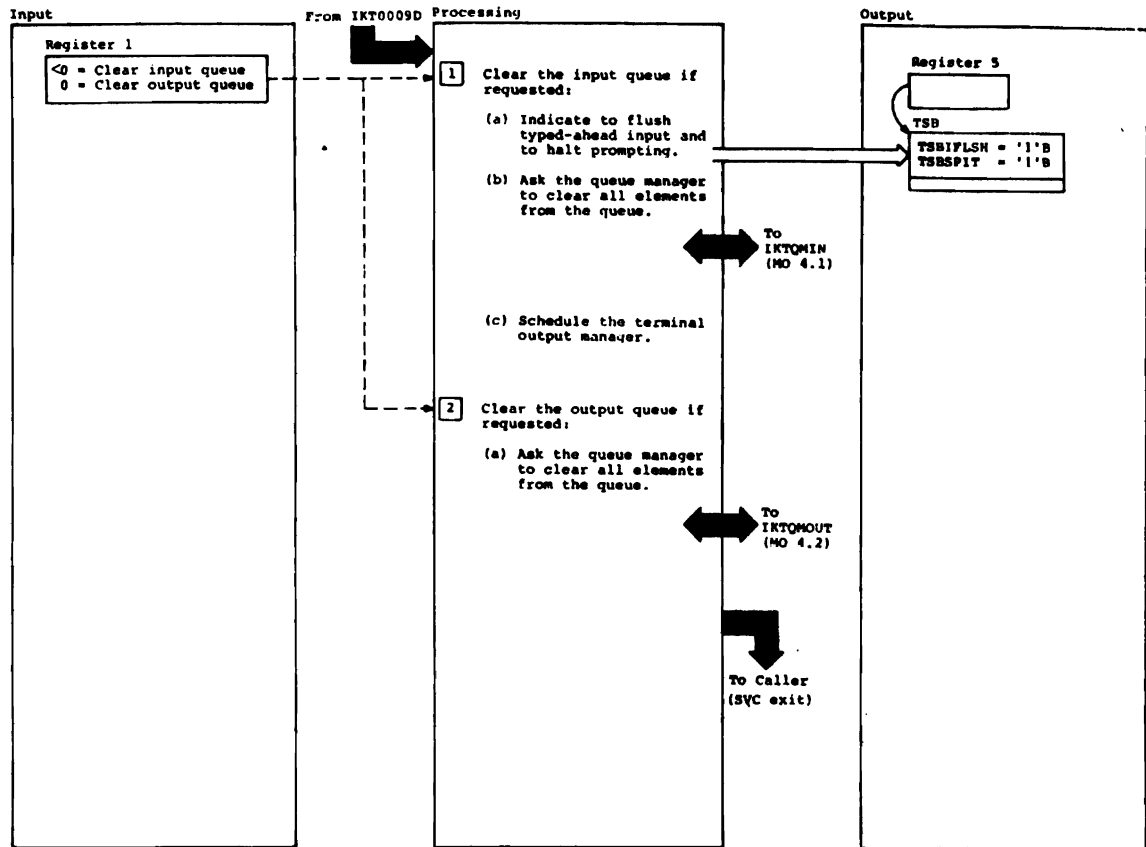
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>2 IKTQMPR1 is the first queue manager FRR. It receives control when the queue manager encounters an error. It builds a queue verification parameter list (QVPL), calls IEAVEQV3 to verify and correct the list, logs the TVMA into the SDWA, and retries the queue manager once. (IEAVEQV3 calls IKTQMEV to ensure that queue element size and buffer size are acceptable. If an unacceptable element is found, message IKT004011 (output data lost) is sent to the terminal.)</p>	IKTQMPR1						
<p>3 IKTQMPR2 is the second queue manager FRR. It receives control when an error occurs during IKTQMPR1 processing. If the element verification routine (IKTQMEV) has failed, the element is marked unacceptable and verification continues. If the queue manager FRR has failed, the TVMA is logged into the SDWA and termination processing continues.</p>	IKTQMPR2						

MO 5.1 Terminal Control Macro Branching Routine (IKT0009D)



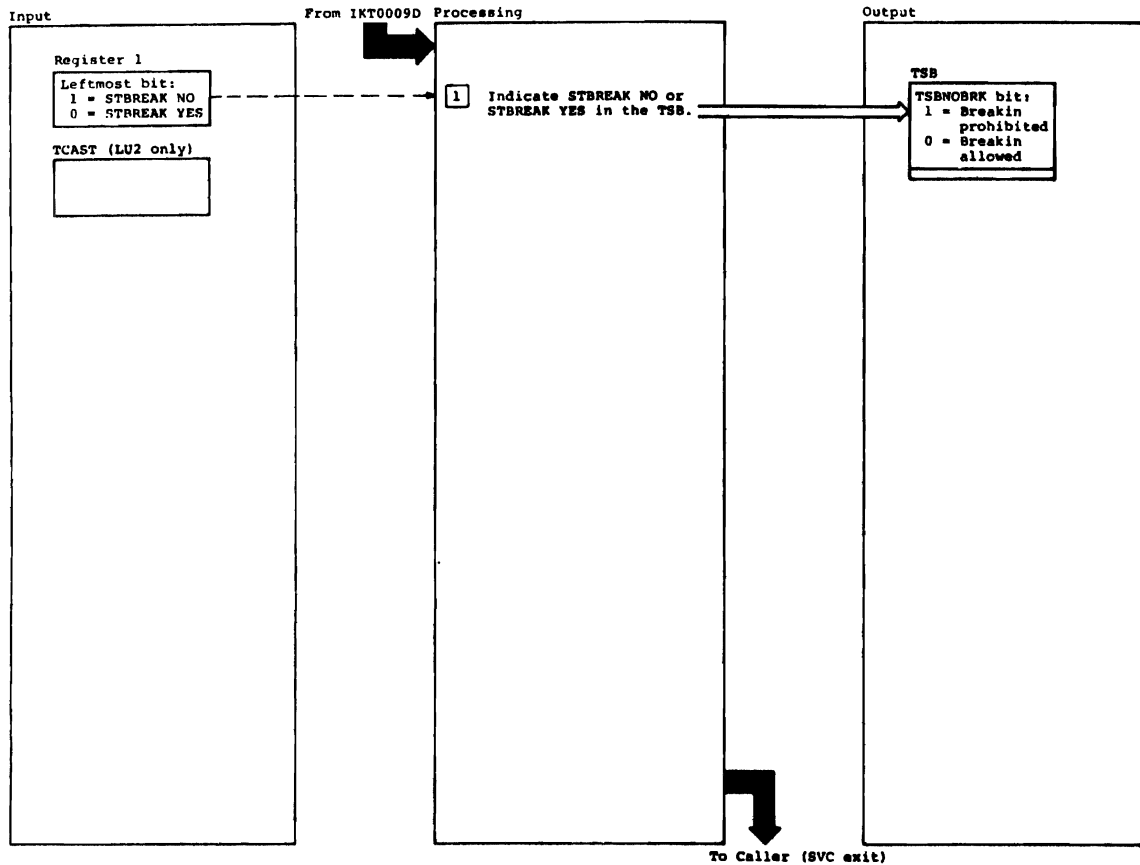
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.																																																												
<p>1 The entry code is in the high-order byte of register 0:</p> <table border="1"> <thead> <tr> <th>Code</th> <th>Module</th> <th>MO</th> </tr> </thead> <tbody> <tr><td>1</td><td>IKT09401</td><td>5.2</td></tr> <tr><td>2</td><td>Invalid</td><td></td></tr> <tr><td>3</td><td>Invalid</td><td></td></tr> <tr><td>4</td><td>IKT09404</td><td>5.3</td></tr> <tr><td>5</td><td>IKT09405</td><td>5.4</td></tr> <tr><td>6</td><td>Invalid</td><td></td></tr> <tr><td>7</td><td>Invalid</td><td></td></tr> <tr><td>8</td><td>Invalid</td><td></td></tr> <tr><td>9</td><td>IKT09409</td><td>5.5</td></tr> <tr><td>10</td><td>IKT0940A</td><td>5.6</td></tr> <tr><td>11</td><td>IKT0940B</td><td></td></tr> <tr><td>12</td><td>IKT0940C</td><td>5.7</td></tr> <tr><td>13</td><td>IKT0940D</td><td>5.8</td></tr> <tr><td>14</td><td>IKT0940E</td><td>5.9</td></tr> <tr><td>15</td><td>IKT0940F</td><td>5.10</td></tr> <tr><td>16</td><td>Invalid</td><td></td></tr> <tr><td>18</td><td>IKT09412</td><td>5.11</td></tr> <tr><td>19</td><td>IKT09413</td><td>5.12</td></tr> <tr><td>20</td><td>IKT09414</td><td>5.13</td></tr> </tbody> </table>	Code	Module	MO	1	IKT09401	5.2	2	Invalid		3	Invalid		4	IKT09404	5.3	5	IKT09405	5.4	6	Invalid		7	Invalid		8	Invalid		9	IKT09409	5.5	10	IKT0940A	5.6	11	IKT0940B		12	IKT0940C	5.7	13	IKT0940D	5.8	14	IKT0940E	5.9	15	IKT0940F	5.10	16	Invalid		18	IKT09412	5.11	19	IKT09413	5.12	20	IKT09414	5.13							
Code	Module	MO																																																																	
1	IKT09401	5.2																																																																	
2	Invalid																																																																		
3	Invalid																																																																		
4	IKT09404	5.3																																																																	
5	IKT09405	5.4																																																																	
6	Invalid																																																																		
7	Invalid																																																																		
8	Invalid																																																																		
9	IKT09409	5.5																																																																	
10	IKT0940A	5.6																																																																	
11	IKT0940B																																																																		
12	IKT0940C	5.7																																																																	
13	IKT0940D	5.8																																																																	
14	IKT0940E	5.9																																																																	
15	IKT0940F	5.10																																																																	
16	Invalid																																																																		
18	IKT09412	5.11																																																																	
19	IKT09413	5.12																																																																	
20	IKT09414	5.13																																																																	

MO 5.2 TCLEARQ Terminal Control Macro Routine (IKT09401)



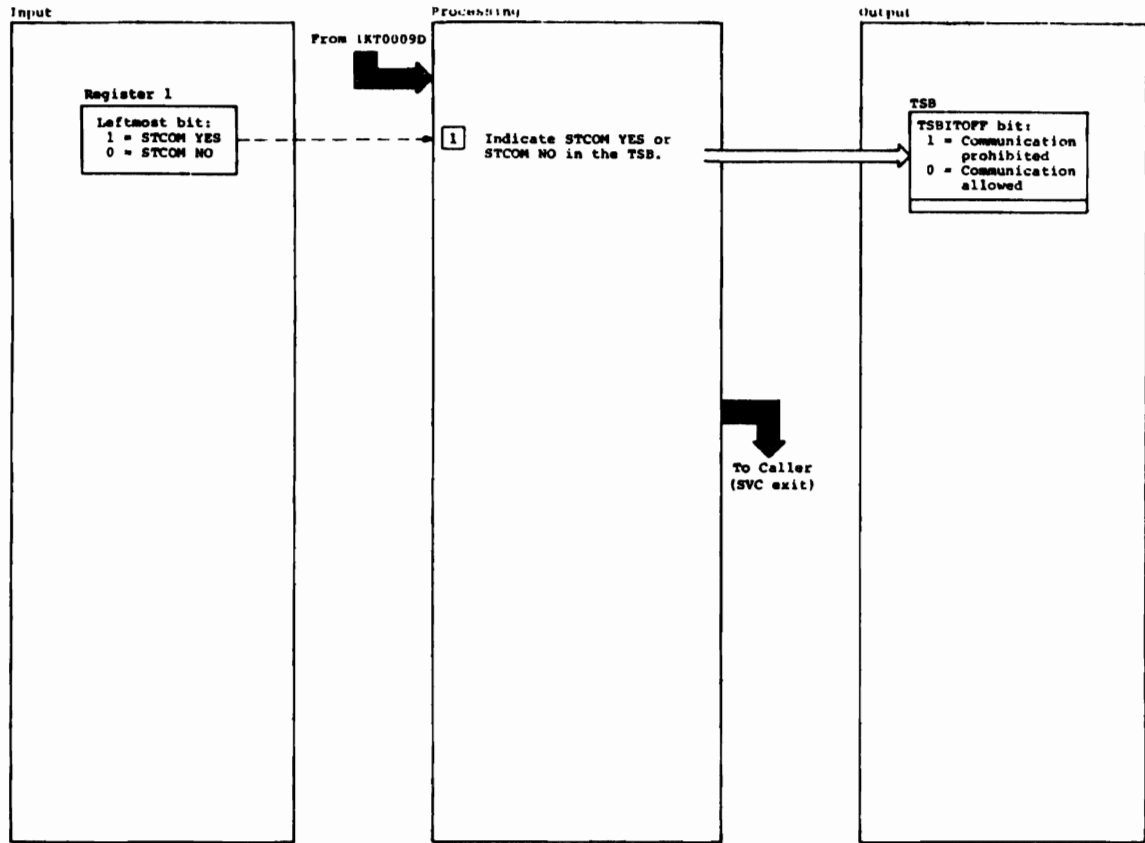
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine services the TCLEARQ macro instruction for TSO/VTAM terminals. It invokes the queue manager to clear the input or output queue.	IKT09401						
1 (b) The following parameter is passed to the input queue manager in IPARMS: Request code (IPREQ) = 5	IKT09401		4.1				
(c) The appropriate terminal output manager is scheduled (SCHEDULE SRB) if it is available (TVWATAS) and not already scheduled (TVWATIS).	IKT09401		3.2 3.4				
2 The output queue is cleared of all but cross-memory messages.	IKT09401						
(a) The following parameter is passed to the output queue manager in OPARMS: Request code (OPREQ) = 5	IKT09401						
Return codes: 0 Successful 4 Register 1 at entry contained a value greater than zero.							

MO 5.3 STBREAK Terminal Control Macro Routine (IKT09404)



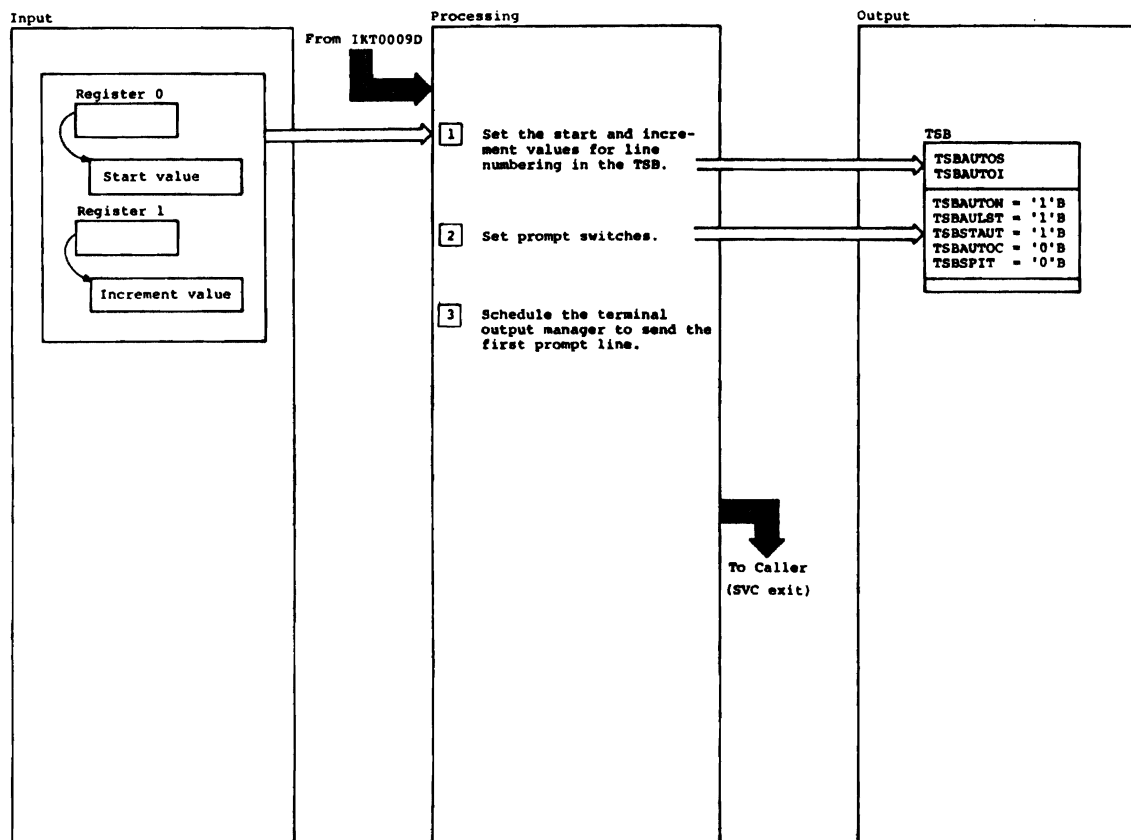
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine services the STBREAK macro instruction for TSO/VTAM terminals. It sets a bit in the TSB, depending on the operand specified with STBREAK, to indicate whether TSO is allowed to interrupt input transmission.</p> <p>1 The TCAS key (key 6) is acquired (MODESET) for access to the TSB. If the terminal user is no longer connected to TSO, control returns immediately to the caller. If the terminal is a 3270 (TSBXMTMP), the use of STBREAK is invalid and the TSB is not set.</p> <p>If TCASMSW = 1, mode switching is allowed for LU2 terminals.</p> <p>Return codes: 0 Successful 4 Register 1 at entry contained a nonzero value in bits 1-31 8 STBREAK was issued for a 3270 terminal</p>	IKT09404						
	IKT09404						

MO 5.4 STCOM Terminal Control Macro Routine (IKT09405)



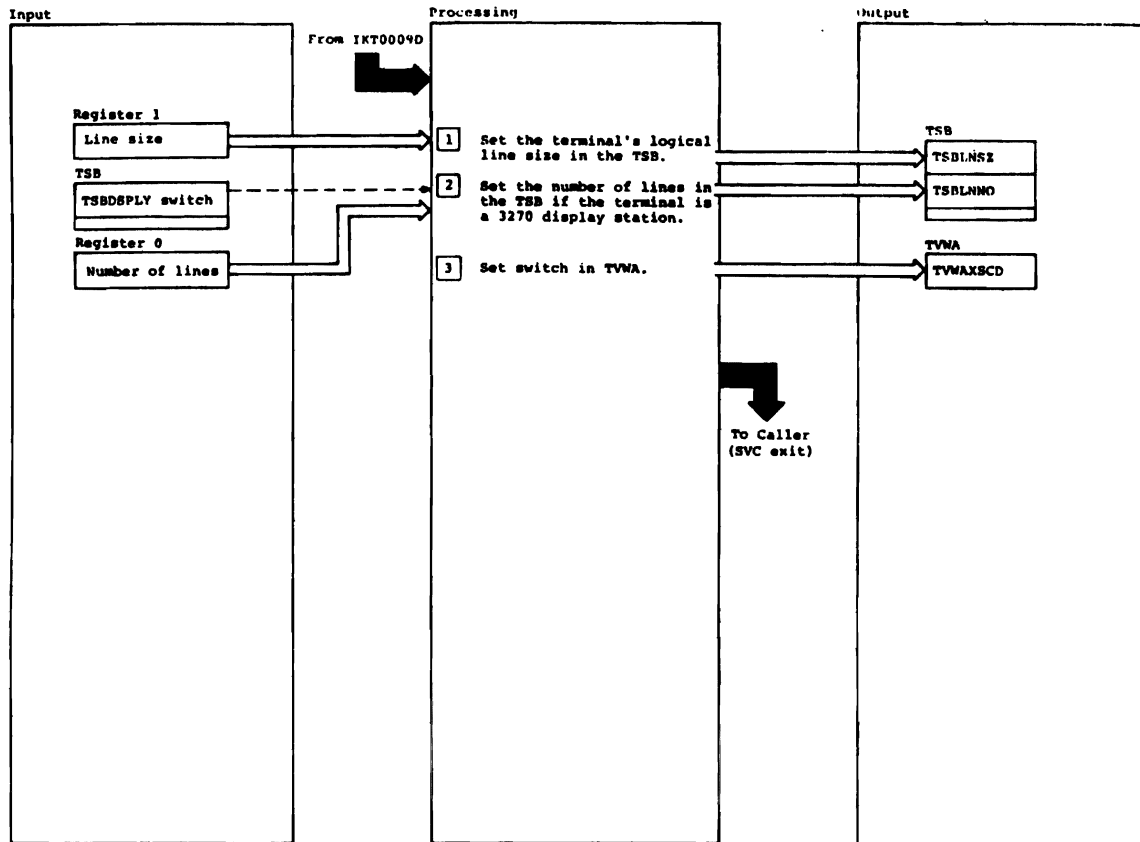
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine services the STCOM macro instruction for TSO/VTAM terminals. It sets a bit in the TSB, depending on the operand specified with STCOM, to indicate whether a terminal is allowed to accept messages from other terminals.	IKT09405						
<p>1 The TCAS key (key 6) is acquired (MODESET) for access to the TSB.</p> <p>Return codes: 0 Successful 4 Input parameter error</p>	IKT09405						

MO 5.5 STAUTOLN Terminal Control Macro Routine (IKT09409)



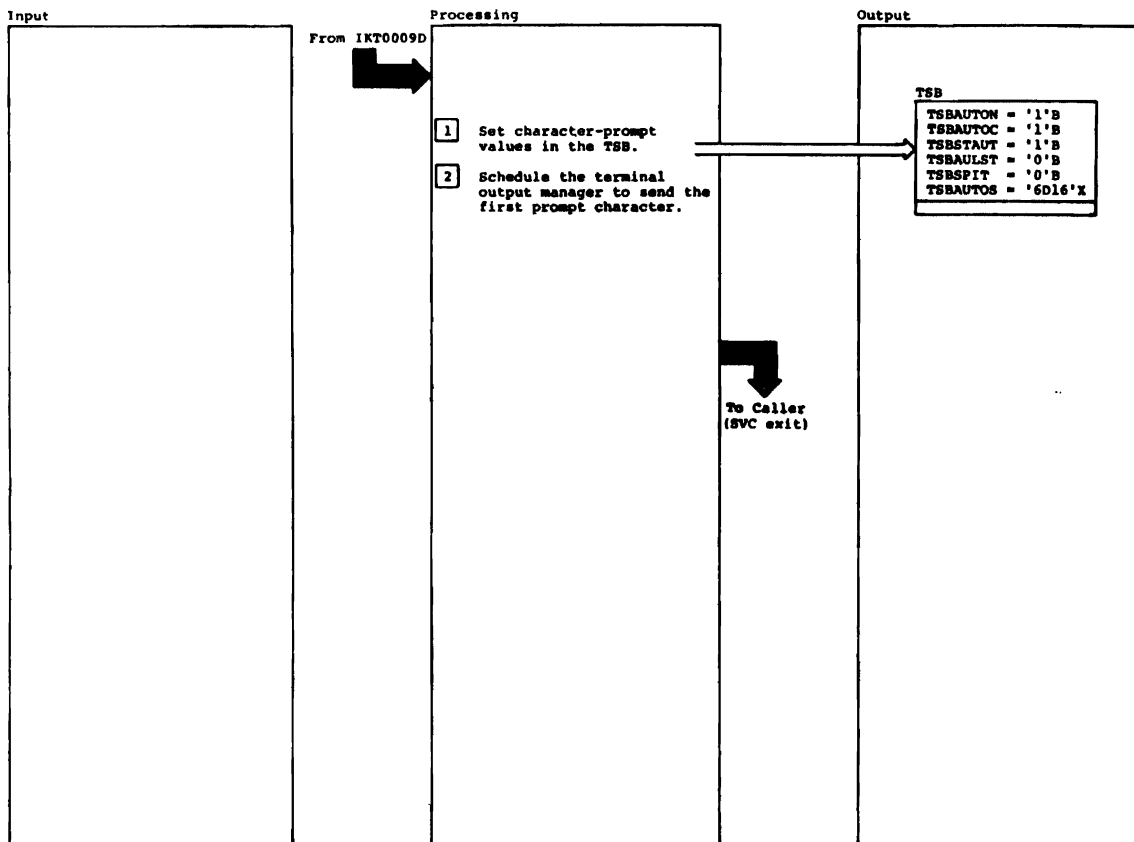
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine services the STAUTOLN macro instruction for TSO/VTAM terminals. It sets values in the TSB, according to the operands specified with STAUTOLN, that are used in providing automatic line numbering.	IKT09409						
1 If the start and increment values are not between 0 and 99,999,999, control returns immediately to the caller.	IKT09409						
3 The appropriate terminal output manager is scheduled (SCHEDULE SRB) if it is available (TVWATAS) and not already scheduled (TVWATIS).	IKT09409		3.2 3.4				
Return codes: 0 Successful 4 Start or increment value not within allowable limits							

MO 5.6 STSIZE Terminal Control Macro Routine (IKT0940A)



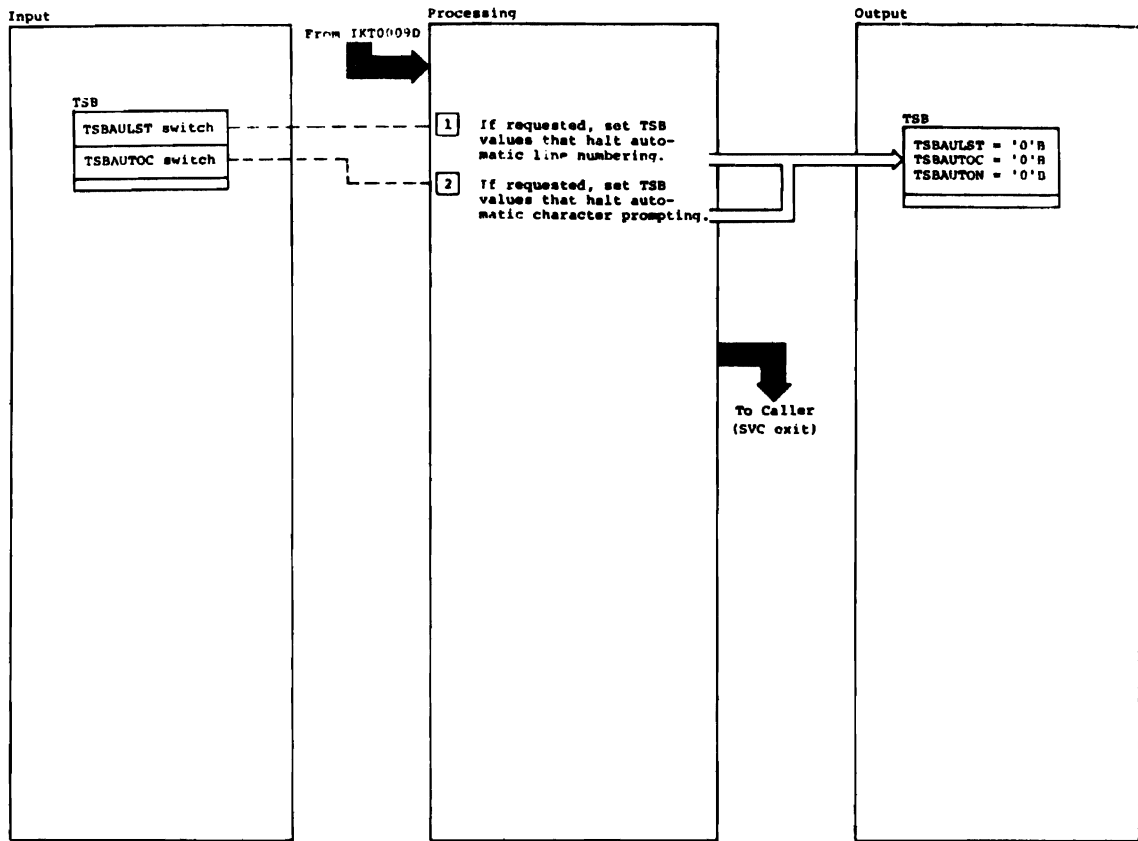
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine services the STSIZE macro instruction for TSO/VTAM terminals. It sets values in the TSB, depending on the operands specified with STSIZE, to indicate the logical line size and, if a display station, the number of lines on the screen.</p>	IKT0940A						
<p>1 The TCAS key (key 6) is acquired (MODESET) to access the TSB.</p>	IKT0940A						
<p>2 Nonstandard screen dimensions are allowed.</p>	IKT0940A	DISPROC					
<p>Return codes (in hex):</p> <p>0 Successful</p> <p>4 Invalid parameter</p> <p>8 Screen size not specified or line size not specified for 3270</p> <p>C Nonstandard screen size specified</p>							
<p>3 TSB screen dimensions may have changed. The output manager should check the screen size to ensure compatibility with the terminal.</p>		DISPROC					

MO 5.7 STAUOCP Terminal Control Macro Routine (IKT0940C)



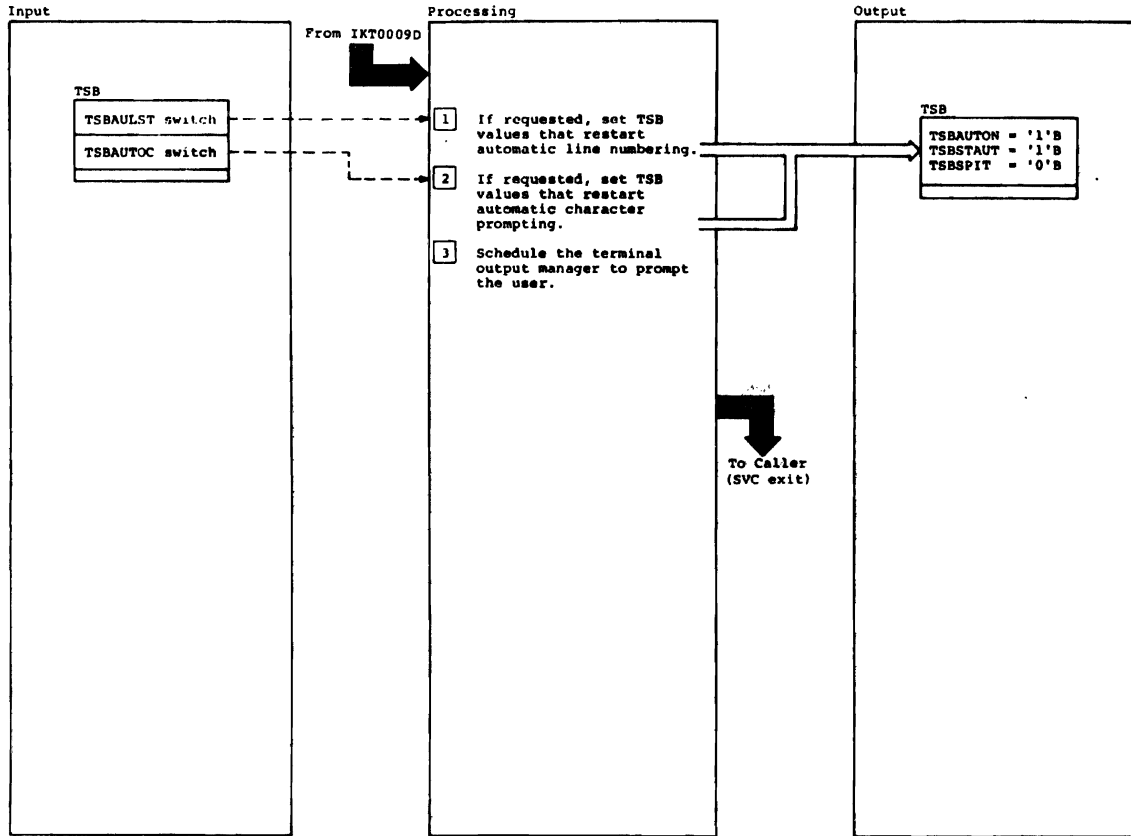
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine services the STAUOCP macro instruction for TSO/VTAM terminals. It sets values in the TSB to start automatic character prompting.	IKT0940C						
1 If register 1 at entry is not zero, or if the terminal is a display terminal (TSBDSPLY), the TSB is not set.	IKT0940C						
2 The appropriate terminal output manager is scheduled (SCHEDULE SRB) if it is available (TVMATAS) and not already scheduled (TVWATIS).	IKT0940C		3.2 3.4				
Return codes: 0 Successful 4 A parameter was specified in register 1 but none was expected							

MO 5.8 SPAUTOPT Terminal Control Macro Routine (IKT0940D)



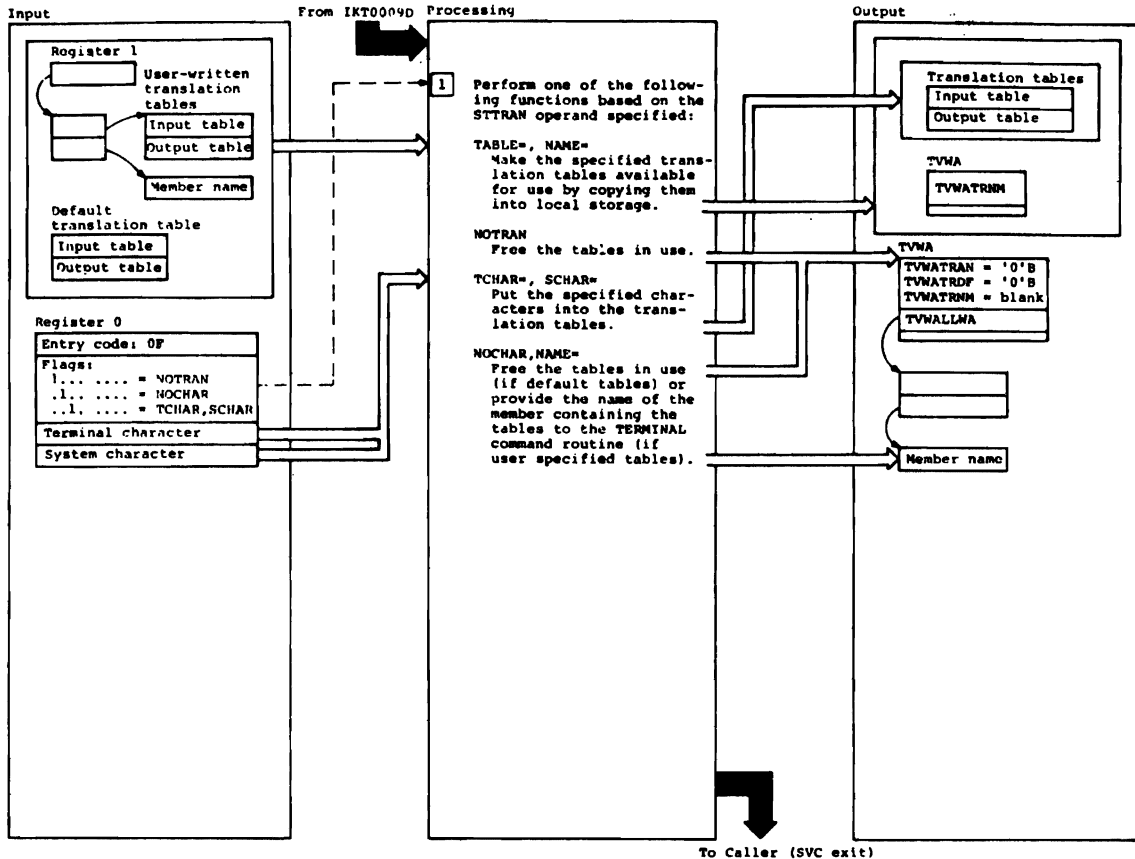
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>This routine services the SPAUTOPT macro instruction for TSO/VTAM terminals. It sets values in the TSB to stop automatic line numbering or automatic character prompting.</p> <p>1,2 The TCAS key (key 6) is acquired (MODESET) to access the TSB.</p> <p>Return codes: 0 Successful 4 A parameter was specified in register 1 but none was expected 8 Neither line numbering nor character prompting was in use</p>	IKT0940D						
	IKT0940D						

MO 5.9 RTAUTOPT Terminal Control Macro Routine (IKT0940E)



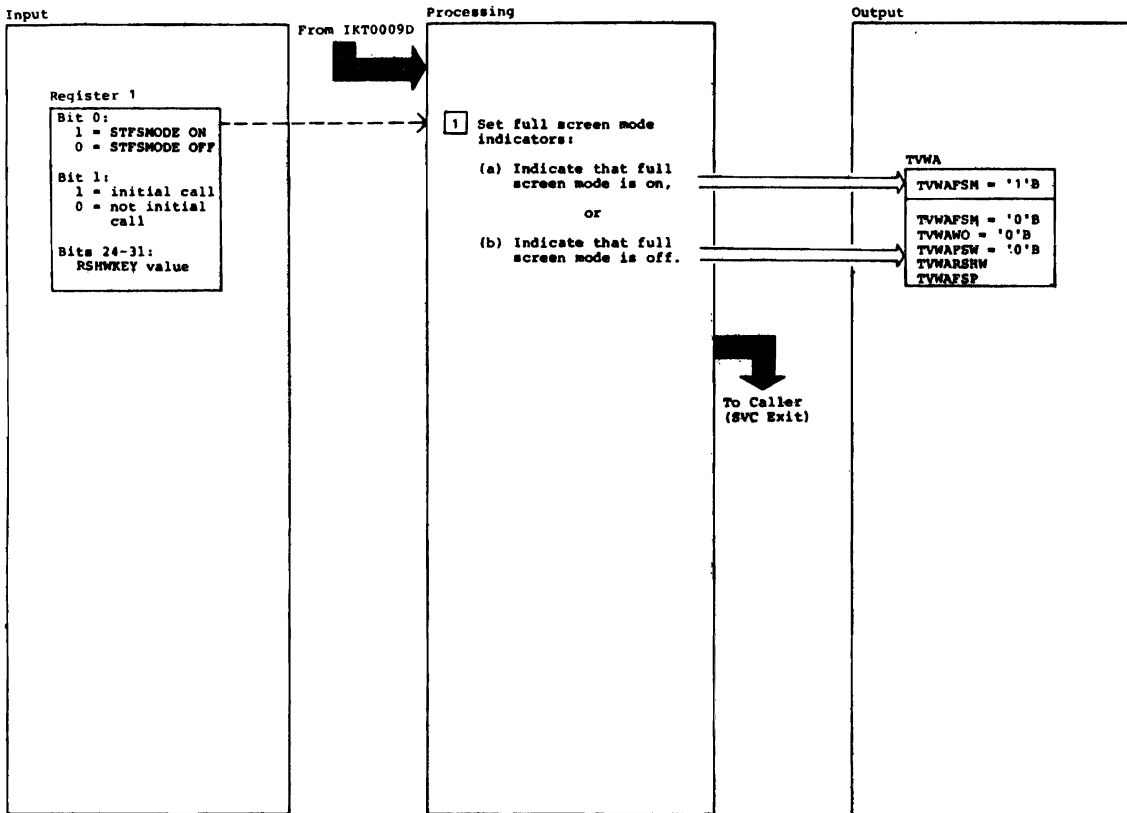
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
This routine services the RTAUTOPT macro instruction for TSO/VTAM terminals. It sets values in the TSB to restart automatic line numbering or automatic character prompting.	IKT0940E						
3 The appropriate terminal output manager is scheduled (SCHEDULE SRB) if it is available (TVWATAS) and not already scheduled (TVWATIS).	IKT0940E		3.2 3.4				
Return codes: 0 Successful 4 A parameter was specified in register 1 but none was expected 8 Prompting was not in use							

MO 5.10 STTRAN Terminal Control Macro Routine (IKT0940F)



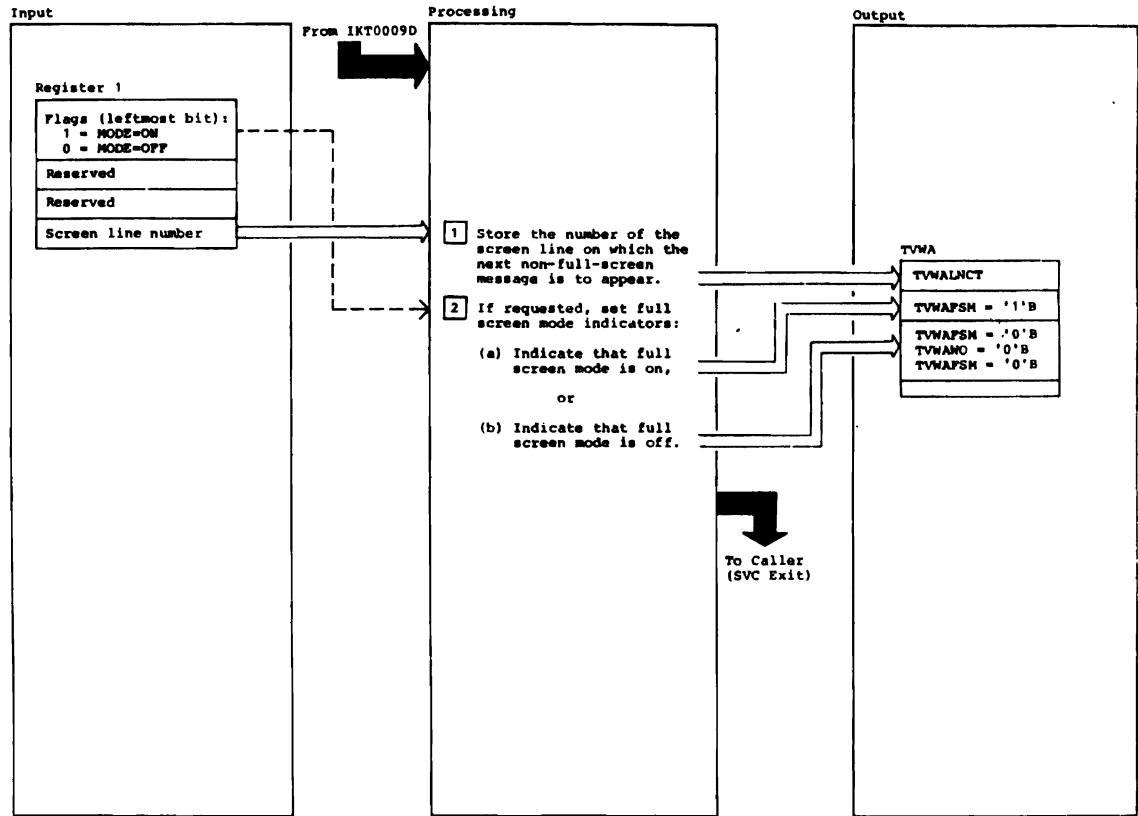
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
This routine services the STTRAN macro instruction for TSO/VTAM terminals. It sets up, modifies, or discontinues the use of translation tables, depending on the operands specified with STTRAN.	IKT0940F			If NOCHAR and NAME are specified, and if default tables are in use, FREEMAIN frees the tables, and translation indicators in the TVWA are set to turn off translation. If user-written tables are in use, the name of the member containing the tables is passed to the TERMINAL command routine.		CLEARTAB	
1 If TABLE and NAME are specified, the name of the member that contains the translation tables is put into the TVWA, storage is allocated (GETMAIN) locally for the tables, and the pair of tables (default or user-written, depending on TVWATRDF) is copied into the local storage. (Default tables are coded in IKT0940F. They translate each character to itself. They are provided to allow TCHAR and SCHAR to be specified without having previously specified TABLE and NAME.)	IKT0940F	GETTAB		Return codes: 0 Successful 4 NOTRAN or NOCHAR was specified but translation was not in effect. 8 TABLE or NOCHAR was specified but an address was not provided with the NAME operand. C Internal error - an unidentifiable flag was set in input register 0.			
If NOTRAN is specified, FREEMAIN frees the tables in use, and translation indicators in the TVWA are set to turn off translation.		CLEARTAB					
If TCHAR and SCHAR are specified, default translation tables are provided if necessary, and the translation characters are put into the translation tables.		GETTAB					

MO 5.11 STFSMODE Terminal Control Macro Routine (IKT09412)



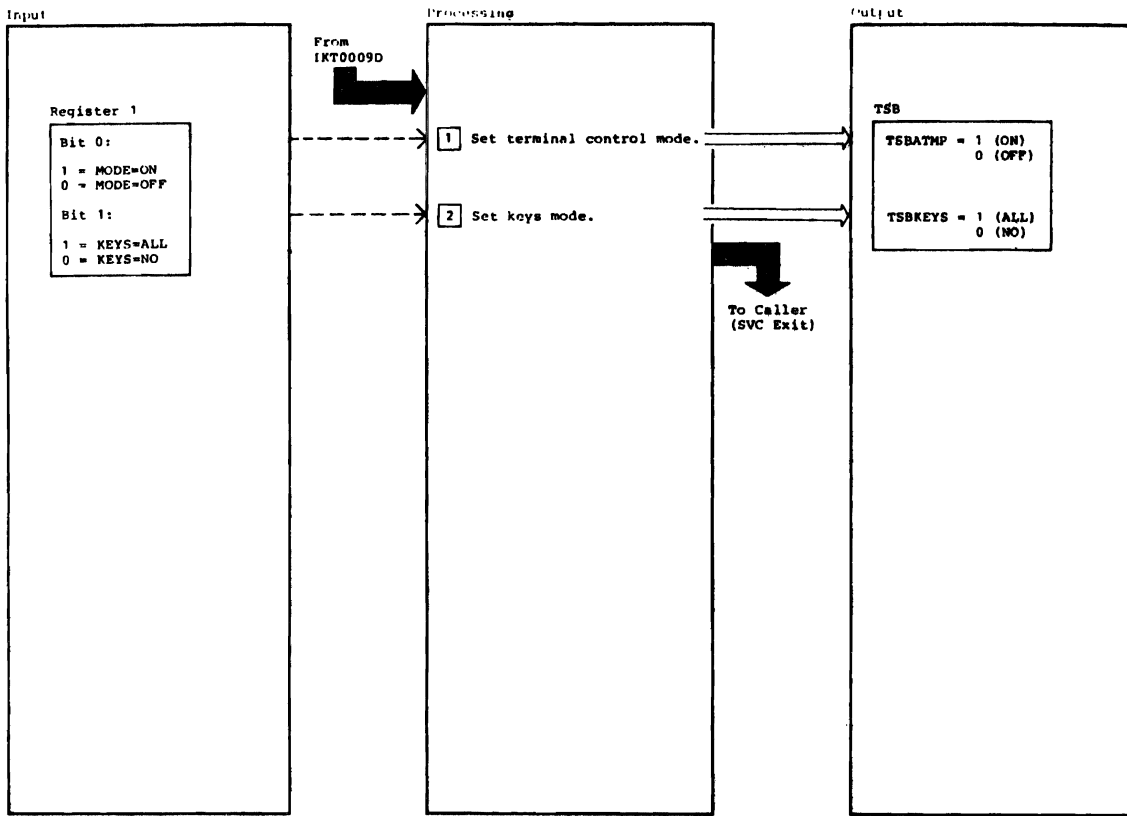
Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
This routine services the STFSMODE macro instruction for TSO/VTAM terminals. It sets bits in the TVMA, depending on the operand specified with STFSMODE, to indicate whether an IBM 3270 display terminal is to operate in full screen mode.	IKT09412						
1 The line counter (TVWALNCT) is set to 1. TVWARSHW is reset to the system default value (PA2).	IKT09412						
Return codes: 0 Successful. 4 Invalid parameter specified to the SVC. 8 Invalid terminal type. This macro instruction is valid only for IBM 3270 display terminals that use TSO/VTAM.							

MO 5.12 STLINENO Terminal Control Macro Routine (IKT09413)



Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine services the STLINENO macro instruction for TSO/VTAM terminals. It sets values in the TVWA for IBM 3270 display terminals, according to the operands specified with STLINENO: it sets the number of the screen line on which the next non-full-screen message should appear, and it indicates whether the terminal is to operate in full screen mode.</p> <p>Return codes: 0 Successful. 4 Invalid parameter specified to the SVC. 8 Invalid terminal type. This macro instruction is valid only for IBM 3270 display terminals that use TSO/VTAM. C The line number specified was 0 or it was greater than the maximum number of lines allowed for the terminal in use.</p>	IKT09413						

MO 5.13 STTMPMD Terminal Control Macro Instruction (IKT09414)



Notes	Routine	Label	Ref.	Notes	Routine	Label	Ref.
<p>This routine processes the STTMPMD macro for TSO/VTAM terminals. It sets bits in the TSB to indicate whether the terminal control routine is active for this terminal and whether the PAL and CLEAR keys are to be passed to the application program as data.</p> <p>1 If the terminal control routine is active, set TSBATMP to 1; if not, set TSBATMP to 0.</p> <p>2 If the PAL and CLEAR keys are to be passed, set TSBKEYS to 1; if not, set TSBKEYS to 0.</p> <p>Return codes:</p> <p>0 Successful.</p> <p>4 Invalid parameter was specified to SVC.</p> <p>8 Invalid terminal type was specified; not a display.</p>							
	IKT09414						
	IKT09414						

VTIOC PROGRAM ORGANIZATION

This section describes VTIOC modules and shows control flow among them. There are two parts:

- Figures (Figures 4, 5, 6, 7, 8, and 9) showing control flow among the VTIOC modules
- A description of each VTIOC module giving:

Names

MO Diagram - that describes the module

Function

Entry from

Registers at Entry - contents of relevant registers

Exit to

Registers at Exit - contents of relevant registers

External References

Control Blocks Used

Mapping Macros Used

Executable Macros Used

Module Attributes

Lock Dependency - requirements at module entry

Messages - detected by and issued from each module

Abend Codes

Data Sets

The module descriptions are arranged alphanumerically by module name.

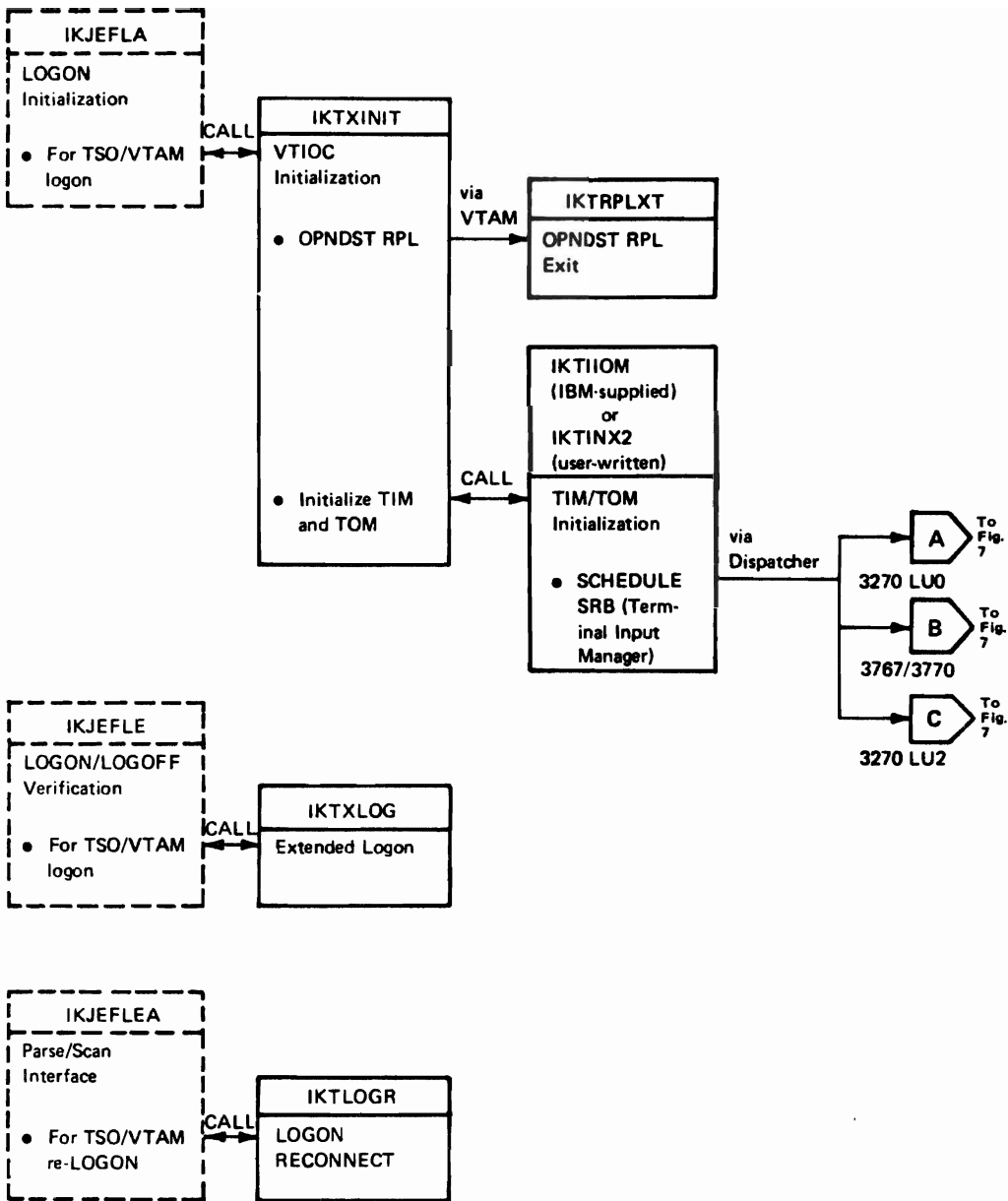


Figure 4. VTIOC Initialization Module Flow

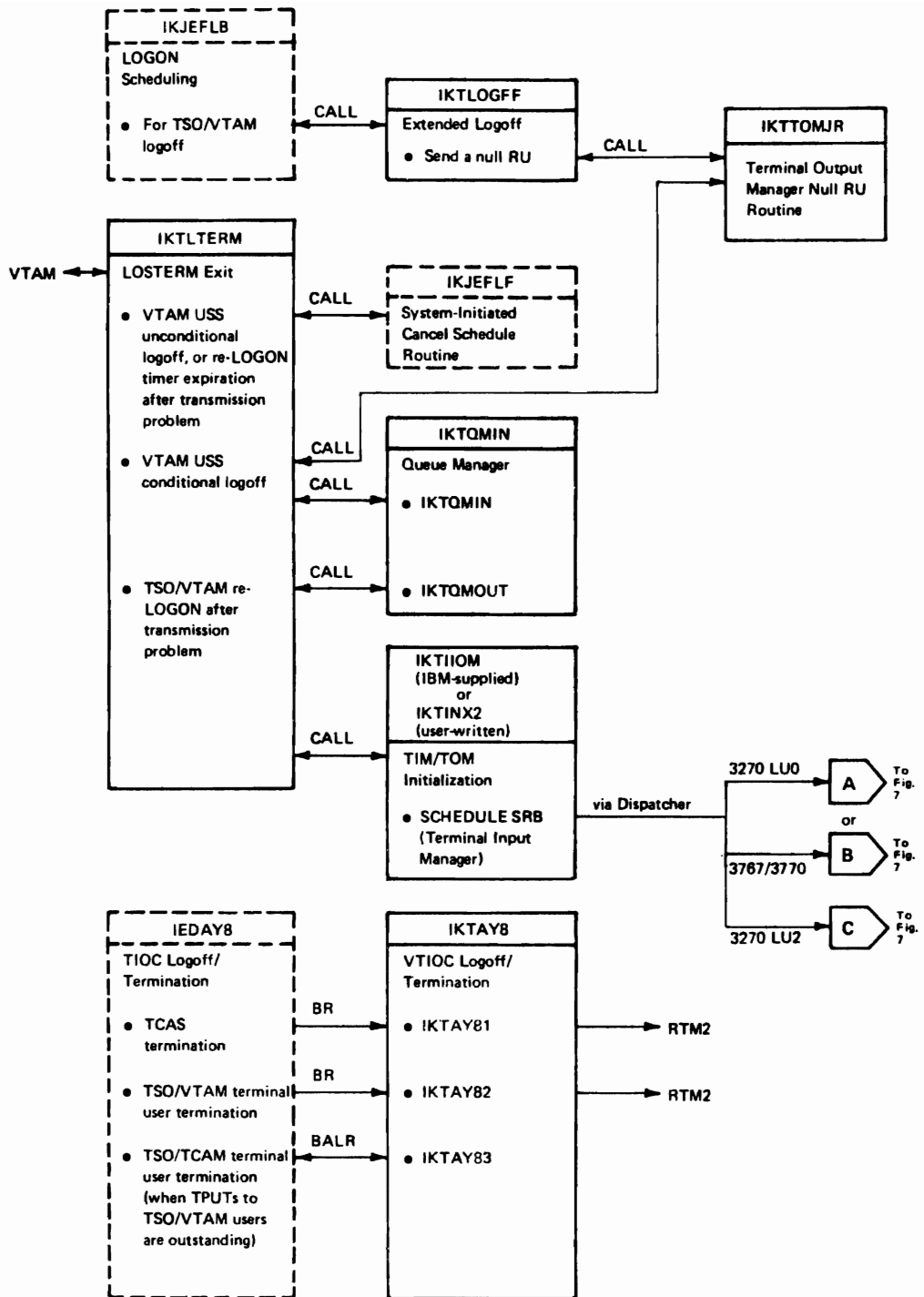


Figure 5. VTIOC Termination Module Flow

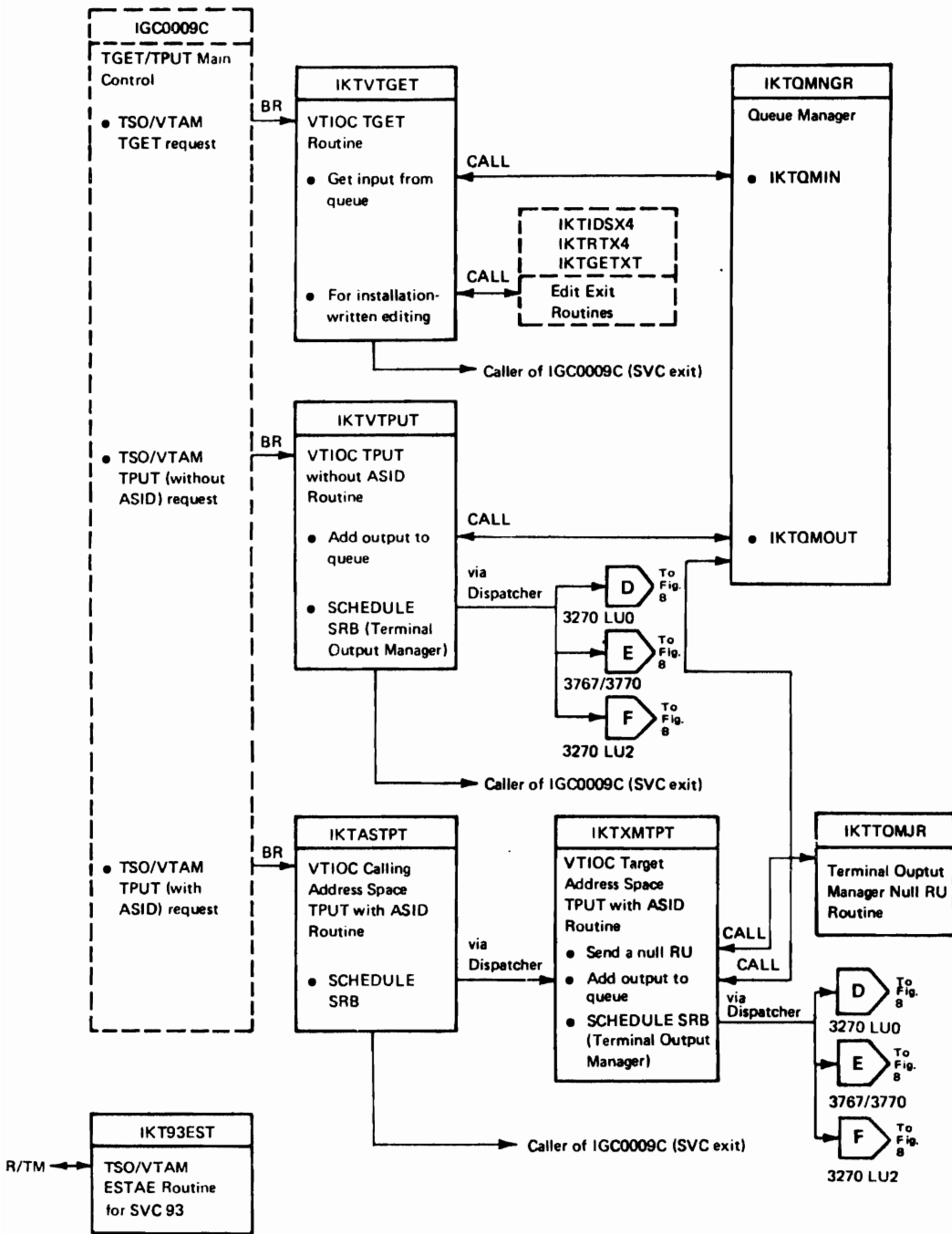


Figure 6. VTIOC TGET/TPUT Module Flow

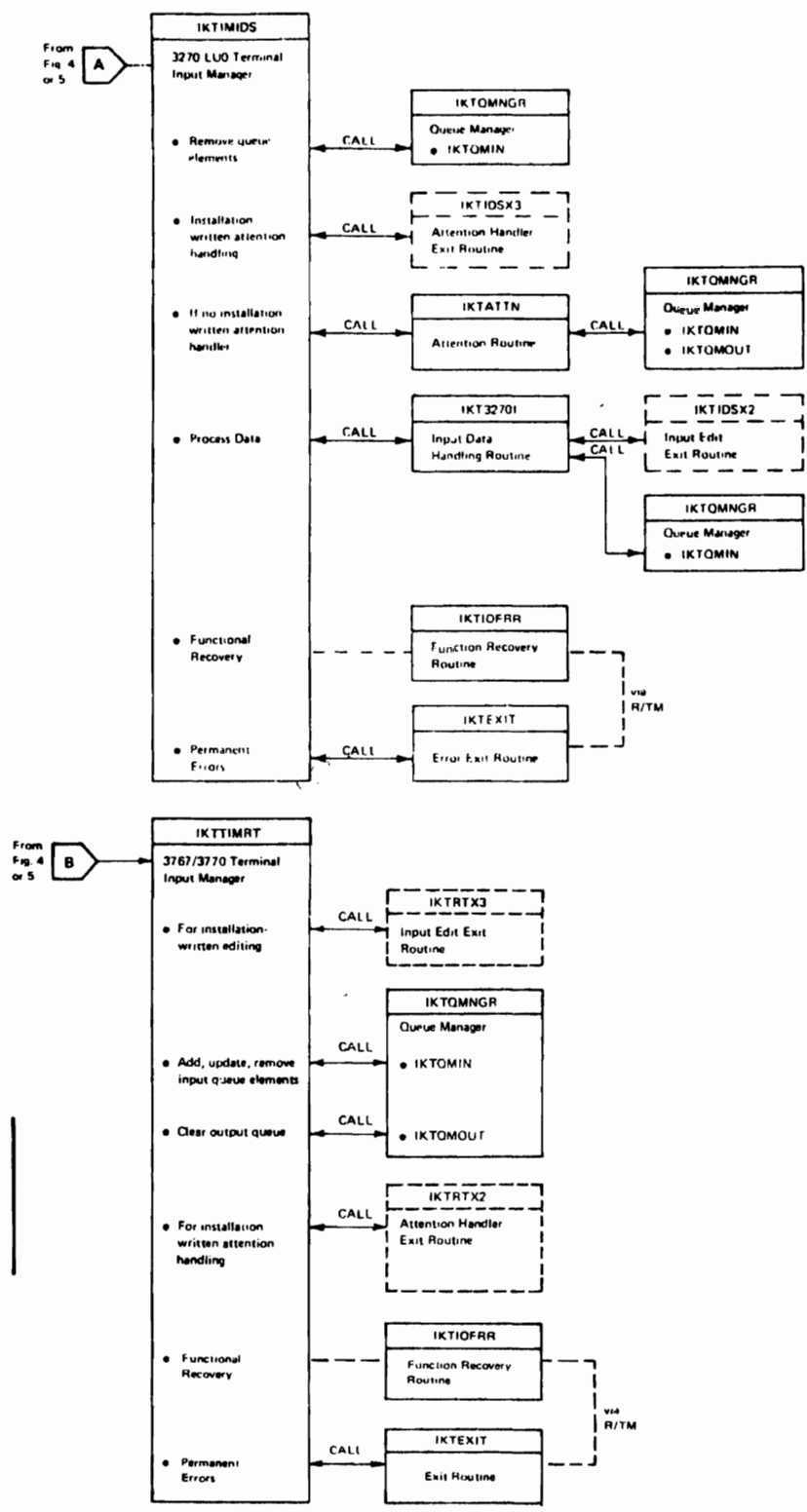


Figure 7. VTIOC Terminal Input Manager Module Flow (part 1 of 2)

From Fig. 4 or 5

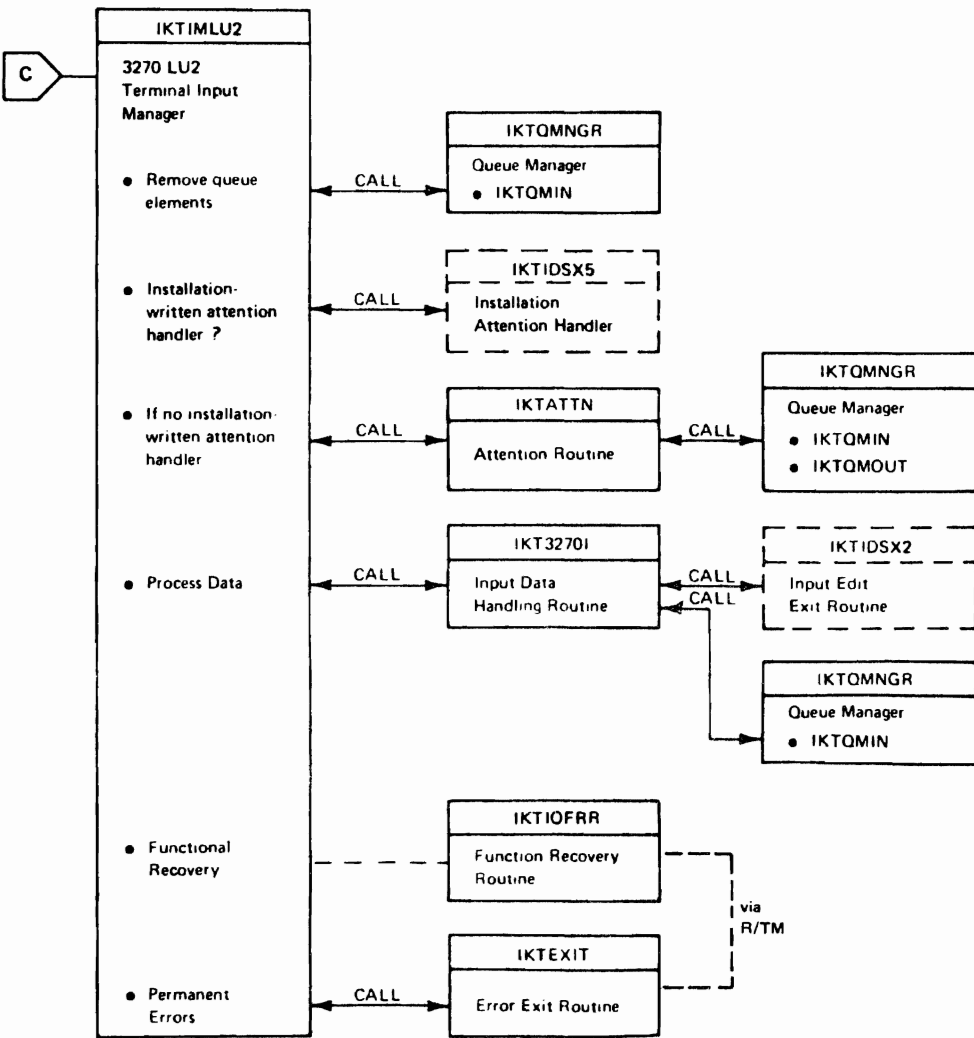


Figure 7. VTIOC Terminal Input Manager Module Flow (part 2 of 2)

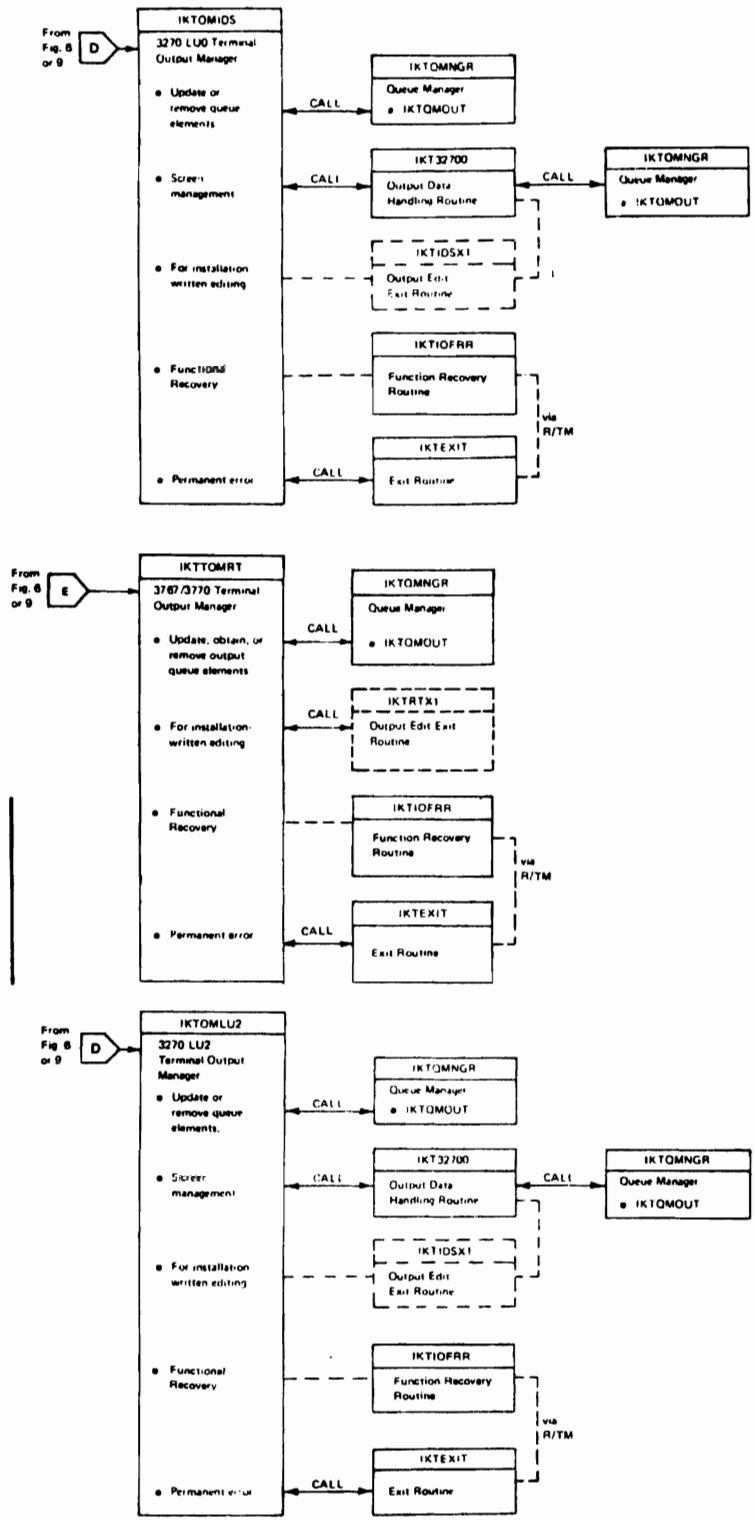


Figure 8. VTIOC Terminal Output Manager Module Flow

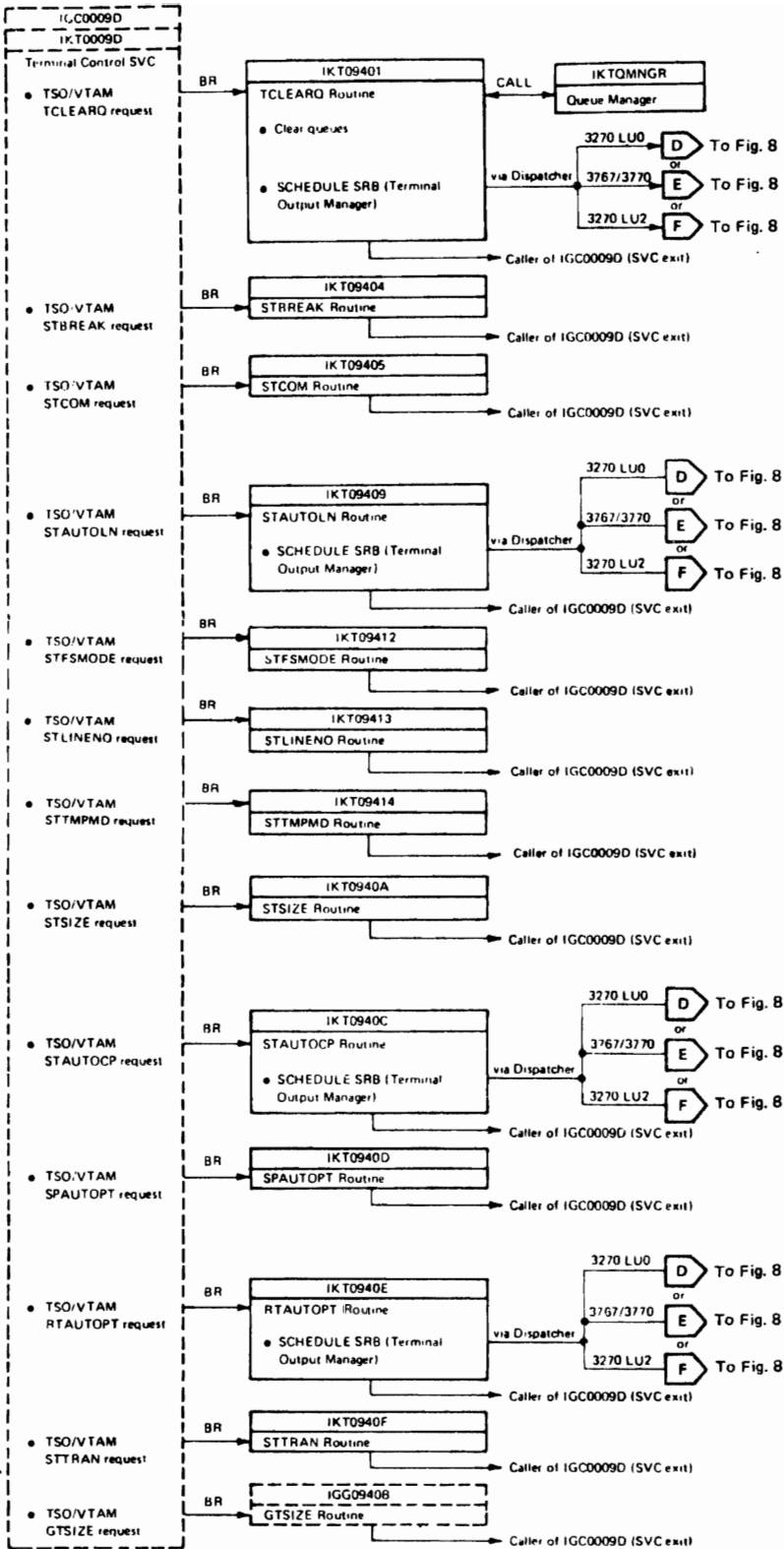


Figure 9. VTIOC Terminal Control Macros Module Flow

IKTASCII -- ASCII-EBCDIC Translation Tables

Names:

Assembly Module: IKTASCII
Object Module: IKTASCII
Alternate Entry Points: None
Load Module: IKTASCII

MC_Diagram: None

Function: This module consists of translation tables for 3270 and 3767/3770 terminals that are used by the I/O managers to translate input from terminals with ASCII keyboards from ASCII code to EBCDIC code, and output from EBCDIC code to ASCII code.

Entry_from: N/A

Registers_at_Entry: N/A

Exit_to: N/A

Registers_at_Exit: N/A

External_References: None

Control_Blocks_Used: None

Mapping_Macros_Used: None

Executable_Macros_Used: None

Module_Attributes: Not executable

Lock_Dependency: None

Messages: None

Abend_Codes: None

Data_Sets: None

IKIASIPT -- Calling Address Space TPUT with ASID Routine

Names:

Assembly Module: IKIASIPT
Object Module: IKIASIPT
Alternate Entry Points: None
Load Module: IGC0009C

MC_Diagram: 2.3

Function: This routine handles TSO/VTAM TPUT with ASID requests for the calling address space. It takes data from the TPUT requester's buffer, edits it, and places it in the TVCS for handling by IKTXMTPT.

Entry_from: IGC0009C

Registers_at_Entry:

Register 3: Address of CVT

Register 4: Address of TCB
Register 5: Address of SVRE
Register 8: Address of ASCE
Register 9: Address of target TSB

Exit_to: Caller (SVC exit)

Registers_at_Exit:

Register 15: Return code (in hex):

00 Successful.
04 A terminal output buffer was not available or ENQ resources were not available, and TPUT NOWAIT was specified.
0C NOINTERCOM (on PROFILE command) was specified by the target terminal operator.
14 The terminal could not be reached.

External_References: IKTXMTPT

Control_Blocks_Used: ASCB, ASVT, CVT, PSA, RB, SRB, TCAST, TCB, TCT, TSE, TSBX, TVCS, WSAVT, XSA

Mapping_Macros_Used: CVT, IEEXSA, IEFTCT, IHAASCB, IHAASVT, IHAPSA, IHARE, IHASRB, IHAWSAVT, IKJTCB, IKJTSB, IKTTCAST, IKTTVCS

Executable_Macros_Used: DEQ, ENQ, FREEMAIN, GETMAIN, MODESET, SCHEDULE, SETLOCK, TESTAUTH, WAIT

Module_Attributes: Reenterable, privileged, enabled, key 0

Lock_Dependency: This module is entered with the local lock held.

Messages: None

Abend_Codes: None

Data_Sets: None

IKTATTN -- Attention Handling Routine

Names:

Assembly Module: IKTATTN
Object Module: IKTATTN
Alternate Entry Points: Ncne
Load Module: IKTIOM00

MC_Diagram: 3.8

Function: IKTATTN handles attention and attention-ignored conditions.

Entry_from: IKTIMLU2 or IKTIMIDS

Registers_at_Entry:

Register 13: Address of a save area

Exit_to:

Caller: Normal

Registers at Exit: N/A

External References: IKTQMIN, IKTQMOUT, POST, STATUS

Control Blocks Used: ASCB, CVT, PSA, SCVT, TCAST, TIMWA, TSE, TSBX, TVWA

Mapping Macros Used: CVT, IHAASCB, IHAPSA, IHASCVT, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST, IKTTIMWA

Executable Macros Used: None

Module Attributes: Reenterable

Lock Dependency: The calling routine must hold the local lock.

Messages: None

Abend Codes: None

Data Sets: None

IKTAY8 -- VTIOC Logoff/Termination Resource Manager

Names:

Assembly Module: IKTAY8
Object Module: IKTAY81
Alternate Entry Points: IKTAY82, IKTAY83
Load Module: IGC0001C

MC Diagram: 1.6, 1.7, 1.8

Function: This routine handles TCAS termination and TSO/VTAM terminal user address space termination, and cleans up TPUT with ASID requests to a TSO/VTAM address space from a terminating TSO/TCAM address space.

Entry from:

IEDAY8 (in load module IGC0001C) to one of the following entry points:

IKTAY81: When TCAS is terminating
IKTAY82: When a TSO/VTAM terminal user's address space is terminating
IKTAY83: When a TSO/TCAM terminal user's address space is terminating and ASID TPUTs to TSO/VTAM address spaces are outstanding (QTIP 29 equivalent)

Registers at Entry:

Tc entry point IKTAY81:

Register 4: Address of TCAST
Register 8: Address of TSE
Register 13: Address of R/TM2 register save area
Register 15: Address of entry point IKTAY81

Tc entry point IKTAY82:

Register 4: Address of TCAST
Register 8: Address of TSE
Register 11: Address of RMPL

Register 13: Address of R/TM2 register save area
Register 15: Address of entry point IKTAY82

Ic entry point IKTAY83:

Register 4: Address of TCAST
Register 8: Address of TSB
Register 11: Address of RMPL
Register 14: Address of return point in IEDAY8
Register 15: Address of entry point IKTAY83

Exit_to:

R/TM2: After entry at IKTAY81
R/TM2: After entry at IKTAY82
IEDAY8: After entry at IKTAY83

Registers_at_Exit:

Register 15: Return code:

0 Successful

External References: IEDAYTPC, POST macro routine, STATUS macro routine

Control Blocks Used: ASCB, CVT, PSA, RMPL, SCVT, SRB, TCAST, TCX, TIOCRPT, TSB, TSBX, TVCS, WSAVT

Mapping Macros Used: CVT, IHAASCB, IHAPSA, IHARMPL, IHASCVT, IHASRB, IHAWSAVT, IKJTI0CP, IKJTSB, IKITCAST, IKTTVCS, IKTTCXD

Executable Macros Used: GETMAIN, FREEMAIN, LOAD, POST, PURGEDEQ, SETLOCK, SPOST

Module Attributes: Reenterable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTEXTIT -- Exit Routine for Clean Up

Names:

Assembly Module: IKTEXTIT
Object Module: IKTEXTIT
Alternate Entry: Ncne
Load Module: IKTIOM00

MC Diagram: 3.7

Function: IKTEXTIT frees work areas for the calling routines and frees the SREs used by either IKTOMIDS or IKTOMLU2.

Entry from: IMTIMIDS, IKTIMLU2, IKTOMIDS, IKTOMLU2

Registers at Entry:

Register 1: contains a pointer to a list of parameters.
These parameters are the return address, work area length
and total TIM/TOM work area length.

Exit to: Caller

Registers at Exit: N/A

External References: POST macro routine, STATUS macro routine

Control Blocks Used: ASCB, CVT, PRRS, PSA, SRB, TSB, TSBY, TVWA

Mapping Macros Used: CVT, IHAASCB, IHAPRRS, IHAPSA, IHASRB, IKJTSB,
IKTTVWA

Executable Macros Used: FREEMAIN, SETFRR, SETLOCK

Module Attributes: Reenterable

Lock Dependency: The local lock is held by the calling routine.

Messages: None

Abend Codes: None

Data Sets: None

IKTGETXT -- Installation-Written Mcnsupported Terminal Edit Routine

Names:

Object Module: IKTGETXT
Load Module: IGCC009C

Function: An installation must write this routine if it wants to use
a terminal not supported by TSO/VTAM. IKTVTGET will call 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.

Entry from: IKTVTGET

Registers at Entry:

Register 0: Address of extended save area (XSA)
Register 1: Address of input queue manager parameter list (IPARMS)
Register 13: Address of register save area (serialized by the local
lock; IKTGETXT should not release the lock if the save
area is used)
Register 14: Return address
Register 15: Entry point address

Exit to: IKTVTGET

Registers at Exit: Restore all registers to the values they had at
entry.

Module Attributes: Reenterable

IKTIDSX1 -- Installation-Written Output Edit Exit Routine for IBM 3270 Terminals

Names:

Object Module: IKTIDSX1
Load Module: IKTIOM02

Function: An installation may write this routine to perform 3270 output editing in place of or in addition to that performed by the IBM-supplied routine IKT32700. The IBM-supplied routine scans data, edits it according to the TPUT operands specified (EDIT, ASIS, CONTROL, or FULLSCR), and moves the edited data from the output queue into the output RU (request unit).

Entry from: IKT32700

Registers at Entry:

Register 0: Address of a parameter list containing:

- Address of output queue manager parameter list (OPARMS)
- Address of editing flags (WRKFLGS)
- Address of RU buffer size (TOMBUFSZ)
- Address of next unused RU byte (TOMBFPTR)
- Address of amount of data in the RU (BUFCNT)
- Address of next unused data byte (QBUFPTR)
- Address of amount of data moved (MOVCNT)
- Address of index indicating amount of data moved (QDTA)

Register 13: Address of register save area

Register 14: Return address

Register 15: Entry point address

Exit to: IKT32700

Registers at Exit:

Registers 0-14: Restore to the values they had at entry.

Register 15: Return code:

- 00 The IBM-supplied routine IKT32700 should perform editing. Data is still on the output queue.
- Nonzero The exit routine performed the entire edit operation; portions of IKT32700 are bypassed. The data has been moved to the output RU.

Module Attributes: Reenterable

IKTIDSX2 -- Installation-Written Input Edit Exit Routine for IBM 3270 Terminals

Names:

Object Module: IKTIDSX2
Load Module: IKTIOM02

Function: An installation 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, broken into line segments, and placed on the input queue.

Entry_from: IKT3270I

Registers_at_Entry:

Register 0: Address of the input data
Register 1: Data length
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKT3270I

Registers_at_Exit: Restore all registers to the values they had at entry.

Module_Attributes: Reenterable

IKTIDSX3 -- Installation-Written Attention Handler for IBM 3270 Terminals

Names:

Object Module: IKTIDSX3
Load Module: IKTIOM02

Function: An installation may write this routine to handle attention interruptions from IBM 3270 LU0 terminals during input editing instead of using the IBM-supplied routine IKTATTN. One use of an installation-written attention handler might be to clear the queues conditionally (rather than unconditionally) when an attention interruption is received.

Entry_from: IKTIMIDS

Registers_at_Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTIMIDS

Registers_at_Exit: Restore all registers to the values they had at entry.

Module_Attributes: Reenterable

IKTIDSX4 -- Installation-Written Edit Exit Routine for IBM 3270 Terminals

Names:

Object Module: IKTIDSX4
Load Module: IGCC009C

Function: An installation may write this routine to perform 3270 editing in place of or in addition to that performed by the IBM-supplied code at statement label EDIT3270 in module IKTVTGET. EDIT3270 scans for invalid data and 3270 control characters and moves the data from the input queue to the TGET requester's data area. An installation might write an edit exit routine to change TGET EDIT editing criteria.

Entry_from: IKTVTGET

Registers_at_Entry:

Register 0: Address of extended save area (XSA)
Register 1: Address of input queue manager parameter list (IPARMS)
Register 13: Address of register save area (serialized by the local lock; IKTIDSX4 should not release the lock if the save area is used)
Register 14: Return address
Register 15: Entry pcint address

Exit_to: IKTVTGET

Registers_at_Exit:

Registers 0-14: Restore to the values they had at entry.
Register 15: Return code:

- 00 The exit routine performed the entire exit operation and moved the input data to the TGET data area.
- 04 The exit routine performed only a data scan; the IBM-supplied code (EDIT3270) should perform editing.

Module_Attributes: Reenterable

IKTIDSX5 -- Installation-Written Attention Handler for IBM 3270 LU2 Terminals

Names:

Object Module: IKTIDSX5
Load Module: IKTICM02

Function: An installation may write this routine to handle attention interruptions from IBM 3270 LU2 terminals during output editing, instead of using the IBM-supplied routine IKTATTN. An installation-written attention handler may be used, for example, to clear the queues conditionally (rather than unconditionally) when an attention interruption is received.

Entry_from: IKTIMLU2

Registers_at_Entry:

Register 13: Address of save area
Register 14: Return address
Register 15: Entry pcint address

Exit_to: IKTIMLU2

Registers_at_Exit: Restore all registers to the values they had at entry.

Module_Attributes: Reenterable

IKTIICH -- TIM/TOM Initialization Routine

Names:

Assembly Module: IKTIICH
Object Module: IKTIICH
Alternate Entry Points: Ncne
Load Module: IKJEFLA

MC Diagram: 1.1.1

Function: This routine initializes the terminal input manager SRB and the terminal output manager SRB.

Entry from: IKTLTERM, IKTXINIT

Registers at Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry pcint address

Exit to: IKTLTERM, IKTXINIT

Registers at Exit:

Register 15: Return code:

0 Success
Nonzero Failure

External References: None

Control Blocks Used: ASCB, CVI, PSA, SCVT, SRB, TCAST, TSB, ISBX, TVWA

Mapping Macros Used: CVI, IHAASCB, IHAPSA, IHASCVT, IHASRB, IKJTSE, IKTTCAST, IKTTVWA

Executable Macros Used: FREEMAIN, GETMAIN, MODESET, SCHEDULE, SETLOCK

Module Attributes: Reenterable, privileged, key 0

Lock Dependency: The local lock is held on entry.

Messages: None

Abend Codes: None

Data Sets: None

IKTIMIDS -- 3270 LU0 Terminal Input Manager

Names:

Assembly Module: IKTIMIDS
Object Module: IKTIMIDS
Alternate Entry Points: Ncne
Load Module: IKTIOM02

MC Diagram: 3.1

Function: This routine obtains (RECEIVE macro) data from VTAM buffers (for IBM 3270 LU0 terminals) and places it on the input queue for processing by the TGET routine.

Entry from: IKTIOM through dispatcher (that is, scheduled as a SRB by IKTIOM) to entry point IKTIMIDS: Normal

Registers at Entry:

Register 1: Address of a work area

Exit to:

Dispatcher: Normal
IKTIOFRR: By way of R/TM

Registers at Exit:

Register 15: Reason code 0103 (hex) ifabend 0AB occurred during IKTIMIDS execution

External References: IKTATTN, IKTEXIT, IKTIDSX3, IKTIOFRR, IKTOHIN, IKT3270I

Control Blocks Used: ASCB, CVT, FRRS, PSA, RPL, SCVT, SDWA, SRB, TCAST, TSP, TSBX, TVWA, WSAVT

Mapping Macros Used: CVT, IFGRPL, IHAASCB, IHAFRRS, IHAPSA, IHASCVT, IHASDWA, IHASRB, IHAWSAVT, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST, IKTIWVA, ISTUSFCB

Executable Macros Used: ABEND, FREEMAIN, RECEIVE, SCHEDULE, SETFRR, SETLOCK

Module Attributes: Reenterable

Lock Dependency: None

Messages: IKT00400I (detected)

Abend Codes: 0AB with reason code 0103 (hex) in register 15 (RECEIVE macro error)

Data Sets: None

IKTIMLU2 -- 3270 LU2 Terminal Input Manager

Names:

Assembly Module: IKTIMLU2
Object Module: IKTIMLU2
Alternate Entry Points: None
Load Module: IKTIOM03

MC Diagram: 3.5

Function: This routine issues the VTAM RECEIVE macro, determines what the input is when the RECEIVE is satisfied, calls the appropriate routine to process the input and loops back to reissue the RECEIVE. The RECEIVE may be satisfied with DATA, CANCEL, LUSTAT, SIGNAL, CLEAR or some kind of error condition.

Entry from: Scheduler

Registers at Entry:

Register 1: Address of a work area

Exit to:

Branch Register 14: Normal
IKTIOFRR: By way of R/TM

Registers at Exit: N/A

External References: IKTATTN, IKTEXTIT, IKTIDSX5, IKTIOFRR, IKTQMIN,
IKI3270I

Control Blocks Used: ASCB, CVT, FRRS, PSA, RPL, SRB, TCAST, TIMWA,
TSE, TSBX, TVWA

Mapping Macros Used: CVT, IFGRPL, IFGRPLVT, IHAASCB, IHAFRRS, IHAPSA,
IHASCVT, IHASRB, IHASAVT, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST,
IKTTIMWA, IKTTVWA, ISTRPLFB, ISTUSPBC

Executable Macros Used: ABEND, FREEMAIN, GETMAIN, RECEIVE, SCHEDULE,
SEND, SETFRR, SETLOCK

Module Attributes: Reenterable

Lock Dependency: Local lock is required to process incoming data

Messages: None

Abend Codes: OAB for no retry (103 reason code)

Data Sets: None

IKTINX2 -- Installation-Written I/O Manager Initialization Routine

Names:

Object Module: IKTINX2
Load Module: IKJEPLA

Function: An installation must write this routine if it wants to use installation-written terminal input managers and terminal output managers. IKTINX2 should perform the same function for installation-written TIMs and TOMs that module IKTIOM performs for the IBM-supplied TIMs and TOMs. IKTIOM allocates storage for and initializes the TIM and TOM SRPs.

Entry from: IKTXINIT

Registers at Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTXINIT

Registers at Exit:

Register 15: Return code:
00 Successful initialization

Nonzero Unsuccessful initialization

Module Attributes: Reenterable

IKTIOFRR -- I/O Functional Recovery Routine

Names:

Assembly Module: IKTIOFRR
Object Module: IKTIOFRR
Alternate Entry Points: None
Load Module: IKTIO00

MO Diagram: 3.11

Function: This routine receives control from R/TM when a nonrecoverable error has been encountered in the terminal input or output manager, when an ABEND has been issued by the input or output manager, or when IKTIOFRR has been percolated to from another PRR.

Entry from: R/TM

Registers at Entry:

Register 0: SDWA availability
Register 1: SDWA address

Exit to:

Caller: Normal

Registers at Exit:

Register 15: Return code when SDWA is not available

External References: None

Control Blocks Used: ASCB, CVT, PSA, SDWA, SRB, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IPAASCB, IPASRB, IHAPSA, IHASCVT, IHASDWA, IKJTSTB, IKTTVWA

Executable Macros Used: CALLBTM, FREEMAIN, SETLOCK

Module Attributes: Reenterable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTIST00 -- Lost Terminal Exit Scheduler

Names:

Assembly Module: IKTIST00
Object Module: IKTIST00

Alternate Entry Points: Ncne
Load Module: IKT10MCO

MC_Diagram: 3.12

Function: This module builds a UECB for the lost terminal exit and then schedules the exit using a TPQUE macro.

Entry_from: IKT10FRR

Registers_at_Entry:

Register 1: Address of RPL

Registers_at_Exit:

Register 15: Return code (in hex):

0 UECB and VRPL built, exit scheduled
4 Storage not available for UECB and VRPL

External_References: None

Control_Blocks_Used: ISTACDEE, ISTATCVT, ISTEPAB, ISTUECB, ISTEPST, ISTECE, IFGACB, IFGEXLST, IHAFSA, IFGRPL, IHAASCB, IHAASXB, IHASRB

Mapping_Macros_Used: None

Executable_Macros_Used: TPQUE, RELSTORE, REQSTORE

Module_Attributes: Reentrant

Lock_Dependency: This module is entered with the local lock held.

Messages: None

Abend_Codes: None

Data_Sets: None

IKTLOGFF -- Extended Logoff Routine

Names:

Assembly Module: IKTLOGFF
Object Module: IKTLOGFF
Alternate Entry Points: Ncne
Load Module: IKJEFLA

MC_Diagram: 1.4

Function: This routine disconnects (CLSDST macro) the user's terminal from ISO and closes the user's ACB.

Entry_from: IKJEFLB

Registers_at_Entry: Irrelevant

Exit_to:

IKJEFLB: Normal
R/TM: Abend OAB

Registers at Exit:

Register 15: Reason code 0203 (hex) if abend 0AB occurred during IKTLOGFF execution

External References: IKITOMJR

Control Blocks Used: ASCB, CVT, PSA, TCAST, TSB, TSBX, TVWA

Mapping Macros Used: IHAASCB, IHAPSA, IKJTSB, IKTTCAST, IKITVWA

Executable Macros Used: ABEND, CLSDST, EXECRPL, SETLOCK

Module Attributes: Reenterable

Lock Dependency: None

Messages: None

Abend Codes: 0AB with reason code 0203 (hex) in register 15 (cannot clcse ACB)

Data Sets: None

IKTLOGR -- Logon Reconnect Routine

Names:

Assembly Module: IKTLOGR
Object Module: IKTLOGR
Alternate Entry Points: None
Load Module: IKJEFLE

MC Diagram: 1.3

Function: This routine establishes reconnection of a TSO/VTAM terminal to the address space from which it was disconnected. It is invoked when a logon command specifies the reconnect operand.

Entry from: IKJEFLEA

Registers at Entry:

Register 1: Address of parameter list containing user ID and password

Exit to: IKJEFLEA

Registers at Exit:

Register 15: Return code (in hex):

00 Successful reconnection
04 Invalid password
08 Invalid user ID
0C Unsuccessful reconnection

External References: IKTMSGs, IKTQMOUT

Control Blocks Used: ASCB, CVT, PSA, TCAST, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IHAASCB, IHAPSA, IKJTSB, IKTOPARM, IKTTCAST, IKTTVWA

Executable_Macros_Used: CLSDST, EXECRPL, POST, SCHEDULE, SETLOCK

Module_Attributes: Reenterable, privileged, key 0

Lock_Dependency: None

Messages: IKTO0301I

Abend_Codes: None

Data_Sets: None

IKTLTERM -- LOSTERM Exit Routine

Names:

Assembly Module: IKTLTERM
Object Module: IKTLTERM
Alternate Entry Points: None
Load Module: IKTLTERM

MC_Diagram: 1.5

Function: This routine processes VTAM USS logoff requests, and handles certain VTAM transmission errors, for TSO/VTAM time-sharing.

Entry_from: Dispatcher (that is, scheduled as an IRE by VTAM)

Registers_at_Entry:

Register 1: Address of a parameter list containing:

Word 1: Address of the terminal's ACB
Word 2: Communications identifier (CID) of the terminal
Word 3: TSB address from node initialization block (NIB) user field
Word 4: Code indicating reason for entering this routine

Exit_to:

Dispatcher: Normal
R/TM: Abend OAB

Registers_at_Exit:

Register 15: Reason code 0105 (hex) if abend OAB occurred during IKTLTERM execution.

External_References: IKTIIOH, IKTINX2, IKTMSGs, IKTQMIN, IKTQMOUT

Control_Blocks_Used: ASCB, CSCB, CVT, ECB, NIB, PSA, RPL, SCVT, SRB, TCAST, TCB, TSB, TSBX, TVWA

Mapping_Macros_Used: CVT, IEECHAIN, IFGRPL, IFGRPLVT, IHAASCB, IHAECB, IHAPSA, IHASCVT, IHASRB, IKJTCE, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTICAST, IKTTVWA, ISTNIB, ISTBPLFB

Executable_Macros_Used: ABEND, CLSDST, EXECRPL, FREEMAIN, GETMAIN, MCESET, OPNDST, SCHEDULE, SETLOCK, STIMER, TIMER, WAIT, WTO

Module_Attributes: Reenterable, privileged, key 0

Lock_Dependency: None

Messages: IKT100I, IKT101I, IKT102I, IKT103I, IKT107I, IKT00300I

Abend_Codes: OAB with reason code 0105 (hex) in register 15 (Bind failure)

Data_Sets: None

IKTMSGSGS --- VTIOC Messages Module

Names:

Assembly Module: IKTMSGSGS
Object Module: IKTMSGSGS
Alternate Entry Points: None
Load Module: IKTMSGSGS

MC_Diagram: None

Function: This module contains VTIOC message texts. VTIOC modules that issue messages obtain the address of IKTMSGSGS from TCASMSGSGS and use the message number as an offset to obtain the text.

Entry_from: N/A

Registers_at_Entry: N/A

Exit_to: N/A

Registers_at_Exit: N/A

External_References: None

Control_Blocks_Used: None

Mapping_Macros_Used: None

Executable_Macros_Used: None

Module_Attributes: Not executable

Lock_Dependency: None

Messages: None

Abend_Codes: None

Data_Sets: None

IKTIOMIDS --- 3270 LU0 Terminal Output Manager

Names:

Assembly Module: IKTCMIDS
Object Module: IKTIOMIDS
Alternate Entry Points: None
Load Module: IKTIOM02

MC_Diagram: 3.2

Function: This routine takes, from the output queue, output processed by the TPUT routine, and sends it to a VTAM buffer for transmission to an IBM 3270 LU0 terminal.

Entry from: IKTIMIDS, IKTLTERM, IKTVTGET, IKTVTPUT, IKTXMTPT, IKT0940C, IKT0940E, IKT09401, or IKT09409 through dispatcher (that is, scheduled as a SRB)

Registers at Entry:

Register 1: Address of a work area

Exit to:

Dispatcher: Normal
IKTIOFRR: By way of R/TM

Registers at Exit:

Register 15: Reason code 0104 (hex) if abend OAB occurred during IKTCMIDS execution

External References: IKTEXTIT, IKTIOFRR, IKTQMOUT, IKT32700

Control Blocks Used: ASCB, CVT, FRRS, PSA, RPL, SCVT, SDWA, SRB, TCAST, TSE, TSEX, TVWA, WSAVT

Mapping Macros Used: CVT, IPGRPL, IHAASCB, IHAFRRS, IHAPSA, IHASCVT, IHASDWA, IHASRB, IHAWSAVT, IKJTSB, IKTEQU, IKTOPARM, IKTTCAST, IKTTVWA

Executable Macros Used: ABEND, FREEMAIN, SEND, SETFRR, SETLOCK, STATUS, SYSEVENT

Module Attributes: Reenterable

Lock Dependency: None

Messages: IKT00400I (issued), IKT00405I

Abend Codes: OAB with reason code 0104 (hex) in register 15 (SEND macro error)

Data Sets: None

IKTCMLU2 -- 3270 LU2 Terminal Output Manager

Names:

Assembly Module: IKTCMLU2
Object Module: IKTCMLU2
Alternate Entry: None
Load Module: IKTIOM03

MC Diagram: 3.6

Function: To send messages and controls to 3270 LU2 terminals.

Entry from: (Through dispatcher) IKTIMLU2, IKTLTERM, IKTVTGET, IKTVTPUT, IKTXMTPT, IKT0940C, IKT0940E, IKT09401, or IKT09409

Registers_at_Entry:

Register 1: Address of work area
Register 0: Address of SRE

Exit_tc:

Branch Register 14: Normal
Abend to IKTIOPRR: By way of R/TM

Registers_at_Exit: N/A

External_References: IKTIOPRR, IKTQMOUT

Control_Blocks_Used: ASCB, ASXB, PRRS, IFGRPL, ISTMPST, ISTOPAB, PSA, RFLCF60, SCVTSECT, SRB, TCAST, TSB, TSBX, TVWA

Mapping_Macros: CVT, IFGRPL, IHAASCB, IHAFRRS, IHAPSA, IHASCVT, IKTEQU, IKTOPARM, IKTTCAST, IKTTCHWA, IKTTVWA

Executable_Macros_Used: ABEND, FREEMAIN, RECEIVE, SEND, SETFRR, SETLOCK, SYSEVENT

Module_Attributes: Reenterable

Lock_Dependency: Local lock must be obtained upon entry

Messages: None

Abend_Codes: .OAB with reason code 104 in register 15

Data_Sets: None

IKTQMNGR -- Queue Manager

Names:

Assembly Module: IKTQMIN
Object Module: IKTQMIN
Alternate Entry Points: IKTQMOUT, IKTQMFR1, IKTQMEV, IKTQMFR2
Load Module: IKTQMIN

MC_Diagram: 4.1, 4.2

Function: This routine adds, updates, and removes input and output queue elements.

Entry_from: IKIATTN, IKTIMIDS, IKTIMLU2, IKTLOGR, IKTLTERM, IKTOMIDS, IKTOMLU2, IKTTIMRT, IKTTQRT, IKTVTGET, IKTVTPUT, IKTXHTPT, IKTO9401, IKT3270I, IKT3270O, to one of the following entry points:

IKTQMIN: Input queue functions
IKTQMOUT: Output queue functions

R/TM to one of the following entry points:

IKTQMFR1: Functional recovery to trap errors and retry the request
IKTQMFR2: Functional recovery to trap errors and retry element verification
IKTQMEV: Element verification during recovery

Registers at Entry:

To entry point IKTQMIN:

Register 1: Address of parameter list IPARMS

To entry point IKTQHOUT:

Register 1: Address of parameter list OPARMS

To entry point IKTQMFR1:

Register 1: Address of SDWA

To entry point IKTQMFR2:

Register 1: Address of SDWA

To entry point IKTQMEV:

Register 0: Address of storage to be verified

Register 1: Contents of register 0 on entry to IEAVEQV3

Exit to: Caller

Registers at Exit:

After entry at IKTQMEV:

Register 15: Return code:

- 00 The storage location contains a queue element and can be referenced.
- 04 The storage location contains a queue element and can be referenced, but the queue element contains data not related to the queuing information.
- 08 Either the storage location cannot be referenced or it can be referenced but does not contain a queue element.

External References: IEAVEQV3, STATUS macro routine

Control Blocks Used: ASCE, CVT, FRRS, PSA, QVPL, SCVT, SDWA, SRB, TCAST, TSB, TSBX, TVCS, TVWA

Mapping Macros Used: CVT, IHAASCB, IHAFRRS, IHAPSA, IHAQVPL, IHASCVT, IHASDWA, IHASRB, IKJTSB, IKTIFARM, IKTOPARM, IKITCAST, IKTTVCS, IKTTVWA

Executable Macros Used: ABEND, BLECPPOOL, FREECELL, FREEMAIN, GETCELL, GETMAIN, MODESET, SCHEDULE, SETPRR

Module Attributes: Reenterable, caller's key, supervisor state

Lock Dependency: The local lock is held on entry.

Messages: IKT00400I, IKT00401I

Abend Codes: OAD (GETCELL/FREECELL macro error)

Data Sets: None

IKTRPLXT -- OPNDST RPL Asynchronous Exit Routine

Names:

Assembly Module: IKTRPLXT
Object Module: IKTRPLXT
Alternate Entry Points: Ncne
Load Module: IKJEFLA

MC Diagram: 1.1.2

Function: This routine checks (CHECK macro) the status of the OPNDST RPL request issued in the VTICC initialization routine (IKTXINIT).

Entry from: Dispatcher (that is, scheduled as an IRB by VTAM)

Registers at Entry: Irrelevant

Exit to: Dispatcher

Registers at Exit: Irrelevant

External References: Ncne

Control Blocks Used: ASCB, PSA, TSB, TSBX, TVWA

Mapping Macros Used: IHAASCB, IHAPSA, IKJTSB, IKTTVWA

Executable Macros Used: CHECK, EXECRPL, MODESET, POST

Module Attributes: Reenterable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTRTY1 -- Installation-Written Output Edit
Exit Routine for IBM 3767 and IBM 3770 Terminals

Names:

Object Module: IKTRTY1
Load Module: IKTIOM01

Function: An installation may write this routine to perform 3767 or 3770 output editing in place of or in addition to that performed by the IBM-supplied code at statement labels EDITASIS and CNTRLOPT in module IKTTOMRT. The IBM-supplied code scans data, edits it according to the TPUT operands specified (EDIT, ASIS, or CONTROL), and moves the edited data from the output queue into the output RU (request unit).

Entry from: IKTTOMRT

Registers at Entry:

Register 0: Address of a parameter list containing:

- Address of output queue manager parameter list (OPARMS)

- Address of next unused RU byte (BYTECNT)
- Address of next unused data byte (BUPPTR)
- Address of the RU (RU)
- Address of amount of printable characters (CHARCNT)
- Address of flags (BITVALUO)
- Address of the TSB (ASCBTSB) (leftmost bit "on" indicates last item in this parameter list)

Register 13: Address of register save area
 Register 14: Return address
 Register 15: Entry point address

Exit_to: IKTTOMRT

Registers_at_Exit:

Registers 0-14: Restore to the values they had at entry.
 Register 15: Return code:

- 00 The IBM-supplied code should perform editing. Data is still on the output queue.
- Nonzero The exit routine performed the entire edit operation; the IBM-supplied code is bypassed. The data has been moved to the output RU.

Module_Attributes: Reenterable

IKTRTX2 -- Installation-Written Input Edit Routine for IBM 3767 and IBM 3770 Terminals

Names:

Object Module: IKTRTX2
 Load Module: IKTIOM01

Function: An installation may write this routine to perform input scanning and editing in addition to that performed by the IBM-supplied code near statement label DATA2 in module IKTTIMRT. If provided, IKTRTX2 is called after the data is translated (if necessary) from ASCII code to EBCDIC but before it is scanned for input line delimiters, broken into single lines, and placed on the input queue.

Entry_from: IKTTIMRT

Registers_at_Entry:

Register 0: Address of the input data
 Register 1: Data length
 Register 13: Address of register save area
 Register 14: Return address
 Register 15: Entry point address

Exit_to: IKTTIMRT

Registers_at_Exit: Restore all registers to the values they had at entry.

Module_Attributes: Reenterable

IKTRTX3 -- Installation-Written Attention Handler for IBM 3767 and
IBM 3770 Terminals

Names:

Object Module: IKTRTX3
Load Module: IKTIOM01

Function: An installation may write this routine to handle attention interruptions from IBM 3767 or IBM 3770 terminals instead of using the IEM-supplied code at statement label ATTN in module IKTTIMRT. One use of an installation-written attention handler might be to clear the queues conditionally (rather than unconditionally) when an attention interruption is received.

Entry_from: IKTTIMRT

Registers_at_Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTTIMRT

Registers_at_Exit: Restore all registers to the values they had at entry.

Module_Attributes: Reenterable

IKTRTX4 -- Installation-Written Edit Exit Routine for IBM 3767 and
IBM 3770 Terminals

Names:

Object Module: IKTRTX4
Load Module: IGCC009C

Function: An installation may write this routine to perform 3767 or 3770 editing in place of or in addition to that performed by the IEM-supplied code at statement label EDIT3767 in module IKTVTGET. EDIT3767 scans for invalid data and moves the data from the input queue to the TGET requester's data area. An installation might write an edit exit routine to change TGET EDIT editing criteria.

Entry_from: IKTVTGET

Registers_at_Entry:

Register 0: Address of extended save area (XSA)
Register 1: Address of input queue manager parameter list (IPARMS)
Register 13: Address of register save area (serialized by the local lock; IKTRTX4 should not release the lock if the save area is used)
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTVTGET

Registers at Exit:

Registers 0-14: Restore to the values they had at entry.
Register 15: Return code:

- 00 The exit routine performed the entire edit operation and moved the input data to the TGET data area.
- 04 The exit routine performed only a data scan; the IBM-supplied code (EDIT3767) should perform editing.

Module Attributes: Reenterable

IKTTIMRT -- 3767/3770 Terminal Input Manager

Names:

Assembly Module: IKTTIMRT
Object Module: IKTTIMRT
Alternate Entry Points: IKTIMFRR
Load Module: IKTIOM01

MC Diagram: 3.3

Function: This routine obtains (RECEIVE macro) data from VTAM buffers (for IBM 3767 and IBM 3770 terminals) and places it on the input queue for processing by the TGET requester.

Entry from: IKTIOM through dispatcher (that is, scheduled as a SRB by IKTIOM) to entry point IKTTIMRT: Normal

R/TM to entry point IKTIMFRR: Error

Registers at Entry:

Register 1: Address of a work area

Exit to:

Dispatcher: Normal
IKTIMFRR: Error

Registers at Exit:

Register 15: Reason code 0101 (hex) if abend 0AB occurred during IKTTIMRT execution

External References: IKIQMIN, IKIQMOUT, IKTRTX2, IKTRTX3, PCST macro routine, STATUS macro routine

Control Blocks Used: ASCB, CVT, FRRS, PSA, RPL, SCVT, SDWA, SRB, TCAST, TSE, TSBX, TVWA, WSAVT

Mapping Macros Used: CVT, IFGRPL, IHAASCB, IHAFRRS, IHAPSA, IHASCVT, IHASDWA, IHASRB, IHAWSAVT, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST, IKITVWA, ISTUSFBC

Executable Macros Used: ABEND, CALLR/TM, FREEMAIN, RECEIVE, SETFRR, SETLCK

Module Attributes: Reenterable

Lock Dependency: None

Messages: IKT00400I (detected), IKT00402I (detected)

Abend Codes: OAB with reason code 0101 (hex) in register 15 (RECEIVE macro error)

Data Sets: None

IKTICMJR -- Terminal Output Manager Null RU Routine

Names:

Assembly Module: IKTICMJR
Object Module: IKTTOMJR
Alternate Entry Pcints: Ncne
Load Module: IKTTOMJR

MC Diagram: 3.2.1

Function: This routine sends (SEND macro) a null RU when one of the terminal output managers cannot send an RU because it is suspended. This happens after a TSO/VTAM user has been canceled by the system-initiated cancellation routine, a conditional logoff has occurred, or a cross-memory TPUT has been sent to a 3767, 3770, or 3270 LU2 terminal. Sending a null RU causes VTAM to be dispatched, thus notifying VTAM that the user's address space is swapped in again so that the terminal output manager can be dispatched.

Entry from: IKTLOGFF, IKTLTERM, IKTXMTPT

Registers at Entry: N/A

Exit to: Caller

Registers at Exit: N/A

External References: Ncne

Control Blocks Used: ASCB, CVT, NIB, PSA, RPL, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IFGRPL, IHAASCB, IHAPSA, IKJTSB, IKTTVWA, ISTNIE

Executable Macros Used: FREEMAIN, GETMAIN, SEND, SETLOCK

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKTICMRT -- 3767/3770 Terminal Output Manager

Names:

Assembly Module: IKTICMRT
Object Module: IKTTOMRT

Alternate Entry Points: IKTOMFRR
Load Module: IKTIOM01

MC Diagram: 3.4

Function: This routine takes, from the output queue, output processed by the TPUT routine, and sends it to a VTAM buffer for transmission to an IBM 3767 or IBM 3770 terminal.

Entry from: IKTLTERM, IKTTIMRT, IKTVTPUT, IKTXMTPT, IKT0940C, IKT0940E, IKT09401, or IKI09409 through dispatcher (that is, scheduled as a SRB) to entry point IKTTOMRT: Normal

R/IM to entry point IKTCMFRR: Error

Registers at Entry:

Register 1: Address of a work area

Exit to:

Dispatcher: Normal
IKTOMFRR: Error

Registers at Exit:

Register 15: Reason code 0102 (hex) if abend OAB occurred during IKTTOMRT execution

External References: IKTRTX1, IKTQMOUT, STATUS macro routine

Control Blocks Used: ASCB, CVT, FRRS, PSA, RPL, SCVT, SDWA, SRB, TCAST, TSE, TSBX, TVWA, WSAVT

Mapping Macros Used: CVT, IFGRPL, IFGRPLVT, IHAASCB, IHAFRRS, IHAPSA, IHASCVT, IHASDWA, IHASRB, IHAWSAVT, IKJTSB, IKTEQU, IKTOPARM, IKTTCAST, IKTTVWA

Executable Macros Used: ABEND, CALLR/TM, FREEMAIN, RECEIVE, RESETSR, SEND, SESSIONC, SETPRR, SETLOCK, SYSEVENT

Module Attributes: Reenterable

Lock Dependency: None

Messages: IKT00400I (issued), IKT00402I (issued), IKT00403I

Abend Codes: OAB with reason code 0102 (hex) in register 15 (SEND, RECEIVE, or SESSIONC macro error)

Data Sets: None

IKTVTGET -- TGET Routine

Names:

Assembly Module: IKTVTGET
Object Module: IKTVTGET
Alternate Entry Points: Ncne
Load Module: IGC0009C

MC Diagram: 2.1

Function: This routine handles TSO/VTAM TGET requests. It takes data from the input queue, edits it, and places it in the TGET requester's buffer.

Entry from: IGC0009C

Registers at Entry:

Register 3: Address of CVT
Register 4: Address of TCE
Register 5: Address of SVRE
Register 8: Address of ASCE
Register 9: Address of TSE

Exit to: Caller (SVC exit)

Registers at Exit:

Register 15: Return code (in hex):

00 Successful.
04 No input was available and TPUT NOWAIT was specified.
08 An attention interruption occurred. The message was not received.
0C The TGET requester's buffer was too small to hold the input line.
14 The terminal was disconnected and could not be reached.

External References: IKGETXT, IKTIDSX4, IKTQMIN, IKTRTX4, STATUS
macro routine

Control Blocks Used: ASCB, ASXB, CVT, MPST, PSA, RB, SCVT, SRB, TCAST, TCB, TCT, TSB, TSBX, TVWA, TXSA

Mapping Macros Used: CVT, IEEXSA, IEFTCT, IHAASCB, IHAASXB, IHAPSA, IHARE, IHASCVT, IHASRB, IKJRB, IKJTCB, IKJTSB, IKTIPARM, IKTTCAST, IKTVWA, ISTPAB, ISTDYAB, ISTMST

Executable Macros Used: MODESET, SCHEDULE, SETLOCK, STATUS, SYSEVENT

Module Attributes: Reenterable, privileged, enabled, key 0

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKTVTPUT -- TPUT without ASID Routine

Names:

Assembly Module: IKTVTPUT
Object Module: IKTVIFUT
Alternate Entry Points: None
Load Module: IGC0009C

MC Diagram: 2.2

Function: This routine handles TSO/VTAM TPUT without ASID requests. It takes data from the TPUT requester's buffer, edits it, and places it on the output queue.

Entry from: IGC0009C

Registers at Entry:

Register 3: Address of CVT
Register 4: Address of TCE
Register 5: Address of SVRE
Register 8: Address of ASCB
Register 9: Address of TSE

Exit to: Caller (SVC exit)

Registers at Exit:

Register 15: Return code (in hex):

00 Successful.
04 A terminal output buffer was not available and TPUT NOWAIT was specified.
08 An attention interruption occurred. The message was not sent.
14 The terminal was disconnected and could not be reached.

External References: IKTCMOUT, STATUS macro routine

Control Blocks Used: ASCB, CVT, PSA, RB, SCVT, SRB, TCAST, TCB, TCT, TSB, TSBX, TVWA, XSA

Mapping Macros Used: CVT, IEEXSA, IEFTCT, IHAASCB, IHAPSA, IHARB, IHASCVT, IHASRB, IKJTCB, IKJTSB, IKTOPARM, IKTTCAST, IKTTVWA

Executable Macros Used: MODESET, SCHEDULE, SETLOCK, STATUS, SYSEVENT

Module Attributes: Reenterable, privileged, enabled, key 0

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKTXINIT --- VTIOC Initialization Routine

Names:

Assembly Module: IKTXINIT
Object Module: IKTXINIT
Alternate Entry Points: None
Load Module: IKJEFLA

MC Diagram: 1.1

Function: This routine initializes VTAM control blocks and the TVWA, and transfers control of the terminal from TCAS to the terminal user's address space.

Entry from: IKJEFLA

Registers at Entry: Irrelevant

Exit to:

IKJEFLA: Normal
R/IM: ABEND OAB

Registers at Exit:

Register 15: Return code:

00 Successful initialization
04 Unsuccessful initialization

Or reason code:

0201 (hex) or 0202 (hex) if abend OAB occurred during IKTXINIT execution

External References: BINDUSER (in IKTLTERM), IKTIOM, IKTX2, IKTRPLYT

Control Blocks Used: ACE, ACEVT, ASCB, CSCB, CVT, ECB, EXLST, EXLVT, NIB, PSA, RPL, TCAST, TCP, TSB, TSBX, TVWA, WESTD

Mapping Macros Used: CVT, IEECHAIN, IFGACB, IFGEXLST, IFGRPL, IFGRPLVT, IHAASCB, IHAECB, IHAPSA, IKJTCE, IKJTSB, IKTOPARM, IKTTCAST, IKTTVWA, IKTWESTD, ISTNIB, ISTRPLFB

Executable Macros Used: ABEND, BLDCPOOL, EXECRPL, FREEMAIN, GETCELL, GETMAIN, MODESET, OPNDST, POST, SETLOCK, WAIT, WTO

Module Attributes: Reenterable, privileged, key 0

Lock Dependency: None

Messages: IKT104I, IKT105I, IKT106I

Abend Codes: OAB with reason code 0201 (hex) (invalid application ID) or 0202 (hex) (cannot open ACE).

Data Sets: None

IKTXLOG -- Extended Logon Routine

Names:

Assembly Module: IKTXLOG
Object Module: IKXTLCG
Alternate Entry Points: Mcne
Load Module: IKJEFLA

MC Diagram: 1.2

Function: This routine sets up the address of the logon command buffer as a parameter for the LCGON Scheduling routines.

Entry_from: IKJEFLE

Registers_at_Entry:

Register 1: Address of a location in which the address of the logon buffer will be stored

Exit_to: IKJEFLE

Registers_at_Exit:

Register 1: Address of the address of a buffer containing the logon command

External_References: None

Control Blocks Used: ASCB, PSA, TSB, TSBX, TVWA

Mapping Macros Used: IHAASCB, IHAPSA, IKJTSB, IKTTVWA

Executable Macros Used: None

Module Attributes: Reenterable, privileged, key 0

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTXMTPT -- Target Address Space TPOT with ASID Routine

Names:

Assembly Module: IKTXMTPT
Object Module: IKTXMTPT
Alternate Entry Points: IKTXMFRF
Load Module: IGC0009C

MC Diagram: 2.3.1

Function: This routine handles TSO/VTAM TPOT with ASID requests for the target address space. It takes data from the TVCS and places it on the output queue.

Entry_from:

IKTASTPT through dispatcher (that is, scheduled as a SRB) to entry point IKTXMTPT: To put message on queue and post source

R/IM to entry point IKTXMFRF: To post source with notice that this address space is terminating

Registers_at_Entry: Irrelevant

Exit_to:

Dispatcher: After entry at IKTXMTPT
R/IM: After entry at IKTXMFRF

Registers at Exit:

Register 14: Address of return point

External References: IKTQMOUT, POST macro routine, SYSEVENT macro routine

Control Blocks Used: ASCB, CVI, FRRS, PSA, SCWA, SRB, TCAST, TSB, TSEX, TVCS, TVWA, WSAVT

Mapping Macros Used: CVI, IHAASCB, IHAFRRS, IHAPSA, IHASDWA, IHASRB, IHAWSAVT, IKJTSE, IKTEQU, IKTOPARM, IKTTCAST, IKTTVCS, IKTTVWA

Executable Macros Used: FREEMAIN, SETFRR, SETLOCK

Module Attributes: Reenterable, privileged, enabled, key 0

Lock Dependency: None

Messages: None

Alert Codes: None

Data Sets: None

IKTC009D --- Terminal Control Macro Branching Routine

Names:

Assembly Module: IKT0009D
Object Module: IKT0009D
Alternate Entry Points: None
Load Module: IGC0009D

MC Diagram: 5.1

Function: This routine checks the entry code received from IGC0009D. If the code is valid, it routes control to the appropriate terminal control macro module.

Entry from: IGC0009D

Registers at Entry:

Register 0: Entry code in high-order byte
Register 1: Address of parameter list
Register 3: Address of CVI
Register 4: Address of TCE
Register 5: Address of SVRE
Register 8: Address of TSE

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code from terminal control macro module that was invoked. See OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor for specific return codes.

External References: Terminal control macro routines (see MO 5.1)

Control Blocks Used: Ncne

Mapping Macros Used: None

Executable Macros Used: Ncne

Module Attributes: Reentrant

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: Ncne

Data Sets: None

IKT0940A -- STSIZE Terminal Control Macro Routine

Names:

Assembly Module: IKT0940A
Object Module: IKT0940A
Alternate Entry Points: Ncne
Load Module: IGC0009E

MC Diagram: 5.5

Function: This routine services the STSIZE macro instruction for TSC/VTAM terminals. It sets values in the TSB, depending on the operand specified with STSIZE, to indicate the logical line size and, if a display station, the number of lines on the screen. It sets a switch in the TVWA so that the terminal output manager will check the screen size.

Entry from: IGC0009D

Registers at Entry:

Register 0: Number of lines
Register 1: Line size
Register 5: Address of TSE
Register 12: Entry point address and address of IKT0940A's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code (in hex):
00 Successful
04 Invalid parameter
08 Line size not specified, or number of lines not specified
for display terminal
0C Nonstandard screen size specified

External References: Ncne

Control Blocks Used: TSB, TVWA

Mapping Macros Used: IKJTSB, IKTTVWA

Executable Macros Used: MODESET

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT0940C -- STAUTOCP Terminal Control Macro Routine

Names:

Assembly Module: IKT0940C
Object Module: IKT0940C
Alternate Entry points: Ncne
Load Module: IGC0009L

MC Diagram: 5.6

Function: This routine services the STAUTOCP macro instruction for TSC/VTAM terminals. It sets values in the TSB to start automatic character prompting.

Entry from: IGC0009D

Registers at Entry:

Register 1: Zero
Register 5: Address of TSB
Register 10: Address of constants
Register 12: Entry point address and address of IKT0940C's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.

04 A parameter was specified in register 1 but none was expected.

External References: Ncne

Control Blocks Used: CVT, PSA, SRB, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IHAPSA, IHASRB, IKJTSB, IKTTVWA

Executable Macros Used: SCHEDULE

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT0940D -- SPAUTOPT Terminal Control Macro Routine

Names:

Assembly Module: IKT0940D
Object Module: IKT0940D
Alternate Entry Points: Ncne
Load Module: IGC0009D

MC Diagram: 5.7

Function: This routine services the SPAUTOPT macro instruction for TSC/VTAM terminals. It sets values in the TSB to stop automatic line numbering or automatic character prompting.

Entry from: IGC0009D

Registers at Entry:

Register 1: Zero
Register 5: Address of TSE
Register 12: Entry point address and address of IKT0940D's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.
04 A parameter was specified in register 1 but none was expected.
08 Neither line numbering nor character prompting was in use.

External References: Ncne

Control Blocks Used: TSE

Mapping Macros Used: IKJTSB

Executable Macros Used: MODESET

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT0940E -- RTAUTOPT Terminal Control Macro Routine

Names:

Assembly Module: IKT0940E
Object Module: IKT0940E
Alternate Entry Points: Ncne
Load Module: IGC0009D

EC_Diagram: 5.8

Function: This routine services the RTAUTOPT macro instruction for TSC/VTAM terminals. It sets values in the TSB to restart automatic line numbering or automatic character prompting.

Entry_from: IGC0009D

Registers_at_Entry:

Register 1: Zero
Register 5: Address of TSE
Register 12: Entry point address and address of IKT0940E's base
Register 14: Return address

Exit_to: Caller (SVC exit)

Registers_at_Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.
04 A parameter was specified in register 1 but none was expected.
08 Prompting was not in use.

External_References: Ncne

Control_Blocks_Used: CVT, PSA, SRB, TSB, TSBX, TVWA

Mapping_Macros_Used: CVT, IHAFSA, IHASRB, IKJTSB, IKTTVWA

Executable_Macros_Used: SCHEDULE

Module_Attributes: Reenterable

Lock_Dependency: This module is entered with the local lock held.

Messages: None

Abend_Codes: None

Data_Sets: None

IKT0940F -- STTRAM Terminal Control Macro Routine

Names:

Assembly Module: IKT0940F
Object Module: IKT0940F
Alternate Entry Points: None
Load Module: IGCC009D

MO_Diagram: 5.9

Function: This routine services the STTRAN macro instruction for ISC/VTAM terminals. It sets up, modifies, or discontinues the use of translation tables, depending on the operands specified with STTRAN.

Entry_from: IGC0009D

Registers_at_Entry:

Register 0: Indicators (one byte each):

- Entry code - 0F
- Flags - bit 0 on = NOTRAN specified
bit 1 on = NCCHAR specified
bit 2 on = TCHAR and SCHAR specified
- Terminal character (to be translated in the system)
- System character (to be translated at the terminal)

Register 1: Address of a parameter list containing:

- Address of a pair of user-written translation tables
- Address of the name of the member containing user-written translation tables

Register 5: Address of TSE

Register 12: Entry point address and address of IKT0940F's base

Register 14: Return address

Exit_to: Caller (SVC exit)

Registers_at_Exit:

Register 14: Return address

Register 15: Return code (in hex):

- 00 Successful.
- 04 NOTRAN or NOCHAR was specified but translation was not in effect.
- 08 TABLE or NOCHAR was specified but an address was not provided with the NAME operand.
- 0C Internal error - an unidentifiable flag was set in input register 0.

External References: None

Control Blocks Used: CVT, PSA, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IHAPSA, IKJTSB, IKTTVWA

Executable Macros Used: FREEMAIN, GETMAIN

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT09401 -- TCLEARQ Terminal Control Macro Routine

Names:

Assembly Module: IKT09401
Object Module: IKT09401
Alternate Entry Points: Ncne
Load Module: IGC0009E

MC_Diagram: 5.1

Function: This routine services the TCLEARQ macro instruction for TSC/VTAM terminals. TCLEARQ clears input and output message queues.

Entry_from: IGC0009D

Registers_at_Entry:

Register 1: Indicator:

00 Clear output queue of all but ASID messages.
<0 Clear input queue.

Register 5: Address of TSE
Register 12: Entry point address and address of IKT09401's base
Register 14: Return address

Exit_to: Caller (SVC exit)

Registers_at_Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.
04 Register 1 at entry contained a value greater than zero.

External References: IKIQMIN, IKTQMOUT

Control Blocks Used: CVT, PSA, SRB, TSB, TSBX, TCAST, TVWA

Mapping Macros Used: CVI, IHAPSA, IHASRB, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST, IKTTVWA

Executable Macros Used: SCHEDULE

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT09404 -- STBREAK Terminal Control Macro Routine

Names:

Assembly Module: IKTC9404
Object Module: IKT09404
Alternate Entry Points: Ncne
Load Module: IGC0009D

MC Diagram: 5.2

Function: This routine services the STBREAK macro instruction for TSC/VTAM terminals. It sets a bit in the TSB, depending on the operand specified with STBREAK, to indicate whether TSO is allowed to interrupt input transmission.

Entry from: IGC0009D

Registers at Entry:

Register 1: Flag (left-most bit):

On = STBREAK NO
Cff = STBREAK YES

Register 5: Address of TSB
Register 12: Entry point address and address of IKT09404's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.
04 Register 1 at entry contained a nonzero value in bits 1-31.
08 STBREAK was issued for a 3270 terminal.

External References: Ncne

Control Blocks Used: CVT, PSA, TCAST, TSB, TSBX

Mapping Macros Used: IHBCVT, IHAPSA, IKJTSB, IKTTCAST

Executable Macros Used: MODESET

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKI09405 -- STCCM Terminal Control Macro Routine

Names:

Assembly Module: IKT09405
Object Module: IKT09405
Alternate Entry Points: Ncne
Load Module: IGC0009D

MC_Diagram: 5.3

Function: This routine services the STCOM macro instruction for ISO/VTAM terminals. It sets a bit in the TSB, depending on the operand specified with STCOM, to indicate whether a terminal is allowed to accept messages from other terminals.

Entry from: IGC0009D

Registers at Entry:

Register 1: Flag (left-most bit):

On = STCOM YES
Off = STCCM NO

Register 5: Address of TSB

Register 12: Entry point address and address of IKT09405's base

Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address

Register 15: Return code:

00 Successful
04 Input parameter error

External References: Ncne

Control Blocks Used: TSE

Mapping Macros Used: IKJTSB

Executable Macros Used: MODESET

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT09409 -- STAUTOLN Terminal Control Macro Routine

Names:

Assembly Module: IKT09409
Object Module: IKT09409
Alternate Entry Points: None
Load Module: IGC0009E

MC Diagram: 5.4

Function: This routine services the STAUTOLN macro instruction for TSO/VTAM terminals. It sets values in the TSB, depending on the operands specified with STAUTCLN, that are used in providing automatic line numbering.

Entry from: IGC0009D

Registers at Entry:

Register 0: Address of start value
Register 1: Address of increment value
Register 5: Address of TSE
Register 10: Address of SVC 94 constants
Register 12: Entry point address and address of IKT09409's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code:

C0 Successful.
C4 Start or increment value not within allowable limits
(0-99,999,999).

External References: None

Control Blocks Used: CVT, PSA, SRB, TSB, TSBX, TVWA

Mapping Macros Used: CVT, IHAPSA, IHASRB, IKJTSB, IKTTVWA

Executable Macros Used: SCHEDULE

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT09412 -- STFSMODE Terminal Control Macro Routine

Names:

Assembly Module: IKT09412
Object Module: IKT09412
Alternate Entry Pcints: Ncne
Load Module: IGCC009E

MC_Diagram: 5.10

Function: This routine services the STFSMODE macro instruction for TSO/VTAM terminals. It sets bits in the TVWA, depending on the operands specified with STFSMODE, to indicate whether an IBM 3270 display terminal is to operate in full screen mode, whether this is the first time during execution of a command processor that full screen mode has been entered, and to indicate which program function (PF) key (if any) is to be used as the reshow key.

Entry_from: IGCC009E

Registers_at_Entry:

Register 1:

Flag Bit 0:	Bit 1:	Bits 24-31:
On = STFSMODE on	On = Initial call	RSHWKEY value
Off = STFSMODE off	Off = Not initial	

Register 5: Address of TSP
Register 12: Entry pcint address and address of IKT09412's base
Register 14: Return address

Exit_to: Caller (SVC exit)

Registers_at_Exit:

Register 14: Return address
Register 15: Return code:

00 Successful.
04 Invalid parameter specified to the SVC.
08 Invalid terminal type. This macro instruction is valid only for IBM 3270 display terminals that use TSO/VTAM.

External_References: Ncne

Control_Blocks_Used: TSE, TSEY, TVWA

Mapping_Macros_Used: IKJTSB, IKTTVWA

Executable_Macros_Used: None

Module_Attributes: Reenterable

Lock_Dependency: This module is entered with the local lock held.

Messages: None

Abend_Codes: None

Data_Sets: None

IKT09413 -- STLINENO Terminal Control Macro Routine

Names:

Assembly Module: IKT09413
Object Module: IKT09413
Alternate Entry Points: Ncne
Lcad Module: IGC0009D

MC Diagram: 5.11

Function: This routine services the STLINENO macro instruction for TSO/VTAM terminals. It sets values in the TVWA for IBM 3270 display terminals, according to the operands specified with STLINENO: it sets the number of the screen line on which the next non-full-screen message should appear, and it indicates whether the terminal is to operate in full screen mode.

Entry from: IGC0009D

Registers at Entry:

Register 1: Indicators (one byte each):

- Flag (left-most bit):
 - On = MODE=CN
 - Off = MODE=CPF
- Reserved
- Reserved
- Screen line number that specifies where the next non-full-screen message should appear

Register 5: Address of TSP
Register 12: Entry point address and address of IKT09413's base
Register 14: Return address

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14: Return address
Register 15: Return code:

- 00 Successful.
- 04 Invalid parameter specified to the SVC.
- 08 Invalid terminal type. This macro instruction is valid only for IBM 3270 display terminals that use TSO/VTAM.
- 0C The line number specified was 0 or it was greater than the maximum number of lines allowed for the terminal in use.

External References: None

Control Blocks Used: TSP, TSPX, TVWA

Mapping Macros Used: IKJTSP, IKTTVWA

Executable Macros Used: None

Module Attributes: Reenterable

Lock Dependency: This module is entered with the local lock held.

Messages: None

Abend Codes: None

Data Sets: None

IKT09414 -- STTMPMD Terminal Control Macro Routine

Names:

Assembly Module: IKT09414
Object Module: IKT09414
Alternate Entry Points: Ncne
Load Module: IGC0009D

MC Diagram: 5.12

Function: This routine processes the STTMPMD macro for a TSO/VTAM display terminal. STTMPMD is issued by an application program that wishes to control all the messages that will be sent to or received from the terminal. The KEYS mode can be set on or off to indicate if the application wants the Attention and Clear keys passed to it as data.

Entry from: IGC0009D

Registers at Entry:

Register 1	-	Bit 0	CN - for MODE 'ON'
			OFF - for MODE 'OFF'
	-	Bit 1	CN - for KEYS=ALL
			OFF - for KEYS=NO
Register 5	-	Address of TSB	
Register 12	-	Base address of IKT09414	
Register 14	-	Return address	

Exit to: Caller (SVC exit)

Registers at Exit:

Register 14:	Return address
Register 15:	Return code:
00	Successful.
04	Invalid parameter specified to the SVC.
08	Invalid terminal type. This is not a display terminal.

External References: Ncne

Control Blocks Used: TSE

Mapping Macros Used: Ncne

Executable_Macros_Used: None

Module_Attributes: Reenterable

Lock_Dependency: This module is entered with the local lock held.

Messages: None

Abend_Codes: None

Data_Sets: None

IKT3270I -- 3270 Input Data Handling Routine

Names:

Assembly Module: IKT3270I
Object Module: IKT3270I
Alternate Entry Points: None
Load Module: IKTIOM02

MC_Diagram: 3.9

Function: This routine checks the data received and if applicable, places the data on the input queue for processing by the correct TSO command processor.

Entry_from: IKTIMIDS or IKTIMLU2

Registers_at_Entry: Register 13: Address of a work area

Exit_to: Branch Register 14 to Caller

Registers_at_Exit: N/A

External_References: IKTIASCII, IKTIDSX2, IKTCMIN

Control_Blocks_Used: ASCB, CVT, IFGRPL, PSA, RPL, SCVTSECT, TCAST, TIMWA, TSB, TVWA, WSAVT

Mapping_Macros_Used: CVT, IFGRPL, IFGRPLVT, IHAASCB, IHAPSA, IHASCVT, IHAWSAVT, IKJTSB, IKTEQU, IKTIPARM, IKTOPARM, IKTTCAST, IKTTIMWA, IKTTVWA, ISTRPLFB

Executable_Macros_Used: None

Module_Attributes: Reenterable

Lock_Dependency: Caller must hold the local lock.

Messages: IKT00400I

Abend_Codes: None

Data_Sets: None

IKT32700 -- 3270 Output Data Handling Routine

Names:

Assembly Module: IKT32700
Object Module: IKT32700
Alternate Entry Points: Ncne
Load Module: IKTIOM02

MC Diagram: 3.10

Function: This routine edits messages and controls for 3270 terminals.
Editing depends on screen size and mode of data being processed.

Entry from: IKTOMIDS or IKTOMLU2

Registers at Entry:

Register 13: Address of save area

Exit to: Caller

Registers at Exit: N/A

External References: IKTASCII, IKTIDSX1, IKTMSGS, IKTQMOUT

Control Blocks Used: ASCB, CVI, PSA, RPL, TCAST, TOMWA, TSB, TVWA

Mapping Macros Used: CVI, IHAASCB, IHASCVT, IKAPSA, IKJTSB, IKTEQU,
IKTOPARM, IKTTCAST, IKTTOMWA, IKTTVWA

Executable Macros Used: None

Module Attributes: Reenterable

Lock Dependency: Caller must hold local lock.

Messages: None

Abend Codes: None

Data Sets: None

IKT93EST -- ESTAE Routine for SVC_93

Names:

Assembly Module: IKT93EST
Object Module: IKT93EST
Alternate Entry Points: Ncne
Load Module: IGC0009C

MC Diagram: 2.4

Function: This routine handles abends encountered while processing
TSC/VTAM TGET and TPUT requests.

Entry from: R/TM

Registers_at_Entry:

Register 1: Address of SDWA (if register 0 does not contain hex C)
Register 13: Address of register save area
Register 14: Address of return point

Exit_to: R/TM

Registers_at_Exit:

Register 15: Return code:
00 Continue with termination.

External_References: None

Control_Blocks_Used: ASCB, ASVT, CVT, PSA, RE, SCVT, SDWA, SRB, TSB, TSPX, TVCS, WSAVT, XSA

Mapping_Macros_Used: CVT, IBEXSA, IHAASCB, IHAASVT, IHAPSA, IHARB, IHARSB, IKJTSB, IKTTVCS, IHASCVT, IHASDWA, IHAWSAVT

Executable_Macros_Used: DEQ, FREEMAIN, SETLOCK, SPOST

Module_Attributes: Reenterable, privileged, enabled, key 0

Lock_Dependency: The local lock is obtained to reference the TSB in the local address space. The SALLOC lock is obtained to branch enter the FREEMAIN macro routine.

Messages: None

Abend_Codes: None

Data_Sets: None



VTIOC DIRECTORY

This section shows the relationship between VTIOC load module names, object module names, alternate entry point names, and assembly module names. It also shows which HIPO diagram (in "VTIOC Method of Operation") and program organization description (in "VTIOC Program Organization") is associated with each name.

There are four columns:

- Name - VTIOC names and their relationships to each other. Load module names appear farthest to the left. Indented under each load module name are the names of the object modules, alternate entry points, and assembly modules contained in the load module. VTIOC-only names have the prefix IKT; names having other prefixes are shared by VTIOC and other time-sharing components.
- Type - The type of name: load module, object module, alternate entry point, or assembly module. Names used in more than one way have more than one type.
- MO Diagram - The identification number of the HIPO diagram (in "VTIOC Method of Operation") in which the function corresponding to each name is described.
- PO Name - The name of the routine in "VTIOC Program Organization" that corresponds to each name in column one.

Name	Type	MO Diagram	PO Name
Initialization/Termination Routines			
IGCC001C	Load	None	None
IKTAY8	Assembly	None	IKTAY8
IKTAY81	Object	1.6	IKTAY8
IKTAY82	Entry	1.7	IKTAY8
IKTAY83	Entry	1.8	IKTAY8
IKJEFLE	Load	None	None
IKTXLOG	Object, Assembly	1.2	IKTXLOG
IKILOGR	Object, Assembly	1.3	IKILOGR
IKJEFLA	Load	None	None
IKIXINIT	Object, Assembly	1.1	IKIXINIT
IKTIOM	Object, Assembly	1.1.1	IKTIOM
IKTRPLXT	Object, Assembly	1.1.2	IKTRPLXT
IKTLOGFF	Object, Assembly	1.	IKTLOGFF
IKTINX2	Object (user-written)	1.1	IKTINX2
IKTLTERM	Load, Object, Assembly	1.5	IKTLTERM

Name	Type	MO Diagram	PO Name
TGET/TFUT Routines			
IGC0009C	Load	None	None
IKTVTGET	Object, Assembly	2.1	IKTVTGET
IKTVTPUT	Object, Assembly	2.2	IKTVTPUT
IKTASTPT	Object, Assembly	2.3	IKTASTPT
IKTXTMPT	Object, Assembly	2.3.1	IKTXTMPT
IKTXTMFR	Entry	2.3.1	IKTXTMPT
IKT93EST	Object, Assembly	2.4	IKT93EST
IKTGETXT	Object (user-written)	2.1	IKTGETXT
IKTIDSX4	Object (user-written)	2.1	IKTIDSX4
IKTRTX4	Object (user-written)	2.1	IKTRTX4
Terminal I/C Managers			
IKTIOM00	Load	None	None
IKTEXIT	Object, Assembly	3.7	IKTEXIT
IKTIIST00	Object, Assembly	3.12	IKTIIST00
IKTIOFRR	Object, Assembly	3.11	IKTIOFRR
IKTIOM01	Load	None	None
IKTTIMRT	Object, Assembly	3.3	IKTTIMRT
IKTTIMFR	Entry	3.3	IKTTIMRT
IKTTOMRT	Object, Assembly	3.4	IKTTOMRT
IKTTOMFR	Entry	3.4	IKTTOMRT
IKTRTX1	Object (user-written)	3.4	IKTRTX1
IKTRTX2	Object (user-written)	3.3	IKTRTX2
IKTRTX3	Object (user-written)	3.3	IKTRTX3
IKTIOM02	Load	None	None
IKTIMIDS	Object, Assembly	3.1	IKTIMIDS
IKTOMIDS	Object, Assembly	3.2	IKTOMIDS
IKTIDSX1	Object (user-written)	3.10	IKTIDSX1
IKTIDSX2	Object (user-written)	3.9	IKTIDSX2
IKTIDSX3	Object (user-written)	3.1	IKTIDSX3
IKTATTN	Object, Assembly	3.8	IKTATTN
IKT3270I	Object, Assembly	3.9	IKT3270I
IKT32700	Object, Assembly	3.10	IKT32700
IKTICM03	Load	None	None
IKTIDSX5	Object (user-written)	3.5	IKTIMLU2
IKTIMLU2	Object, Assembly	3.5	IKTIMLU2
IKTOMLU2	Object, Assembly	3.6	IKTOMLU2
IKTASCII	Load, Object, Assembly	3.1,3.2, 3.3,3.4	IKTASCII
IKTTOMJR	Load, Object, Assembly	3.2.1	IKTTOMJR
Queue Manager			
IKTQMNGR	Assembly	None	None
IKTQMIN	Load, Object	4.1	IKTQMNGR
IKTQMOUT	Entry	4.2	IKTQMNGR
IKTQMFR1	Entry	4.1,4.2	IKTQMNGR
IKTQMFR2	Entry	4.1,4.2	IKTQMNGR
IKTQMEV	Entry	4.1,4.2	IKTQMNGR

Name	Type	HO Diagram	PO Name
Terminal Control Macro Routines			
IGC0009D	Load	None	None
IKT0009D	Object, Assembly	5.1	IKT0009D
IKT09401	Object, Assembly	5.2	IKT09401
IKT09404	Object, Assembly	5.3	IKT09404
IKT09405	Object, Assembly	5.4	IKT09405
IKT09409	Object, Assembly	5.5	IKT09409
IKT0940A	Object, Assembly	5.6	IKT0940A
IKT0940C	Object, Assembly	5.7	IKT0940C
IKT0940D	Object, Assembly	5.8	IKT0940D
IKT0940E	Object, Assembly	5.9	IKT0940E
IKT0940F	Object, Assembly	5.10	IKT0940F
IKT09412	Object, Assembly	5.11	IKT09412
IKT09413	Object, Assembly	5.12	IKT09413
IKT09414	Object, Assembly	5.13	IKT09414
IKTMSGs	Load, Object, Assembly	None	IKTMSGs



VTIOC DATA AREAS

This section provides information about VTIOC data areas. It contains:

- A VTIOC control block overview (Figure 10)
- Descriptions of several key internal data areas used by VTIOC routines

Descriptions of the control blocks used by VTIOC can be found in OS/VS2 System Programming Library: Debugging Handbook (Volume 2), GC28-0709, and OS/VS2 System Data Areas, SYB8-0606 (microfiche).

A cross-reference between data area names and VTIOC object modules that use them is contained in the system data area usage table microfiche, OS/VS2 Data Area Usage Table, SYB8-0742. A list giving the names of data area fields and the VTIOC object modules that update them is contained in the system symbol usage table microfiche, OS/VS2 Symbol Usage Table, SYB8-0744.

Consult TSO/VTAM System Information for SU Supplement order numbers.

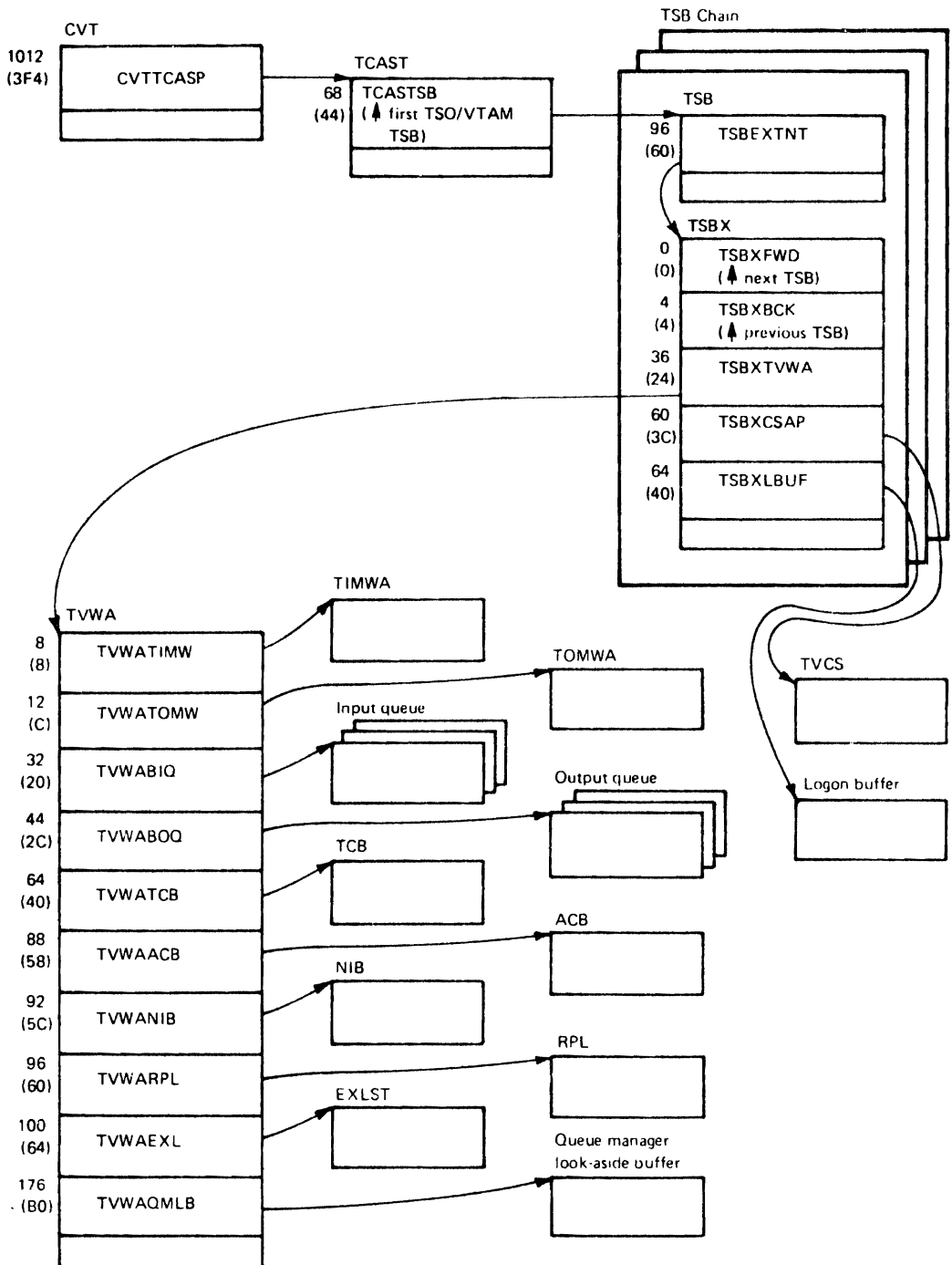


Figure 10. VTIOC Control Block Overview

Input Queue Manager Parameter List

The input queue manager parameter list is used by VTIOC routines that request input queue manager services. It is used to pass parameters to the input queue manager and to receive parameters from the input queue manager. Macro IKTIIPARM (acronym IPARMS) maps the parameter list.

<u>Cff-</u> <u>set</u>	<u>Length</u>	<u>Name</u>	<u>Description</u>
0 (0)	4	IPBUFADR	Address of queue element's message area
4 (4)	1	IPREQ	Request code: 1 = Add element to queue 2 = Obtain message from next element 3 = Remove element from queue 4 = Update an element 5 = Remove all elements 6 = Remove unavailable elements 7 = Mark elements available
5 (5)	1	IPRC	Return code: 1 = Successful (request code 1 or 4) or successful and there are more elements on the queue (request code 2 or 3) 2 = Unsuccessful (request code 1 or 4) or successful and this is last element on queue (request code 2) or successful and queue is empty (request code 3) 3 = Unsuccessful (request code 3)
6 (6)	2	IPBFSZ	Message length
8 (8)	2	IPFLAGS IPALLOC	Flags: X... .. Element allocation method: 0 = GETCELL 1 = GETMAIN .XXX XXXX Reserved
		IPAVBL	1... .. This message is available
		IPPRMPT	.1.. This message is prompted for
		IPPRTL	..1. This is a partial message
			...X XXXX Reserved
10 (A)	1	IPCCC	Control character count
11 (E)	1	IPTRMTYP	Terminal type
12 (C)	2	*	Reserved
14 (E)	2	IPNXBFSZ	Size of next message
16 (10)	2	IPNXFLGS IPNXALC	Flags of next message: X... .. Element allocation method: 0 = GETCELL 1 = GETMAIN .XXX XXXX Reserved
		IPNXAVBL	1... .. The following message is available
		IPNXPMT	.1.. The following message is prompted for

<u>Cff-</u> <u>set Length</u>	<u>Name</u>	<u>Description</u>
	IPNXPRTL	..1. The following message is a partial message ...X XXXX Reserved
18(12) 1	IPNXCCC	Control character count of next message
19(13) 1	IPNXTMTP	Terminal type associated with next message

Output Queue Manager Parameter List

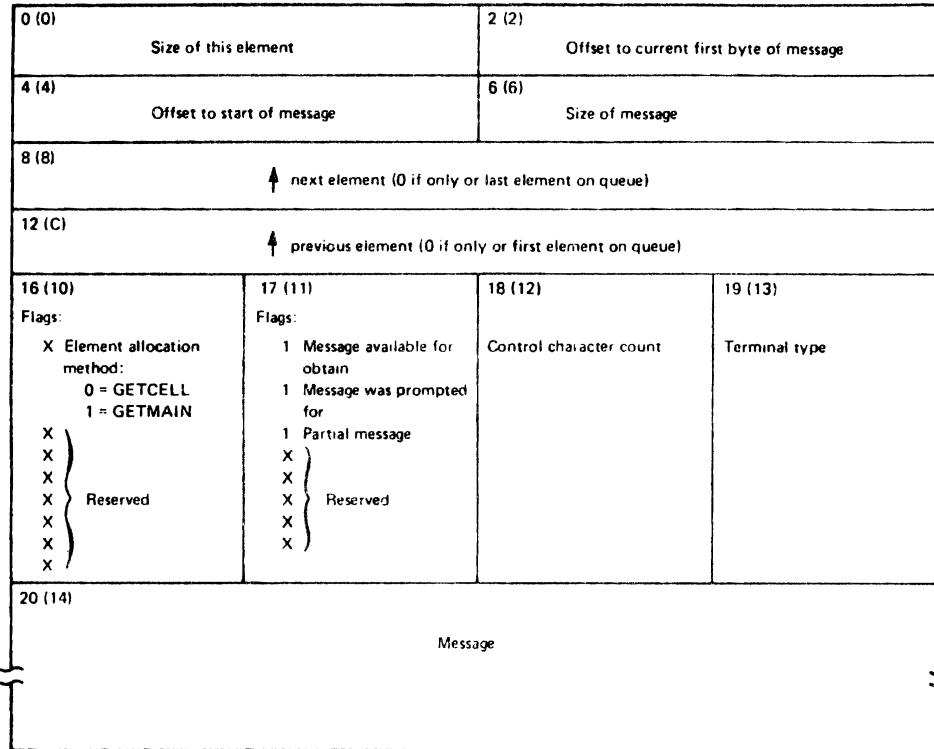
The output queue manager parameter list is used by VTIOC routines that request output queue manager services. It is used to pass parameters to the output queue manager and to receive parameters from the output queue manager. Macro IKIOPARM (acronym OPARMS) maps the parameter list.

<u>Cff-</u> <u>set Length</u>	<u>Name</u>	<u>Description</u>
0(0) 4	OPBUPADR	Address of queue element's message area
4(4) 1	OPREQ	Request code: 1 = Add element to bottom of queue 2 = Obtain message from next element 3 = Remove element from queue 4 = Update an element 5 = Remove all elements 6 = Reset next-element pointer to point to top of queue and obtain message from next element 7 = Reset next-element pointer to point to top of queue 8 = Add element to top of queue
5(5) 1	OPRC	Return code: 1 = Successful (request code 1, 4, or 8) or successful and there are more elements on the queue (request code 2 or 6) 2 = Unsuccessful (request code 1, 4, or 8) or successful and this is last element on queue (request code 2 or 6) 3 = Unsuccessful (request code 2 or 6)
6(6) 2	OPBFSZ	Message length
8(8) 1	OPFLAGS OPALLCC OPADTOP	Flags: X... Element allocation method: 0 = GETCELL 1 = GETMAIN ..1... This element added to top of queue ..XX XXXX Reserved
9(9) 1	OPOFTMS OPHOLD	Options: X... 0 = TPUT NOHOLD, 1 = TPUT HOLD

<u>Cff-</u> <u>set Length</u>	<u>Name</u>	<u>Description</u>
	OPBRK	.X... 0 = TPUT NOBREAK, 1 = TPUT BREAK
	OPEDIT	..XX 00 = TPUT EDIT, 01 = TPUT ASIS, 10 = TPUT CONTROL, 11 = TPUT FULLSCR
	OPASID 1... TPUT ASIDXXX Reserved
10 (A) 4	*	Reserved
14 (E) 2	OPNIBFSZ	Size of next message
16 (10) 1	OPNYPLGS	Flags of next message:
	OPNYALC	X... Element allocation method: 0 = GETCELL 1 = GETMAIN
	OPNYADTP	.1... This element added to top of queue ..XX IXXX Reserved
17 (11) 1		Options of next message:
	OPNXHOLD	X... 0 = TPUT NOHOLD, 1 = TPUT HOLD
	OPNXBRK	.X... 0 = TPUT NOBREAK, 1 = TPUT BREAK
	OPNXEDIT	..XX 00 = TPUT EDIT, 01 = TPUT ASIS, 10 = TPUT CONTROL, 11 = TPUT FULLSCR
	OPNXASID 1... TPUT ASIDXXX Reserved
18 (12) 2		Reserved

Input Queue Element

Input queue elements make up the input queue. They are used by the queue manager to handle input data.



Output Queue Element

Output queue elements make up the output queue. They are used by the queue manager to handle output data.

0 (0)	Size of this element			2 (2)	Offset to current first byte of message		
4 (4)	Offset to start of message			6 (6)	Size of message		
8 (8)	↑ next element (0 if only or last element on queue)						
12 (C)	↑ previous element (0 if only or first element on queue)						
16 (10)	17 (11)		18 (12)		19 (13)		
Flags: X Element allocation method: 0 = GETCELL 1 = GETMAIN 1 Add to top of queue X) X) X) X) X) X) X)	Flags: X 0 = TPUT NOHOLD 1 = TPUT HOLD X 0 = TPUT NOBREAK 1 = TPUT BREAKIN XX 00 = TPUT EDIT 01 = TPUT ASIS 10 = TPUT CONTROL 11 = TPUT FULLSCR 1 TPUT ASID X) X) X)		Reserved		Terminal type		
20 (14)	Message						



VTIOC DIAGNOSTIC AIDS

This section contains information that can be used to diagnose VTIOC problems. It contains:

- Message lists containing the names of object modules that detect the conditions requiring messages, issue the messages, and contain the message text
- An ABEND codes list containing the names of object modules that detect the conditions requiring ABEND codes and issue the codes

Messages

The following messages are issued by VTIOC object modules. For explanations of the system messages see OS/VS Message Library: VS2 System Messages, GC38-1002. For explanations of the terminal messages see OS/VS2 TSO Terminal Messages Directory, SY28-0654.

System Messages

<u>Message</u> <u>IE-----</u>	<u>Module</u> <u>Detecting</u>	<u>Module</u> <u>Issuing</u>	<u>Module</u> <u>Containing</u>
IKT100I	IKTLTERM	IKTLTERM	IKTMSGSG
IKI101I	IKTLTERM	IKTLTERM	IKTMSGSG
IKI102I	IKTLTERM	IKTLTERM	IKTMSGSG
IKT103I	IKTLTERM	IKTLTERM	IKTMSGSG
IKT104I	IKTXINIT	IKTXINIT	IKTMSGSG
IKI105I	IKTXINIT	IKTXINIT	IKTMSGSG
IKT106I	IKTXINIT	IKTXINIT	IKTMSGSG
IKI107I	IKTLTERM	IKTLTERM	IKTMSGSG

Terminal Messages

<u>Message</u> <u>IE-----</u>	<u>Module</u> <u>Detecting</u>	<u>Module</u> <u>Issuing</u>	<u>Module</u> <u>Containing</u>
IKTC0300I	IKTLTERM	IKTLTERM	IKTMSGSG
IKTC0301I	IKTLOGR	IKTLOGR	IKTMSGSG
IKTC0400I	IKTIMIDS IKTTIMRT IKTQMIN IKTIMLU2 IKT3270I	IKTOMIDS IKTTOMRT IKTQMIN IKTCMLU2	IKTMSGSG
IKTC0401I	IKTQMIN	IKTQMIN	IKTMSGSG
IKTC0402I	IKTTIMRT	IKTTOMRT	IKTMSGSG
IKTC0403I	IKTTCMRT	IKTTOMRT	IKTMSGSG
IKTC0405I	IKTOMIDS IKTIMLU2 IKTOMLU2	IKTOMIDS IKTCMLU2	IKTMSGSG

Abend Codes

The following abend codes are issued by VTIOC object modules. For explanations of the codes see OS/VS Message Library: VS2 System Codes, GC38-1008.

<u>Code</u>	<u>Module Detecting</u>	<u>Module Issuing</u>
0AB	IKTIMIDS	IKTIMIDS
	IKTIMLU2	IKTIMLU2
	IKTLOGPF	IKTLOGPF
	IKTILTERM	IKTILTERM
	IKTOMIDS	IKTOMIDS
	IKTOMLU2	IKTOMLU2
	IKTIMRT	IKTIMRT
	IKTOMRT	IKTOMRT
	IKTXINIT	IKTXINIT
0AD	IKTQMIN	IKTQMIN



TCAS INTRODUCTION

The terminal control address space (TCAS) accepts logon requests from TSO/VTAM users and creates an address space for each user (see Figure 1). When the system operator issues a START command to start TSO/VTAM time sharing, an address space is created for TCAS as a VTAM application program. Then, when a user attempts to log on to TSO/VTAM time sharing, TCAS accepts the logon request, creates an address space having a unique VTAM application ID for the user, and transfers control of the terminal to the newly created address space. From this point on communication goes from the terminal through VTAM to the user's address space (that is, to TSO via the VTAM terminal I/O coordinator (VTIOC)).

TCAS has four functional components:

- **TCAS main task.** The main task processes the START command, handles TCAS initialization and termination, creates the TCAS table (TCAST), processes START and MODIFY command parameters, and attaches the three TCAS subtasks.
- **VTAM interface subtask.** This subtask opens and closes ACBs, accepts and quiesces logons, transfers and releases terminal control, and provides VTAM exit routines (logon, TPEND, CLSDST, and Bind).
- **User interface subtask.** This subtask creates and terminates user address spaces, and adds TSPs to the TSO/VTAM TSB chain.
- **Console communication subtask.** This subtask processes the MODIFY and STOP commands.

TCAS carries out its functions by performing units of work that are represented by work elements (see "TCAS Data Areas" for a description of a work element). The work elements are queued and thus represent a series of work to be done by a task. (The one exception is VTAM exit routines, which are not queued as work elements but are scheduled by VTAM.) There are two types of queues: synchronous and asynchronous. There is one synchronous queue for all the TCAS tasks. There are one or more asynchronous queues for each TCAS task.

Each TCAS task, when its ECB is posted, looks for work on the synchronous queue. If the synchronous queue is empty, it looks for work on one of the asynchronous queues. When each task finishes its work it enters a wait state, or posts another task if any work was created for one.

The synchronous queue contains work elements to handle work requiring synchronization of all the TCAS tasks. Elements on the synchronous queue are created to process the START command, the MODIFY command, the STOP command, and error recovery. The asynchronous queues consist of the ACB control queue, the global terminal handling queue, the local terminal handling queue, the TPEND queue, the address space creation queue, the STOP command queue, and the MODIFY command queue.

When the work represented by the first work element on the synchronous queue is finished, the task that executed the first work element examines the second work element to determine which task should receive control next. The first work element is dequeued and the task of the second work element is posted. This processing is propagated until all the work on the synchronous queue is finished.

When the work represented by the first work element on an asynchronous queue is finished, the task that executed the first work element dequeues it and executes the next element. This processing is propagated until all the work on an asynchronous queue is finished.

TCAS METHOD OF OPERATION

This section uses the HIPO (hierarchy plus input-process-output) technique to graphically describe the functions performed by TCAS. It contains a visual table of contents and diagrams.

The visual table of contents (Figure 11) contains the names and identification numbers of all the diagrams. There is a diagram for each major TCAS function. The diagrams show functions performed and significant inputs and outputs; they do not show organization of a routine or processing flow within a routine.

Conventions Used in HIPO Diagrams

At the top of each diagram is a diagram ID consisting of an identification number (MO number) and a description of the routine. The identification number provides a way of locating a diagram through the visual table of contents or the directory ("TCAS Directory"). Below the diagram ID are input-process-output blocks and extended description blocks.

The input block shows data that serves as input to the processing steps in the processing block; the output block shows data that is output from the processing steps. The symbols used in and between these blocks are explained in the legend (Figure 2). Each processing step is numbered; the number corresponds to a note in the extended description block. The notes provide additional information for the processing steps. The routine name and labels identify the code that performs the function of each step.

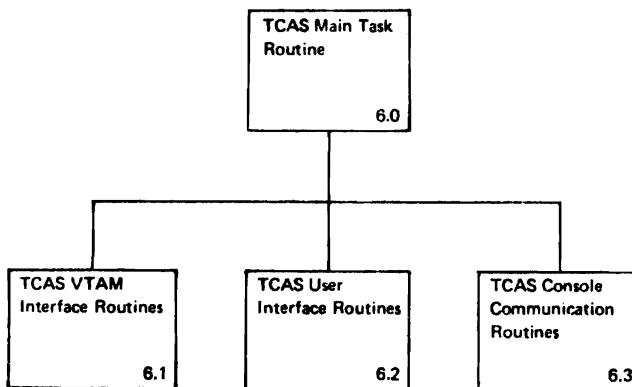
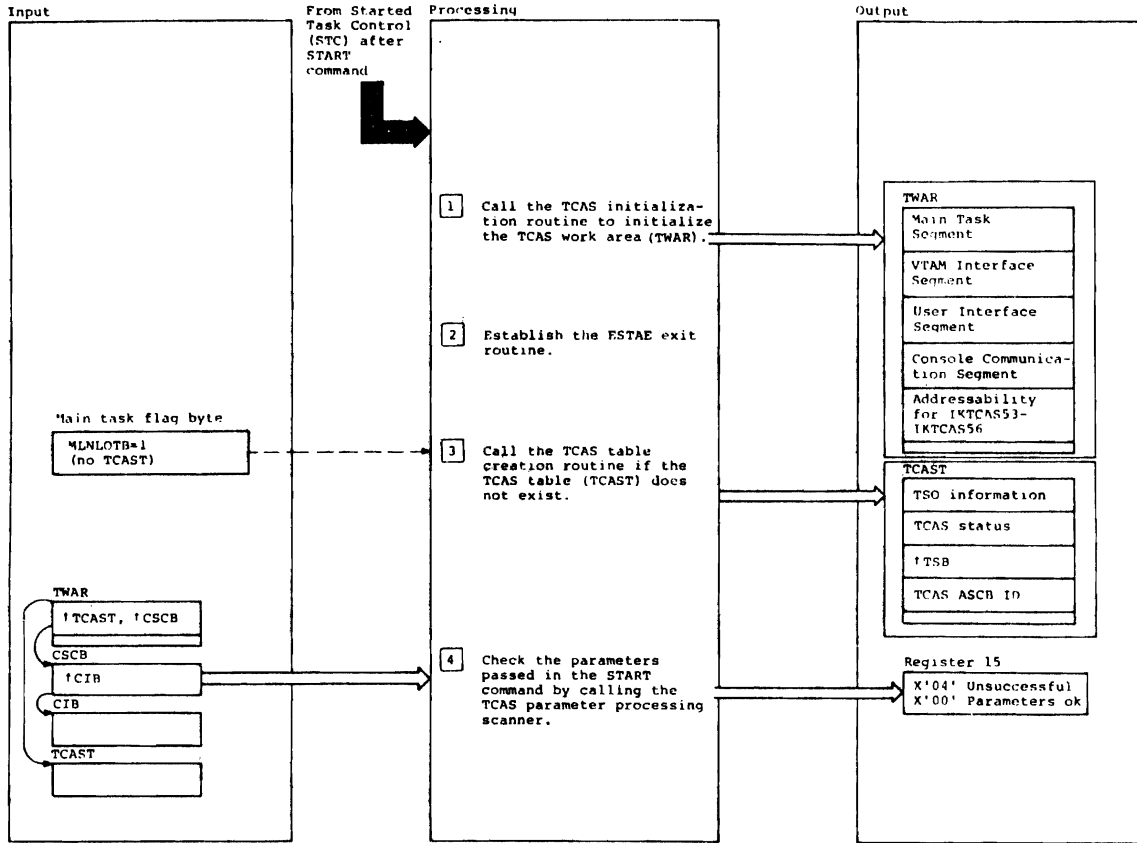


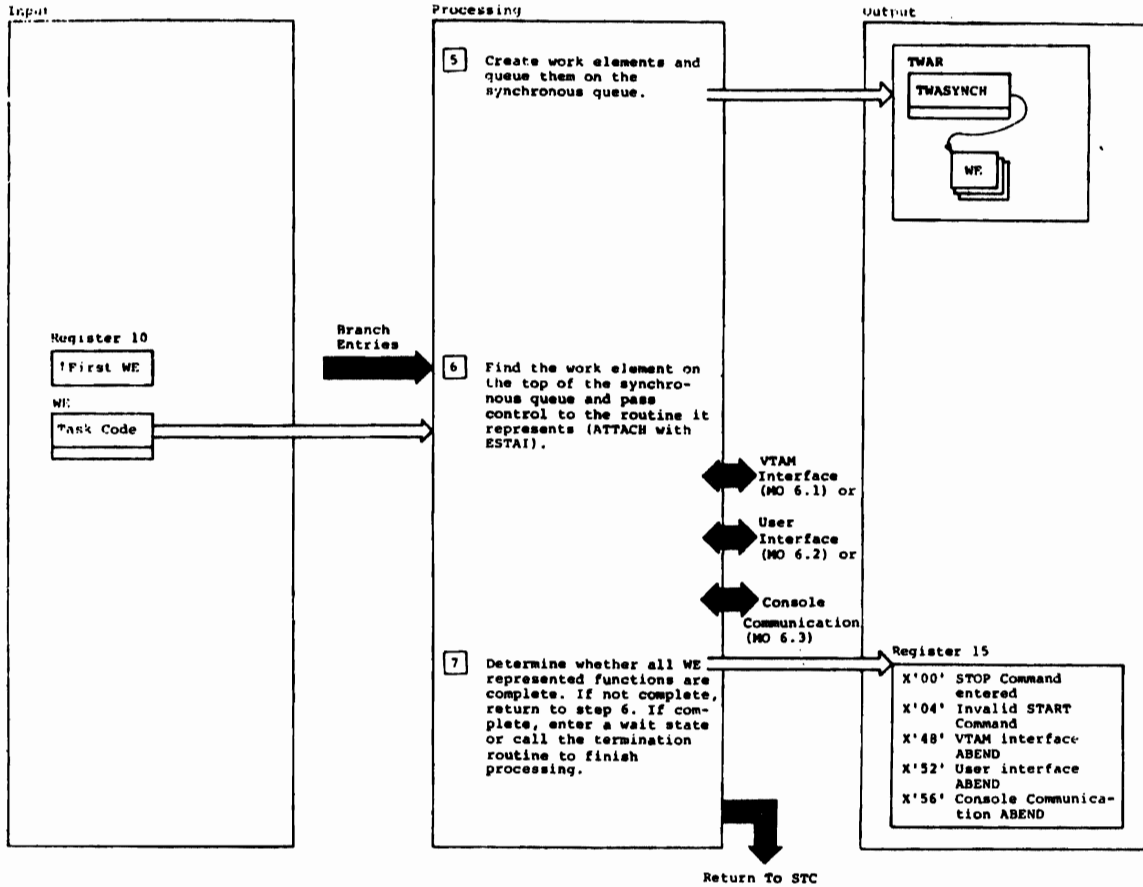
Figure 11. TCAS Visual Table of Contents

MO 6.0 (Part 1 of 2) TCAS Main Task



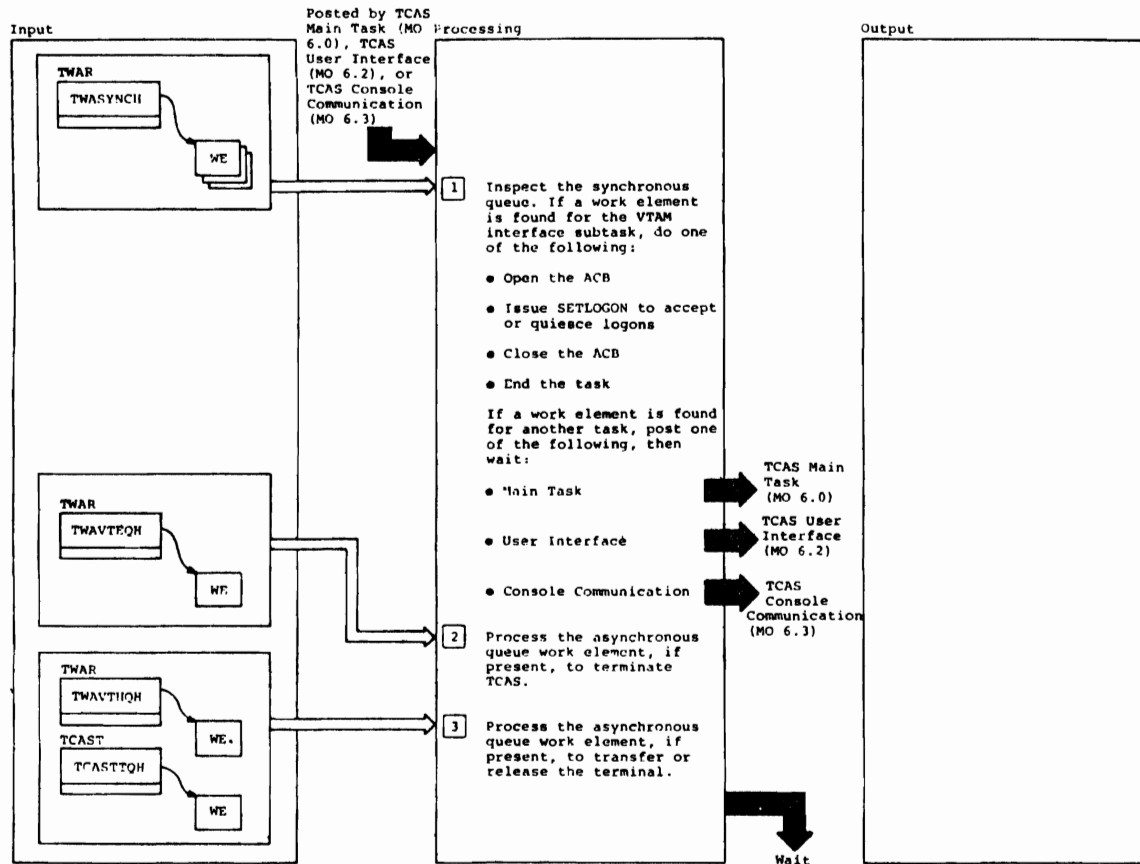
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
The TCAS main task processes the START command and coordinates the TCAS subtasks.	IKTCAS00			<p>4 The parameter processing scanner routine scans the parameters passed from the START command, reads the parameter member from parmlib, updates the TCAST with the parmlib parameters, and writes the parameters used to SYSOUT.</p>	IKTCAS54	SCANPARM	
1 The initialization routine allocates storage for the TWAR, initializes it, and establishes addressability for other TCAS routines by storing their addresses in the TWAR.	IKTCAS51					READMBR	
2 The ESTAE exit routine receives control from R/TM. It performs the following functions: <ul style="list-style-type: none"> • Passes control back to R/TM for cleanup • Passes control back to R/TM to retry the failing task • Creates a work element to reattach the failing task if the task has not previously abended • Creates work elements to terminate TCAS if the main task or a previously abended subtask is failing. 	IKTCAS55				OVERLAY		
3 The TCAS table creation routine allocates storage for the TCAST and initializes it.	IKTCAS53						

MO 6.0 (Part 2 of 2) TCAS Main Task



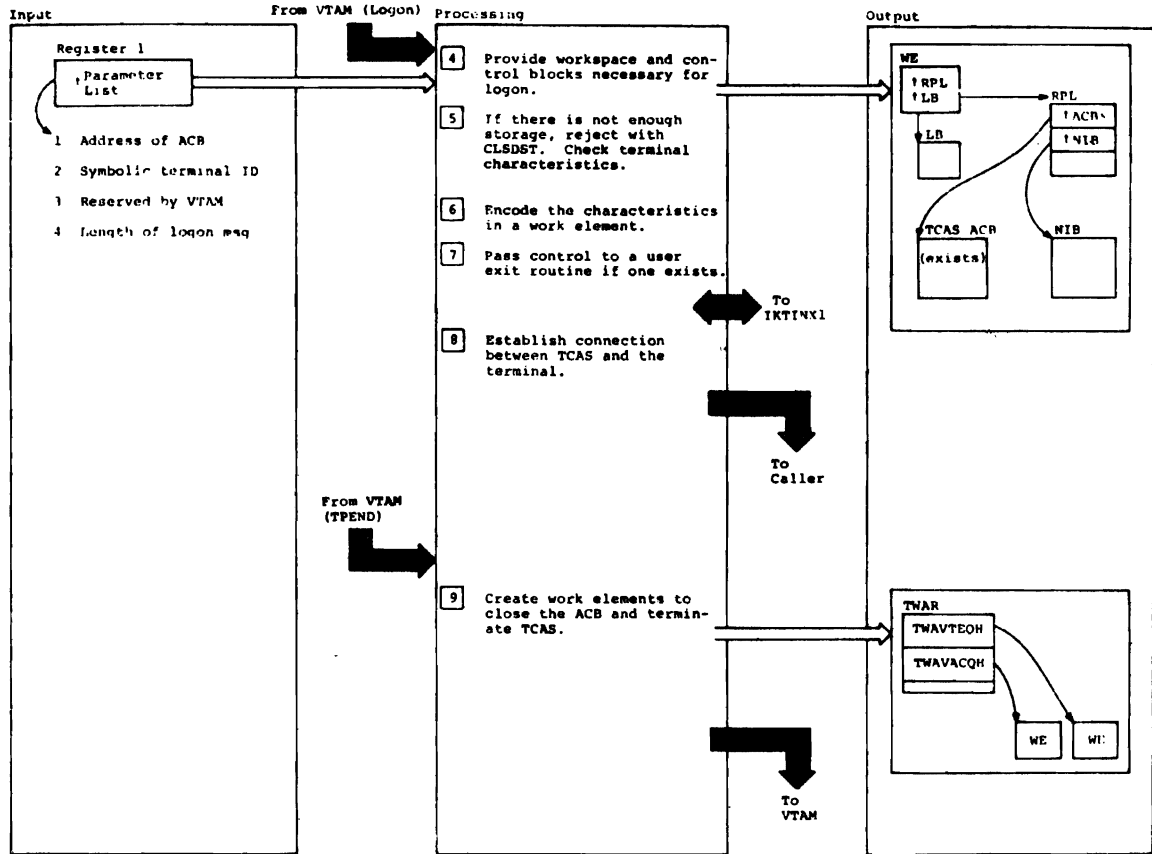
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>5 The sequence of work elements is as follows:</p> <ul style="list-style-type: none"> A. Attach user interface driver. B. Terminate the address space (if any TSO users are active and the operator wishes to terminate them). C. Attach VTAM interface driver. D. Attach console communication driver. E. Start logon. F. Return. 	IKTCAS00			<p>7 If the system operator has entered STOP TSO, the termination service routine is called to perform final processing. It issues the SDUMP macro to provide a virtual storage dump if the system operator replies DUMP to termination message IKT012D. It then detaches the VTAM interface, user interface, and console communication sub-tasks, and frees cross-memory requests on the work element queue by posting the ECBA specified in the work elements and by freeing work element storage.</p>	IKTCAS52		
<p>6 The subroutine that selects work elements from the top of the synchronous queue posts each work element in order. After the routine represented by the work element finishes its work, it returns to MLESLT in the main task.</p>	IKTCAS00	MLESLT					

MO 6.1 (Part 1 of 2) TCAS VTAM Interface



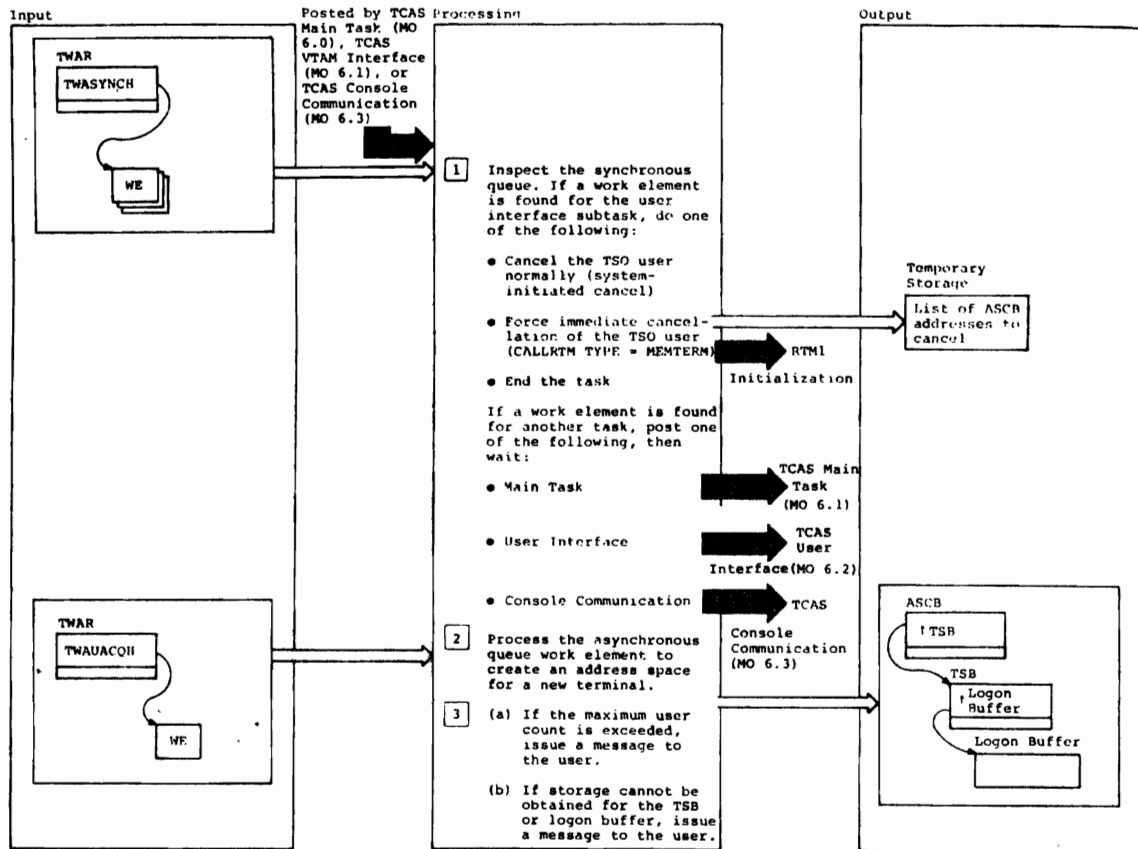
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>1 The TCAS VTAM interface subtask inspects the TCAS queues in this order:</p> <p>(1) Synchronous queue</p> <p>(2) TPEND queue</p> <p>(3) Terminal handling queue.</p> <p>It gives control to a routine represented by a work element on one of the above queues.</p> <p>If the request cannot be processed by IKTCAS21, the function code is changed to the originator's function code to give control back to the originator.</p> <p>If IKTCAS21 processes the request successfully, it sends back an end of task work element which IKTCAS20 dequeues. IKTCAS20 then posts the user interface and console communication routines to see if they have any work.</p>	IKTCAS20			<p>Based on the work element function code, IKTCAS22 calls IKTCASCX (Release/Pass Asynchronous Exit), which</p> <p>(1) Verifies the completion of a session with a terminal</p> <p>or</p> <p>(2) Verifies the completion of a request to pass control of the terminal to another application program.</p> <p>If either operation is not complete, IKTCASCX can initiate a retry.</p> <p>If control of a terminal was passed, IKTCASCX frees the control blocks allocated in step 4.</p>	IKTCASCX		
<p>2 A TPEND work element is added to the asynchronous TPEND queue if the synchronous queue is not empty.</p>	IKTCAS24						
<p>3 IKTCAS20 calls IKTCAS22 to process work elements on the global and local terminal handling queues.</p>	IKTCAS22						

MO 6.1 (Part 2 of 2) TCAS VTAM Interface



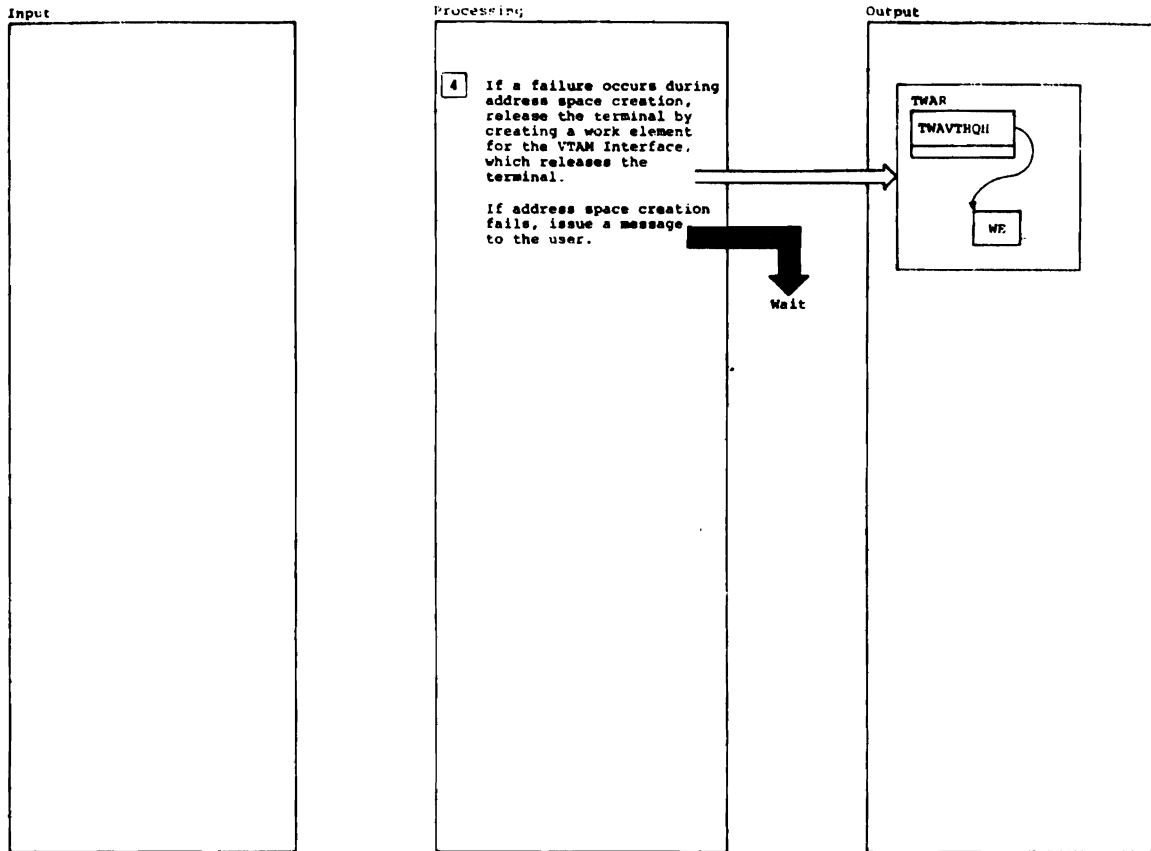
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
4 Get storage for and begin to initialize the WE, LB, RPL, and NIB.	IKTCAS23	RPLENTRY		the synchronous queue is not empty, the TPEND exit routine creates a TPEND work element and queues it on the TPEND asynchronous queue.			
5 IKTCAS23 checks device characteristics by issuing an INQUIRE macro to obtain session parameters and by comparing these parameters to the terminal characteristics. If the request is in error, a CLSDST RELEASE macro is issued to tell VTAM to disconnect the terminal. When the request is honored, IKTCASCX receives control to free the control blocks allocated in step 4.	IKTCAS23	DEVERIFY					
		TERMINAT					
	IKTCASCX						
7 The user exit routine, IKTINX1, can halt logon processing if it finds an invalid device.	IKTINX1						
	IKTCAS23	RPLINTO					
	IKTCASOX						
	IKTCAS20						
9 IKTCAS20 dequeues the last work element on the asynchronous queue and calls IKTCAS24, the TCAS VTAM TPEND exit routine. If the synchronous queue is empty, work elements are put on the synchronous queue to close the ACR and terminate TCAS. If	IKTCAS24	ADDORQ					

MO 6.2 (Part 1 of 2) TCAS User Interface



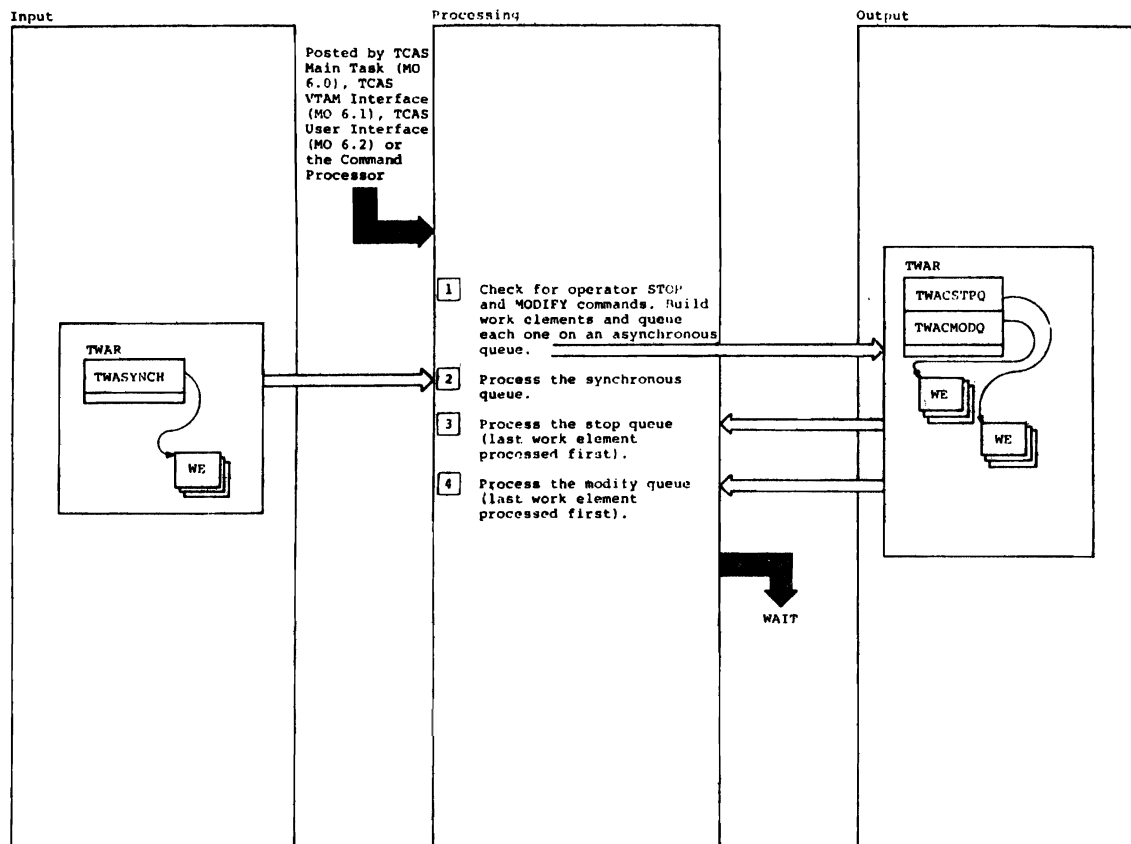
Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>1 The TCAS user interface subtask inspects the TCAS queues in this order:</p> <p>(1) Synchronous queue.</p> <p>(2) Address space creation queue (address space for use by a terminal).</p> <p>IKTCAS30 gives control to a routine represented by a work element on one of the above queues.</p> <p>To free user address spaces, IKTCAS30 calls IKTCAS32, the address space termination routine, which gets temporary storage to list the address of each ASCB.</p> <p>IKTCAS32 calls IKTCAS33, the TSB manipulation routine, which lists the ASCB addresses in the temporary storage. IKTCAS32 then issues the CALLRTM macro instruction to terminate the users' address spaces.</p> <p>If the request is to issue a system cancel, IKTCAS30 loads each ASCB address and calls SIC to cancel each user.</p>	IKTCAS30			<p>2 IKTCAS30 dequeues the work element and then calls IKTCAS31, the address space creation routine, which</p> <p>(a) Check maximum user count.</p> <p>(b) Gets an address space by issuing SVC 34.</p> <p>(c) Initializes an ASCB, a TSB, a TSBX, and a logon buffer.</p> <p>(d) Calls IEDAY3 to synchronize the new address space initialization.</p> <p>(e) Calls IKTCAS33, the TSB manipulation routine, to queue a TSB.</p>	IKTCAS31		
	IKTCAS32				IKTCAS31	TELLUSER	
	IKTCAS33						

MO 6.2 (Part 2 of 2) TCAS User Interface



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
<p>4 After IKTCAS31 returns to IKTCAS30, if address space creation failed, IKTCAS30 posts the VTAM interface.</p>	IKTCAS31	RELTERM					

MO 6.3 TCAS Console Communication



Notes	Routine	Label	Ref	Notes	Routine	Label	Ref
The console communication subtask handles the STOP and MODIFY operator commands. Other TCAS functions can use this routine by placing a work element on either the STOP queue or the MODIFY queue.	IKTCAS40			elements are built to cause an orderly shutdown of TCAS:			
1 After using a CIB (command input buffer) to build a work element, IKTCAS40 uses the QEDIT macro to free the CIB.	IKTCAS40	CHKECBS BLDSTPWE BLDMODWE		<ul style="list-style-type: none"> • A WE for the VTAM interface to close the ACB. • A WE for the user interface to terminate the address space. • A WE to tell about the termination request. 			
2 If the synchronous queue is empty, continue processing with step 3.	IKTCAS40	CHKSYNCO		4 IKTCAS40 calls IKTCAS42 to handle work elements on the modify queue by building a set of work elements for the synchronous queue that cause the modifications to be made. Parameters in the work element extension show what type of work elements should be built. When finished building work elements, IKTCAS42 posts the task indicated by the top work element on the synchronous queue.	IKTCAS42	PRCSMODE	
If the first work element is not for console communication, post the owner and wait.		POSTIT					
If the first work element is for console communication, process work elements for normal return, end of task, and abnormal end.		PRCSSYNE					
If the queue is empty now, post the other TCAS tasks and exit.							
Repeat processing if the new first work element is for console communication.							
3 IKTCAS40 calls IKTCAS41 to handle work elements on the STOP queue by building work elements on the synchronous queue; the following work	IKTCAS41	PRCSSPTE					

TCAS PROGRAM ORGANIZATION

This section describes TCAS modules and shows control flow among them. There are two parts:

- A figure (Figure 12) showing control flow among the TCAS modules
- A description of each TCAS module giving:

Names

MO Diagram - in which the module is described

Function

Entry from

Registers at Entry - contents of relevant registers

Exit to

Registers at Exit - contents of relevant registers

External References

Control Blocks Used

Mapping Macros Used

Executable Macros Used

Module Attributes

Lock Dependency - requirements at module entry

Messages - detected by and issued from each module

Abend Codes

Data Sets

The module descriptions are arranged alphanumerically by module name.

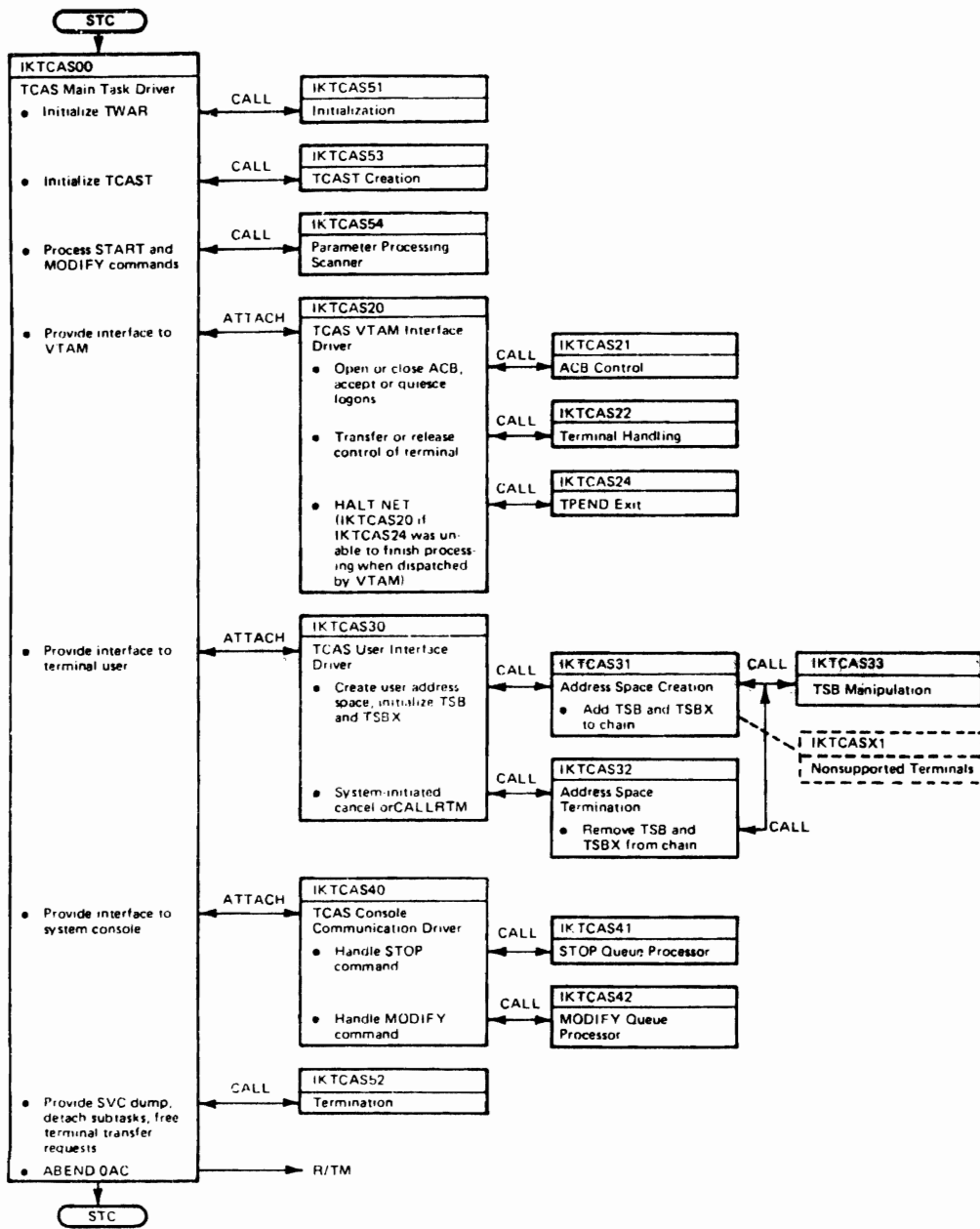


Figure 12 (Part 1 of 2). TCAS Module Flow

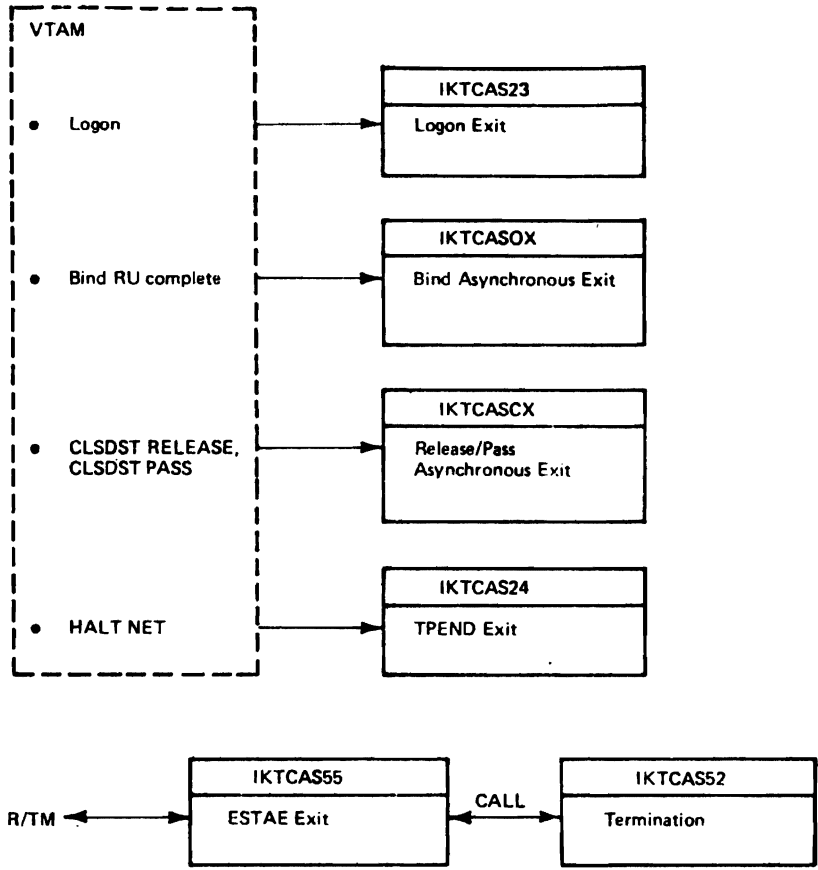


Figure 12 (Part 2 of 2). TCAS Module Flow

IKTCASCX -- Release/Pass Asynchronous Exit Routine

Names:

Assembly Module: IKTCASCX
Object Module: IKTCASCX
Alternate Entry Points: Ncne
Load Module: IKTCAS20

MC Diagram: 6.1

Function: VTAM schedules this routine upon completion of either a request to terminate a terminal session or a request to pass control of the terminal from the user's address space to another address space. IKTCASCX verifies the completion of the request and frees the space allocated to the terminal.

Entry from: VTAM

Registers at Entry:

Register 1: Address of RPL
Register 14: Return address
Register 15: Entry point address

Exit to: VTAM

Registers at Exit:

Register 1: Address of RPL
Register 14: Return address
Register 15: Entry point address

External References: IKTCAS56

Control Blocks Used: CVT, NIP, PSA, RPL, TCAST, TWAR, WESTD

Mapping Macros Used: CVT, IFGRPL, IFGRPLVT, IHAPSA, IKTCASWA IKTTCAST, IKIWESTD, ISTNIB, ISTRPLFB

Executable Macros Used: CHECK, CLSDST, EXECRPL, FREMAIN, POST, WTO

Module Attributes: Supervisor state, key 6, serially reusable

Lock Dependency: None

Messages: IKT019I

Abend Codes: None

Data Sets: None

IKTCASCX -- Bind Asynchronous Exit Routine

Names:

Assembly Module: IKTCASOX
Object Module: IKTCASCX
Alternate Entry Points: Ncne
Load Module: IKTCAS20

MC Diagram: 6.1

Function: VTAM uses an IRB to dispatch this routine after VTAM receives notice of completion of an asynchronous Bind RU to a terminal attempting to log on to TSO/VTAM. If the Bind or a retry of the Bind is successful, IKTCASOX requests an address space for the terminal. If the Bind is unsuccessful, the terminal is freed.

Entry_from: VTAM

Registers_at_Entry:

Register 1: Address of RPL
Register 14: Return address
Register 15: Entry point address

Exit_to: VTAM

Registers_at_Exit:

Register 1: Address of RPL
Register 14: Return address
Register 15: Entry point address

External References: IKTCASOX

Control Blocks Used: CVT, NIB, PSA, RPL, TCAST, TWAR, WESTD

Mapping Macros Used: CVT, IFGRPL, IFGRPLVT, IHAPSA, IKTCASWA, IKTTCAST, IKTWESID, ISTNIB, ISTRPLFB

Executable Macros Used: CHECK, CLSDST, EXECRPL, FREEMAIN, POST, WTO

Module Attributes: Supervisor state, key 6, serially reusable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTCASX1 -- Installation-Written Exit for Nonsupported Terminals

Names:

Object Module: IKTCASX1
Load Module: IKTCAS30

Function: An installation must write this routine if it wants 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.

Entry_from: IKTCAS31

Registers_at_Entry:

Register 1: Address of the message text, preceded by a full word that indicates the total length of the text in bytes.
Register 0: Address of the RU buffer into which the edited message is put.

Register 14: Return address
Register 15: Entry point address

Exits_to: IKTCAS 31

Registers_at_Exit:

Register 0: Address of the output RU
Register 1: Total length of the RU indicated by the EXIT routine
Register 15: Return codes:

0 - IBM routine should perform editing.
Nonzero - Exit routine performed all editing.

IKTCAS00 --- TCAS Main Task Driver

Names:

Assembly Module: IKTCAS00
Object Module: IKTCAS00
Alternate Entry Points: Ncne
Load Module: IKTCAS00

MC_Diagram: 6.0

Function: This routine processes the START command by calling the initialization, TCAST creation, and parameter processing scanner routines. It creates work elements as necessary for user interface processing, console communication, logon processing and START command completion. It calls the routines that these work elements represent. It also calls the termination routine if a termination work element is reached, or if there is an abnormal termination.

Entry_from: Started task control (attached after START command)

Registers_at_Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to:

Started task control: Normal
R/TM: ABEND OAC

Registers_at_Exit:

Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

- 00 STOP command entered
- 04 Invalid START command
- 10 TCAS VTAM interface subtask not attached
- 14 TCAS user interface subtask not attached
- 18 TCAS console communication subtask not attached
- 1C Storage for TCAST not available
- 30 TCAS VTAM interface subtask abend
- 34 TCAS user interface subtask abend
- 38 TCAS console communication subtask abend
- 3C TPEND occurred

External References: IKTCAS51, IKTCAS52, IKTCAS53, IKTCAS54, IKTCAS56
Control Blocks Used: ASCB, CIP, CSCB, CVT, PSA, TCAST, TWAR
Mapping Macros Used: CVT, IECHAIN, IEZCIB, IHAASCB, IHAPSA, IKTCASWA, IKTTCAST
Executable Macros Used: ABEND, ATTACH, ESTAB, FREEMAIN, GETMAIN, LOAD, MCDESET, POST, SETLOCK, WAIT, WTO, WTOR
Module Attributes: Serially reusable
Lock Dependency: None
Messages: IKT001D, IKT002I, IKT003D, IKT004D, IKT005I, IKT006D
Abend Codes: OAC
Data Sets: None

IKTCAS20 -- TCAS VTAM Interface Driver

Names:

Assembly Module: IKTCAS20
Object Module: IKTCAS20
Alternate Entry Points: Ncne
Load Module: IKTCAS20

MC Diagram: 6.1

Function: This routine processes work element queues for the VTAM interface subtask by giving control to routines indicated by work element function codes.

Entry from: IKTCAS00

Registers at Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTCAS00

Registers at Exit:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External References: IKTCAS21, IKTCAS22, IKTCAS24

Control Blocks Used: TCAST, TWAR

Mapping Macros Used: IKTCASWA, IKTTCAST

Executable Macros Used: POST, WAIT

Module Attributes: Serially reusable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTCAS21 -- ACB Control Routine

Names:

Assembly Module: IKTCAS21
Object Module: IKTCAS21
Alternate Entry Points: None
Load Module: IKTCAS20

MO Diagram: 6.1

Function: This routine opens the terminal's ACB, issues the SETLOGON macro to accept or quiesce logons, and closes the ACB.

Entry from: IKTCAS20

Registers at Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTCAS20

Registers at Exit:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External References: IKTCAS24, IKTCAS56

Control Blocks Used: ACB, EXLST, RPL, TCAST, THAR

Mapping Macros Used: IFGACB, IFGEXLST, IFGRPL, IKTCASWA, IKTTCAST

Executable Macros Used: CLOSE, OPEN, SETLOGON, SHOWCB, WTO

Module Attributes: Serially reusable

Lock Dependency: None

Messages: IKT007I, IKT008I

Abend Codes: None

Data Sets: None

IKTCAS22 -- Terminal Handling Routine

Names:

Assembly Module: IKTCAS22
Object Module: IKTCAS22

Alternate Entry Points: None
Load Module: IKTCAS20

MC Diagram: 6.1

Function: This routine transfers terminal control from TCAS to the user's address space (CLSDST FASS) or releases terminal control (CLSDST RELEASE).

Entry from: IKTCAS20

Registers at Entry:

Register 10: Address of work element
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTCAS20

Registers at Exit:

Register 10: Address of work element
Register 13: Address of register save area
Register 14: Return Address
Register 15: Entry point address

External References: IKTCASCY

Control Blocks Used: RPL, WESTD

Mapping Macros Used: IFGRPL, IFGRPLVT, IKTWESTD, ISTRPLFB

Executable Macros Used: CLSDST, FREEMAIN

Module Attributes: Serially reusable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTCAS23 -- Logon Exit Routine

Names:

Assembly Module: IKTCAS23
Object Module: IKTCAS23
Alternate Entry Points: None
Load Module: IKTCAS20

MC Diagram: 6.1

Function: This routine provides the work space and control blocks necessary for processing a logon request.

Entry from: VTAM (dispatched as a logon exit)

Registers at Entry:

Register 1: Address of parameter list containing:

- Address of the terminal's ACB
- Symbolic terminal ID
- Reserved
- Length of the logon message

Register 14: Return address

Register 15: Entry point address

Exit to: VTAM

Registers at Exit:

Register 1: Address of parameter list containing:

- Address of the terminal's ACB
- Symbolic terminal ID
- Reserved
- Length of the logon message

Register 14: Return address

Register 15: Entry point address

External References: IKTCASCX, IKTCASOX, IKTCAS56, IKTIMX1

Control Blocks Used: BIND, CVT, NIB, PSA, RPL, TCAST, TWAR, WESTD

Mapping Macros Used: CVT, IFGRPL, IFGRPLVT, IHAPSA, IKTCASWA, IKTTCAST, IKTWESTD, ISTNIB, ISTRPLFB

Executable Macros Used: CLSDST, EXECRPL, GETHAIN, INQUIRE, OPNDST, WTC

Module Attributes: Serially reusable, key 6, supervisor state

Lock Dependency: None

Messages: IKTO19I

Abend Codes: None

Data Sets: None

IKTCAS24 -- TPEND Exit Routine

Names:

Assembly Module: IKTCAS24
Object Module: IKTCAS24
Alternate Entry Points: None
Load Module: IKTCAS20

MC Diagram: 6.1

Function: This routine initiates TCAS termination.

Entry from: VTAM (normal) or IKTCAS20 (if processing was not completed after normal entry)

Registers at Entry:

Register 1: Address of a parameter list (on entry from VTAM) containing:

- ACB address of the terminating program
- Reason for the termination
- or
- Zero (on entry from IKTCAS20)

Register 10:

Irrelevant (on entry from VTAM)
or
Address of TPEND work element on asynchronous queue (on entry from IKTCAS20)

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: Caller

Registers at Exit:

Register 1: Address of a parameter list (on entry from VTAM) containing:

- ACB address of the terminating program
- Reason for the termination
- or
- Zero (on entry from IKTCAS20)

Register 10:

Irrelevant (on entry from VTAM)
or
Address of TPEND work element on asynchronous queue (on entry from IKTCAS20)

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

00 Normal return

External References: IKTCASCY, IKTCASOX, IKTCAS56, IKTWAPTR (in IKTCAS20)

Control Blocks Used: TWAR

Mapping Macros Used: IKTCASWA

Executable Macros Used: GETMAIN, WTO

Module Attributes: Serially reusable

Lock Dependency: None

Messages: IKT009I

Abend Codes: Ncne

Data_Sets: None

IKTCAS30 -- TCAS User Interface Driver

Names:

Assembly Module: IKTCAS30
Object Module: IKTCAS30
Alternate Entry Points: None
Load Module: IKTCAS30

MC_Diagram: 6.2

Function: This routine processes work element queues for the user interface subtask by giving control to routines indicated by work element function codes.

Entry_from: IKTCAS00

Registers_at_Entry:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS00

Registers_at_Exit:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External_References: IKTCAS31, IKTCAS32

Control_Blocks_Used: ECP, TWAR

Mapping_Macros_Used: IHAECB, IKTCASWA

Executable_Macros_Used: POST, WAIT

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: None

Abend_Codes: None

Data_Sets: None

IKTCAS31 -- Address Space Creation Routine

Names:

Assembly Module: IKTCAS31
Object Module: IKTCAS31
Alternate Entry Points: None

Load Module: IKTCAS30

MC_Diagram: 6.2

Function: This routine issues SVC 34 to create a TSO user address space; allocates storage for and initializes the TSB, TSBX, and logon buffer; and synchronizes the initialization of the new address space.

Entry_from: IKTCAS30

Registers_at_Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS30

Registers_at_Exit:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External_References: IKTCASX1, IKTCAS33, IKTCAS56

Control_Blocks_Used: ASCB, ASVT, BASEA, CVT, NIB, RPL, TCAST, TSB, TSEX, TWAR, WESTD

Mapping_Macros_Used: CVT, IEEBASEA, IFGRPL, IHAASCB, IHAASVT, IKJTSB, IKTCASWA, IKTTCAST, IKTWESTD, ISTNIB

Executable_Macros_Used: FREEMAIN, GETAMIN, MODESET, POST, SEND

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: IKT00201I, IKT00202I, IKT00203I

Abend_Codes: None

Data_Sets: None

IKTCAS32 -- Address Space Termination Routine

Names:

Assembly Module: IKTCAS32
Object Module: IKTCAS32
Alternate Entry Points: None
Load Module: IKTCAS30

MC_Diagram: 6.2

Function: This routine calls the system-initiated cancel routine or issues CALLRTM to terminate a user address space.

Entry_from: IKTCAS00

Registers at Entry:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTCAS00

Registers at Exit:

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External References: IKTCAS33, system-initiated cancel routine

Control Blocks Used: ASCB, CVT, TCAST, TWAR

Mapping Macros Used: CVT, IHAASCB, IKTCASWA, IKTTCAST

Executable Macros Used: CALLR/TM, FREEMAIN, GETMAIN, MODESET

Module Attributes: Serially reusable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTCAS33 -- TSB Manipulation Routine

Names:

Assembly Module: IKTCAS33
Object Module: IKTCAS33
Alternate Entry Points: None
Load Module: IKTCAS30

MC Diagram: 6.2

Function: This routine adds TSEs to the TSO/VTAM TSB chain. It also obtains the ASCB addresses of all TSO/VTAM users.

Entry from: IKTCAS31 (to chain a TSB) or IKTCAS32 (to obtain ASCB addresses)

Registers at Entry:

Register 8: Function code (in hex):

31 Chain a TSB
32 Obtain ASCB addresses

Register 9:

Address of TSB (if function code 31)
or
Address of storage for ASCB addresses (if function code 32)

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: Caller

Registers_at_Exit:

Register 8: Function code (in hex):

31 Chain a TSB
32 Obtain ASCB addresses

Register 9:

Address of TSB (if function code 31)
or
Address of storage for ASCB addresses (if function code 32)

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External References: None

Control Blocks Used: CVT, PSA, TCAST, TSB, TSBX, TWAR

Mapping Macros Used: CVT, IHAPSA, IKJTSB, IKTCASWA, IKTTCAST

Executable Macros Used: SETLOCK

Module Attributes: Serially reusable

Lock Dependency: None

Messages: None

Abend Codes: None

Data Sets: None

IKTCAS40 -- TCAS Console Communication Driver

Names:

Assembly Module: IKTCAS40
Object Module: IKTCAS40
Alternate Entry Points:
Load Module: IKTCAS40

MC Diagram: 6.3

Function: This routine processes work element queues for the console communication subtask by giving control to routines indicated by work element function codes.

Entry from: IKTCAS00

Registers_at_Entry:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS00

Registers_at_Exit:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

00 Normal return

External_References: IKTCAS41, IKTCAS42, IKTCAS56

Control_Blocks_Used: CIB, CSCE, TWAR

Mapping_Macros_Used: IECHAIN, IEZCIB, IKTCASWA

Executable_Macros_Used: FREEMAIN, GETMAIN, MODESET, POST, WAIT, WTO

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: IKT011I

Abend_Codes: None

Data_Sets: None

IKTCAS41 -- STOP Queue Processor

Names:

Assembly Module: IKTCAS41
Object Module: IKTCAS41
Alternate Entry Points: None
Load Module: IKTCAS40

MC_Diagram: 6.3

Function: This routine handles work elements on the STOP queue by building a set of work elements for the synchronous queue, causing an orderly shutdown of TCAS.

Entry_From: IKTCAS40

Registers_at_Entry:

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS40

Registers_at_Exit:

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

00 Normal return

External_References: IKTCAS56

Control_Blocks_Used: TCAST, TWAR

Mapping_Macros_Used: IKTCASWA, IKTTCAST

Executable_Macros_Used: FREEMAIN, GETMAIN, POST, WAIT, WTOR

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: IKT010I, IKT016D

Abend_Codes: None

Data_Sets: None

IKTCAS42 -- MODIFY_Queue_Processor

Names:

Assembly Module: IKTCAS42
Object Module: IKTCAS42
Alternate Entry Pcints: Mcne
Load Module: IKTCAS40

MC_Diagram: 6.3

Function: This routine handles work elements on the MODIFY queue by building a set of work elements for the synchronous queue, causing TCAS to be modified.

Entry_from: IKTCAS40

Registers_at_Entry:

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS40

Registers_at_Exit:

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

00 Normal return

External_References: IKTCAS56

Control_Blocks_Used: TCAST, TWAR
Mapping_Macros_Used: IKTCASWA, IKTTCAST
Executable_Macros_Used: FREEMAIN, GETMAIN, POST, WTO
Module_Attributes: Serially reusable
Lock_Dependency: None
Messages: IKT015I
Abend_Codes: None
Data_Sets: None

IKTCAS51 -- Initializaticn Routine

Names:

Assembly Module: IKTCAS51
Object Module: IKTCAS51
Alternate Entry Points: Ncne
Load Module: IKTCAS51

MO_Diagram: 6.0

Function: This routine allocates storage for and initializes the TWAR, and establishes the addressability of all TCAS routines.

Entry_from: IKTCAS00

Registers_at_Entry:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS00

Registers_at_Exit:

Register 1: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External_References: IKTCAS52, IKTCAS53, IKTCAS54, IKTCAS55, IKTCAS56

Control_Blocks_Used: TWAR

Mapping_Macros_Used: IKTCASWA

Executable_Macros_Used: GETMAIN

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: IKT002I, IKT012D

Abend_Codes: None

Data_Sets: None

IKTCAS52 --- Termination Routine

Names:

Assembly Module: IKTCAS52
Object Module: IKTCAS52
Alternate Entry Points: Ncne
Load Module: IKTCAS51

MC_Diagram: 6.0

Function: This routine provides an SVC dump if the system operator requests one. It also detaches subtasks and frees terminal transfer requests.

Entry_from: IKTCAS00 (normal) or IKTCAS55 (error)

Registers_at_Entry:

Register 1: Request code:

01 Provide SVC dump
02 Detach subtasks
04 Free cross-memory requests

Register 10: Address of a work element
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: Caller

Registers_at_Exit:

Register 1: Request code:

01 Provide SVC dump
02 Detach subtasks
04 Free cross-memory requests

Register 10: Address of a work element
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External References: None

Control Blocks Used: CVT, TCAST, TWAR

Mapping Macros Used: CVT, IKTCASWA, IKTCAST

Executable Macros Used: DETACH, FREEMAIN, POST, SDUMP, WAIT, WTOR

Module Attributes: Serially reusable

Lock Dependency: None

Messages: None

Abend_Codes: None

Data_Sets: None

IKTCAS53 -- TCAST Creation Routine

Names:

Assembly Module: IKTCAS53
Object Module: IKTCAS53
Alternate Entry Points: Ncne
Load Module: IKTCAS51

MO_Diagram: 6.0

Function: This routine allocates storage for and initializes the TCAST.

Entry_from: IKTCAS00

Registers_at_Entry:

Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS00

Registers_at_Exit:

Register 1: Address of TCAST
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code (in hex):

00 Normal return
04 Invalid START command
1C No storage available in the CSA

External_References: Ncne

Control_Blocks_Used: CVT, TCAST, TWAR

Mapping_Macros_Used: CVT, IKTCASWA, IKTTCAST

Executable_Macros_Used: FREEMAIN, GETAMIN, LOAD, MODESET

Module_Attributes: Serially reusable

Lock_Dependency: None

Messages: None

Abend_Codes: None

Data_Sets: None

IKTCAS54 -- Parameter Processing Scanner

Names:

Assembly Module: IKTCAS54

Object Module: IKTCAS54
Alternate Entry Points: None
Load Module: IKTCAS51

MC Diagram: 6.0

Function: This routine validates and processes START and MODIFY command parameters.

Entry from: IKTCAS00

Registers at Entry:

Register 9: Address of input options list
Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit to: IKTCAS00

Registers at Exit:

Register 9: Address of input options list
Register 11: Address of TWAR
Register 13: Address of register save area
Register 14: Return address
Register 15: Return code:

00 Normal return
04 Input options list error

External References: IKTCAS56

Control Blocks Used: CIE, CSCB, DCB, JFCB, TCAST, TWAR

Mapping Macros Used: IEECHAIN, IEFJFCBN, IEZCIE, IHADCB, IHADCBDP, IKTCASWA, IKTTCAST

Executable Macros Used: CLOSE, FREEMAIN, GET, GETMAIN, OPEN, PUT, QEDIT, RDJFCB, RETURN, WIO

Module Attributes: Serially reusable

Lock Dependency: None

Messages: IKT013I, IKT014I, IKT017I, IKT018I

Abend Codes: None

Data Sets: None

IKTCAS55 -- ESTAE Exit Routine

Names:

Assembly Module: IKTCAS55
Object Module: IKTCAS55
Alternate Entry Points: None
Load Module: IKTCAS51

MC Diagram: 6.0

Function: This routine provides recovery facilities for TCAS. If passes control back to R/TM for cleanup or retry, it creates work elements to reattach subtasks, or it creates a work element to end TCAS.

Entry_from: R/TM

Registers_at_Entry:

Register 0: Code X'12' (if no SDWA supplied by R/TM)
Register 1: Address of SDWA (if SDWA supplied by R/TM)
Register 2: Address of parameter passed from ESTAE macro (if no SDWA supplied by R/TM)
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: R/TM

Registers_at_Exit:

Register 0: Code X'12' (if no SDWA supplied by R/TM)
Register 1: Address of SDWA (if SDWA supplied by R/TM)
Register 2: Address of parameter passed from ESTAE macro (if no SDWA supplied by R/TM)
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External_References: IKTCAS56

Control_Blocks_Used: TWAR

Mapping_Macros_Used: IKTCASWA

Executable_Macros_Used: FREEMAIN, GETMAIN, RETURN, SETRP, WTO

Module_Attributes: Reenterable

Lock_Dependency: None

Messages: IKTO20I

Abend_Codes: None

Data_Sets: None

IKTCAS56 -- Message Routine

Names:

Assembly Module: IKTCAS56
Object Module: IKTCAS56
Alternate Entry Points: None
Load Module: IKTCAS51

MC_Diagram: None

Function: This routine provides the message text and the list form of the WTO or WTOR macro for TCAS routines that issue messages.

Entry_from: Any TCAS routine that issues a message

Registers_at_Entry:

Register 1: Message number
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_tc: Caller

Registers_at_Exit:

Register 1: Address of storage that contains the message length
and the message, or zero if an invalid message number
Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

External_References: None

Control_Blocks_Used: None

Mapping_Macros_Used: None

Executable_Macros_Used: WTO, WTOR

Module_Attributes: Reenterable

Lock_Dependency: None

Messages: Contains all TCAS messages

Abend_Codes: None

Data_Sets: None

IKTINX1 -- Installation-Written Logon Exit Exit Routine

Names:

Object Module: IKTINX1
Load Module: IKTCAS20

Function: An installation must write this routine if it wants 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. IKTINX1 must verify that the terminal is supported by installation-written routines (that is, a terminal input manager, a terminal output manager, and edit routine IKTGETXT), and it must set the terminal type (to a value other than X'01' or X'02') and the buffer size.

Entry_from: IKTCAS23

Registers_at_Entry:

Register 1: Address of a parameter list containing:

- Address of the RPL
- Address of a byte fcr indicating terminal type
- Address of two bytes for indicating terminal buffer size

Register 13: Address of register save area
Register 14: Return address
Register 15: Entry point address

Exit_to: IKTCAS23

Registers at Exit:

Register 15: Return code:

00 Recognized terminal type; logon processing will continue
04 Unrecognized terminal type; logon processing will terminate

TCAS DIRECTORY

This section shows the relationship between TCAS load module and object module names. It also shows which HIPO diagram (in "TCAS Method of Operation") is associated with each object module, and whether the module has a program organization description (in "TCAS Program Organization").

There are three columns:

- Load module name, object module name - Load module names appear farthest to the left. Indented under each load module name are the names of the object modules contained in the load module. Assembly module names are the same as the object module names. There are no alternate entry points in any TCAS modules.
- MO Diagram - The identification number of the HIPO diagram (in "TCAS Method of Operation") in which the function corresponding to each object module is described.
- PO Description - An indication (X) as to whether there is a description in "TCAS Program Organization" associated with each object module.

Load Module Name Object Module Name	MO Diagram	PO Description
Main Task		
IKTCAS00		
IKTCAS00	6.0	X
IKTCAS51		
IKTCAS51	6.0	X
IKTCAS52	6.0	X
IKTCAS53	6.0	X
IKTCAS54	6.0	X
IKTCAS55	6.0	X
IKTCAS56	None	X
VTAM Interface Subtask		
IKTCAS20		
IKTCAS20	6.1	X
IKTCAS21	6.1	X
IKTCAS22	6.1	X
IKTCAS23	6.1	X
IKTCAS24	6.1	X
IKTCASCX	6.1	X
IKTCASOX	6.1	X
IKTINX1	6.1	X
User Interface Subtask		
IKTCAS30		
IKTCAS30	6.2	X
IKTCAS31	6.2	X
IKTCAS32	6.2	X
IKTCAS33	6.2	X
IKTCASX1	6.2	X
Console Communication Subtask		
IKTCAS40		
IKTCAS40	6.3	X
IKTCAS41	6.3	X
IKTCAS42	6.3	X

TCAS DATA AREAS

This section provides information about TCAS data areas. It contains:

- A TCAS control block overview (Figure 13)
- A description of a work element

Descriptions of the control blocks used by TCAS can be found in OS/VS2 System Programming Library: Debugging Handbook (Volume 2), GC28-0709, and in OS/VS2 System Data Areas, SYB8-0606 (microfiche).

A cross reference between data area names and TCAS object modules that use them is contained in the system data area usage table microfiche, OS/VS2 Data Area Usage Table, SYB8-0742. A list giving the names of data area fields and the TCAS object modules that update them is contained in the system symbol usage table microfiche, OS/VS2 Symbol Usage Table, SYB8-0744.

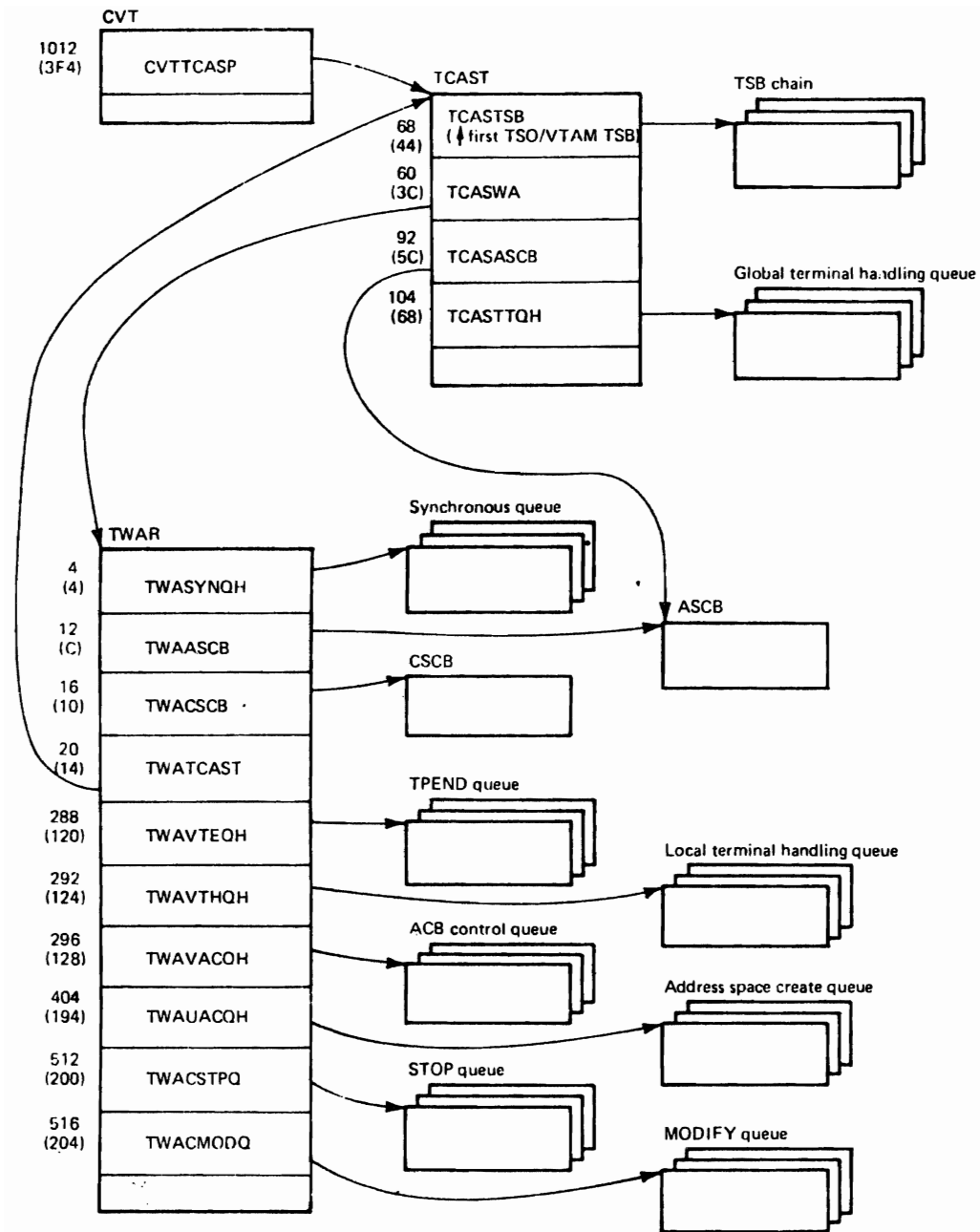


Figure 13. TCAS Control Block Overview

Work Element

A work element (see Figure 14) represents a unit of work to be performed by TCAS. Each work element has primary and secondary task and function codes. The primary task code indicates which TCAS task should perform the function indicated by the primary function code. The secondary task code specifies which task should be dispatched when the primary task is unable to complete its work. The field containing the address of the next work element provides a means of queuing work elements. The entry code/return code field is sometimes used for intertask communication. The basic work element is 12 bytes long; if a variable length work element is used, the length field will have a value greater than 12. Variable work elements are mapped by IKWESTD.

A list of task codes and function codes appears below. Function codes marked with an asterisk (*) are used only for error recording and identification (see "TCAS Diagnostic Aids"); their functions are not processed as work elements.

<u>Task Code</u> <u>(hexadecimal)</u>	<u>Function Code</u> <u>(hexadecimal)</u>	<u>Function</u>
01		TCAS main task (IKTCAS00)
	10	START command processing (IKTCAS00)
	*20	Call TCAS initialization routine IKTCAS51
	*30	Call TCAST creation routine IKTCAS53
	*40	Call parameter processing scanner IKTCAS54
	50	Attach VTAM interface subtask driver IKTCAS20
	60	Attach user interface subtask driver IKTCAS30
	70	Attach console communication subtask driver IKTCAS40
	80	Call TCAS termination routine IKTCAS52
	91	End of task (IKTCAS00)
	92	TCAS main task abend (IKTCAS00)
	*A0	Wait routine (IKTCAS00)
	FF	Return
02		VTAM interface subtask (IKTCAS20)
	*01	VTAM interface initialization (IKTCAS20)
	*02	Route control (IKTCAS20)
	03	End of task (IKTCAS20)
	11	ACB control - open ACB (IKTCAS21)
	12	ACB control - close ACB (IKTCAS21)
	13	ACB control - quiesce logons (IKTCAS21)
	14	ACB control - start logons (IKTCAS21)
	21	Terminal handling - transfer terminal (IKTCAS22)
	22	Terminal handling - release terminal (IKTCAS22)
	*23	Terminal handling - free terminal control blocks (IKTCAS22)
	*31	Logon processing - initial terminal processing (IKTCAS0X)
	*32	Logon processing - request address space (IKTCASCX)

<u>Task Code</u> <u>(hexadecimal)</u>	<u>Function Code</u> <u>(hexadecimal)</u>	<u>Function</u>
	41	TPEND (IKTCAS24)
	FF	Return
03		User interface subtask (IKTCAS30)
	*01	User interface initialization (IKTCAS30)
	*02	Route control (IKTCAS30)
	03	End of task (IKTCAS30)
	11	Address space creation (IKTCAS31)
	21	Address space termination - system initiated cancel (IKTCAS32)
	22	Address space termination - CALLR/TM (IKTCAS32)
	31	TSB manipulation - chain TSB (IKTCAS33)
	32	TSB manipulation - extract ASCB address (IKTCAS33)
	FF	Return
04		Console communication subtask (IKTCAS40)
	*01	Console communication initialization (IKTCAS40)
	*02	Route control (IKTCAS40)
	03	End of task (IKTCAS40)
	11	Process STOP command (IKTCAS41)
	21	Process MODIFY command (IKTCAS42)
	FF	Return

0(0) Primary task code	1(1) Primary function code	2(2) Secondary task code	3(3) Secondary function code
4(4) ↑ next work element			
8(8) Entry code or return code		10(A) Length of work element (12 bytes if not variable)	

Figure 14. A TCAS Work Element

TCAS DIAGNOSTIC AIDS

This section contains information that can be used to diagnose TCAS problems. It contains:

- A message list containing the names of object modules that detect the conditions requiring messages, issue the messages, and contain the message text
- An abend code list containing the name of an object module that detects the condition requiring an abend code and issues the code
- A description of TCAS error recording

Messages

The following messages are issued by TCAS object modules. For explanations of the messages see OS/VS Message Library: VS2 System Messages, GC38-1002. For explanations of the terminal messages, see CS/VS2 TSO Terminal Messages Directory, SY28-0654.

System Messages

<u>Message</u> <u>ID</u> _____	<u>Module</u> <u>Detecting</u>	<u>Module</u> <u>Issuing</u>	<u>Module</u> <u>Containing</u>
IKT001D	IKTCAS00	IKTCAS00	IKTCAS56
IKIC02I	IKTCAS00 IKTCAS51	IKTCAS00 IKTCAS51	IKTCAS56 IKTCAS56
IKT003D	IKTCAS00	IKTCAS00	IKTCAS56
IKT004D	IKTCAS00	IKTCAS00	IKTCAS56
IKIC05I	IKTCAS00	IKTCAS00	IKTCAS56
IKIC06I	IKTCAS00	IKTCAS00	IKTCAS56
IKT007I	IKTCAS21	IKTCAS21	IKTCAS56
IKT008I	IKTCAS21	IKTCAS21	IKTCAS56
IKT009I	IKTCAS24	IKTCAS24	IKTCAS56
IKIC10I	IKTCAS41	IKTCAS41	IKTCAS56
IKT011I	IKTCAS40	IKTCAS40	IKTCAS56
IKT012D	IKTCAS51	IKTCAS51	IKTCAS56
IKT013I	IKTCAS54	IKTCAS54	IKTCAS56
IKT014I	IKTCAS54	IKTCAS54	IKTCAS56
IKT015I	IKTCAS42	IKTCAS42	IKTCAS56
IKT016D	IKTCAS41	IKTCAS41	IKTCAS56
IKT017I	IKTCAS54	IKTCAS54	IKTCAS56
IKT018I	IKTCAS54	IKTCAS54	IKTCAS56
IKT019I	IKTCASCX IKTCASOX IKTCAS23	IKTCASCX IKTCASOX IKTCAS23	IKTCAS56
IKT020I	IKTCAS55	IKTCAS55	IKTCAS56

Terminal Messages

<u>Message</u> <u>ID</u> _____	<u>Module</u> <u>Detecting</u>	<u>Module</u> <u>Issuing</u>	<u>Module</u> <u>Containing</u>
IKIC0201I	IKTCAS31	IKTCAS31	IKTCAS56
IKT00202I	IKTCAS31	IKTCAS31	IKTCAS56
IKIC0203I	IKTCAS31	IKTCAS31	IKTCAS56

Abend Codes

The following abend code is issued by TCAS. For an explanation of the code, see OS/VS Message Library: VS2 System Codes, GC38-1008.

<u>Code</u>	<u>Module Detecting</u>	<u>Module Issuing</u>
CAC	IKTCAS00	IKTCAS00

How TCAS Records Errors

Each TCAS task has a corresponding segment in the TCAS work area (TWAR) for recording errors. The segments are initialized to zero. When each routine of a task begins execution, the function code, retry address, and register save area address of the routine are stored in the task's footprint area (TWAME, TWAVE, TWAUE, or TWACE). Each footprint area is 32 bytes long; each footprint is eight bytes long (function code-1 byte, retry address-3 bytes, register save area address-4 bytes). A footprint is stored in the first nonzero eight bytes of a footprint area. When a routine finishes processing, it resets its footprint to zero.

When a TCAS task abnormally terminates, the function code corresponding to the function being performed at time of abend is stored in the abend recording area (TWAMABFC, TWAVABFC, TWAUABFC, or TWACABFC) of the abending task. Because a TCAS task is not allowed to abend more than twice, only the first 2 bytes of any TCAS task's abend recording area are used.

When a TCAS routine fails, before R/TM gets control, the function code corresponding to the function being performed at time of failure is stored in the retry recording area (TWAMRTFC, TWAVRTFC, TWAURTFPC, or TWACRTFC) of the task executing the failing routine. The task terminates if retry is attempted twice.

Note: A list of TCAS function codes appears in "TCAS Data Areas."



APPENDIX A. ABBREVIATIONS

ACE Access method control block
ASCP Address space control block
ASCII American national standard code for information interchange
ASVT Address space vector table
ATTN Attention

CIB Command input buffer
CID Communications identifier
CMS Cross memory services
CSA Common service area
CSCB Command scheduling control block
CVT Communications vector table

EBCDIC Extended binary coded decimal interchange code
ECP Event control block
EXLST Exit list

FME Function management end
FRR Functional recovery routine
FRRS Functional recovery routine stack

HIFO Hierarchy plus input-process-output

IPARMS Input queue parameters
IRE Interruption request block

LB Logon buffer

MO Method of operation

NIE Node information block

OPARMS Output queue parameters

PG Program organization
PSA Prefixed save area

QVPL Queue verification parameter list

RB Request block
RMPL Resource manager parameter list
RPL Request parameter list
RU Request unit
R/TM Recovery/termination management

SCVT Secondary communications vector table
SEWA System diagnostic work area
SIC System-initiated cancel
SRB Service request block
SRM System resources manager
SVRB Supervisor request block

TCAM Telecommunications access method
TCAS Terminal control address space
TCAST TCAS table
TCP Task control block
TCI Timing control table
TCXD TCAM CVT extension

TIM	Terminal input manager
IIMWA	Terminal input manager work area
TIOC	Terminal I/O coordinator
TIOCRPT	TIOC reference pointer table
TCM	Terminal output manager
IOMWA	Terminal output manager work area
TSE	Terminal status block
ISEX	TSB extension
TVCS	TSO/VTAM CSA area
IVWA	TSO/VTAM work area
TWAR	TCAS work area
USS	Unformatted system services
VTAM	Virtual telecommunications access method
VTIOC	VTAM terminal I/O coordinator
WE	Work element
WESTC	Standard work element
WSAVI	Work/save area vector table
ISA	Extended save area

APPENDIX B. INSTALLATION-WRITTEN EXIT ROUTINES

VTIOC and TCAS code contains hooks to exit routines that can be written by an installation 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 record mode terminals not supported by TSO/VTAM. (Note that an installation must write its own terminal input manager and terminal output manager, in addition to exit routines IKTGETXT, IKTIMX2, and IKTIMX1, in order to support record mode terminals not supported by TSO/VTAM. Also note that the IBM-supplied I/O managers have a record mode interface to VTAM.)

The exit routines are optional. Before an exit routine can be used it must be link-edited with the object module that calls it. 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 below. More detailed descriptions, including the function of each routine and required linkages, are provided in "VTIOC Program Organization" and "TCAS Program Organization."

Name	Purpose	Terminal Type	Caller
VTICC			
IKTCASX1	Replace or supplement IBM-supplied output editing	Non-supported	IKTCAS31
IKTGETXT	Edit input data	Non-supported	IKTVTGET
IKTIDSX1	Replace or supplement IBM-supplied output editing	3270	IKT32700
IKTIDSX2	Supplement IBM-supplied input editing	3270	IKT3270I
IKTIDSX3	Replace IBM-supplied attention handling	3270 (LU0)	IKTIMIDS
IKTIDSX4	Replace or supplement IBM-supplied input editing	3270	IKTVTGET
IKTIDSX5	Replace IBM-supplied attention handling	3270 (LU2)	IKTIMLU2
IKTIMX2	Initialize installation-written I/O managers	Non-supported	IKTXINIT
IKTRTX1	Replace or supplement IBM-supplied output editing	3767/3770	IKTTOMRT
IKTRTX2	Supplement IBM-supplied input editing	3767/3770	IKTTIMRT
IKTRTX3	Replace IBM-supplied attention handling	3767/3770	IKTTIMRT
IKTRTX4	Replace or supplement IBM-supplied input editing	3767/3770	IKTVTGET

Name	Purpose	Terminal Type	Caller
TCAS			
IKTINX1	Set terminal type and buffer size	Non-supported	IKTCAS23

INDEX

- abbreviations 179
- ABEND codes
 - TCAS 177
 - VTIOC 133
- ACB (access method control block)
 - allocation and initialization 7
 - in VTIOC control block overview 124
- ACB control queue
 - creation 141
 - in TCAS control block overview 172
- address space create queue
 - in TCAS control block overview 172
 - processing 142
- ASCB (address space control block)
 - allocation and initialization 134
 - in TCAS control block overview 164
- ASCII data
 - conversion to EBCDIC 27,31
- attention handling 40
 - (see also IKTATTM)
- attention interruption
 - receiving 26,32
 - sending 28,33
- automatic character prompting
 - halting 59
 - restarting 60
 - sending 28,33
 - setting 58
- automatic line numbering
 - halting 59
 - restarting 60
 - sending 28,33
 - setting 56
- branching, terminal control macro 52
- breakin, setting 54
- BUFRSIZE value 8
- calling address space TPUT with ASID routine 23
 - (see also IKTASTPT)
- CALLRTM macro 142
- cancel command 30
- cancellation of user address space 142
- character prompting, automatic
 - halting 59
 - restarting 60
 - sending 28,33
 - setting 58
- character translation
 - on input 27,31
 - on output 28,33
 - with STRAN macro 61
- CLEAR key 26
- clearing queue 53
- closing of ACB 13,140
- CLSDST macro, issued 8,12,13,14,141
- codes, ABEND
 - TCAS 177
 - VTIOC 133
- command processor, in TCAS Console
 - Communication MO diagram 144
- console communication, TCAS 144
- control blocks, overview
 - TCAS 172
 - VTIOC 124
- control flow, module to module
 - TCAS 146
 - VTIOC 67
- converting data
 - ASCII to EBCDIC 27,31
 - EBCDIC to ASCII 28,33
 - user-specified 27,31
- creation of user address space 142
- cross-memory TPUT, sending 23
- CSCB (command scheduling control block)
 - in TCAS control block overview 172
- CVT (communications vector table)
 - in TCAS control block overview 172
 - in VTIOC control block overview 124
- data areas
 - TCAS 171
 - VTIOC 123
- directory
 - TCAS 169
 - VTIOC 119
- EBCDIC data
 - conversion to ASCII 28,33
- editing
 - on input 20,27,31
 - on output 22,23,28,33
- ESTAE routine for SVC 93 (IKT93EST) 25
- exit routines
 - Bind (IKTCASOX) 141
 - ESTAE (IKTCAS55) 138
 - installation-written 181
 - logon (IKTCAS23) 141
 - LOSTERM (IKTLTERM) 14
 - OPNDST RPL (IKTRPLXT) 10
 - release/pass (IKTCASCX) 141
 - TPEND (IKTCAS24) 140
- EXLST (exit list)
 - allocation and initialization 8
 - in VTIOC control block overview 124
- extended logoff routine 13
 - (see also IKTLOGFF)
- extended logon routine 11
 - (see also IKTXLOG)
- flashback line, sending 28
- FRR routines
 - IKTIMFRR 31
 - IKTIOFRR 43
 - IKTOMFRR 29,33
 - IKTQMFRR1 48,51
 - IKTQMFRR2 48,51
 - IKTXMFRR 24

global terminal handling queue
 creation 17
 freeing 16
 in TCAS control block overview 172
 processing 140

I/O manager initialization routine 9
 (see also IKTIOM)

IEAVEQV3
 in IKTQMIN MO diagram 48
 in IKTQMOUT MO diagram 51

IEDAY8
 in control flow 67
 in IKTAY81 MO diagram 16
 in IKTAY82 MO diagram 17
 in IKTAY83 MO diagram 18

IGC0001C
 in directory 119
 in IKTAY8 PO description 76

IGC0009C
 in control flow 68
 in directory 120
 in IKTASTPT MO diagram 23
 in IKTASTPT PO description 73
 in IKTGETXT PO description 77
 in IKTIDSX4 PO description 79
 in IKTRTX4 PO description 94
 in IKTVTGET MO diagram 20
 in IKTVTGET PO description 98
 in IKTVTPUT MO diagram 22
 in IKTVTPUT PO description 99
 in TGET/TPUT overview MO diagram 19

IGC0009D
 in control flow 72
 in directory 121
 in IKT0940A MO diagram 57
 in IKT0940A PO description 103
 in IKT0940C MO diagram 58
 in IKT0940C PO description 104
 in IKT0940D MO diagram 59
 in IKT0940D PO description 105
 in IKT0940E MO diagram 60
 in IKT0940E PO description 106
 in IKT0940F MO diagram 61
 in IKT0940F PO description 106
 in IKT09401 MO diagram 53
 in IKT09401 PO description 108
 in IKT09404 MO diagram 54
 in IKT09404 PO description 109
 in IKT09405 MO diagram 55
 in IKT09405 PO description 110
 in IKT09409 MO diagram 56
 in IKT09409 PO description 111
 in IKT09412 MO diagram 62
 in IKT09412 PO description 112
 in IKT09413 MO diagram 63
 in IKT09413 PO description 113
 in IKT09414 MO diagram 64
 in IKT09414 PO description 114

IGG09301
 in TGET/TPUT overview MO diagram 19

IGG09302
 in TGET/TPUT overview MO diagram 19

IKJEFLA
 in control flow 66
 in directory 120
 in IKTIOM PO description 81

 in IKTINX2 PO description 83
 in IKTLOGFF PO description 84
 in IKTRPLXT PO description 92
 in IKTXINIT MO diagram 8
 in IKTXINIT PO description 99

IKJEFLB
 in control flow 67
 in IKTLOGFF MO diagram 13
 in IKTLOGFF PO description 84

IKJEFLC
 in control flow 66
 in directory 119
 in IKTLOGR PO description 86
 in IKTXLOG MO diagram 11
 in IKTXLOG PO description 100

IKJEFLD
 in control flow 66
 in IKTLOGR MO diagram 12
 in IKTLOGR PO description 86

IKJEFLF
 in control flow 67
 in IKTLTERM MO diagram 14,15

IKTASCII
 in directory 120
 PO description 73

IKTASTPT
 in control flow 68
 in directory 120
 MO diagram 23
 PO description 73

IKTATTN
 in directory 120
 MO diagram 40
 PO description 74

IKTAY8
 (see also IKTAY81, IKTAY82, IKTAY83)
 in control flow 67
 in directory 119
 PO description 75

IKTAY81
 in control flow 67
 in directory 119
 in PO description 75
 MO diagram 16

IKTAY82
 in control flow 67
 in directory 119
 in PO description 75
 MO diagram 17

IKTAY83
 in control flow 67
 in directory 119
 in PO description 75
 MO diagram 18

IKTCASCX
 in control flow 147
 in directory 170
 in MO diagram 141
 PO description 148

IKTCASOX
 in control flow 147
 in directory 170
 in MO diagram 141
 PO description 148

IKTCASX1
 in control flow 146
 in directory 170
 in installation exit summary 182

in MO diagram 142
 PO description 149
 IKTCAS00
 in control flow 146
 in directory 170
 in MO diagram 138
 PO description 150
 IKTCAS20
 in control flow 146
 in directory 170
 in MO diagram 140
 PO description 151
 IKTCAS21
 in control flow 146
 in directory 170
 in MO diagram 140
 PO description 152
 IKTCAS22
 in control flow 146
 in directory 170
 in MO diagram 140
 PO description 152
 IKTCAS23
 in control flow 147
 in directory 170
 in MO diagram 141
 PO description 153
 IKTCAS24
 in control flow 146,147
 in directory 170
 in MO diagram 140
 PO description 154
 IKTCAS30
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 156
 IKTCAS31
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 156
 IKTCAS32
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 157
 IKTCAS33
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 158
 IKTCAS40
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 159
 IKTCAS41
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 160
 IKTCAS42
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 161
 IKTCAS51
 in control flow 146
 in directory 170
 in MO diagram 142
 PO description 162
 IKTCAS52
 in control flow 147
 in directory 170
 in MO diagram 143
 PO description 163
 IKTCAS53
 in control flow 146
 in directory 170
 in MO diagram 138
 PO description 164
 IKTCAS54
 in control flow 146
 in directory 170
 in MO diagram 138
 PO description 164
 IKTCAS55
 in control flow 147
 in directory 170
 in MO diagram 138
 PO description 165
 IKTCAS56
 in directory 170
 PO description 166
 IKTEXT
 in directory 120
 MO diagram 38
 PO description 76
 IKTGETXT
 in control flow 68
 in directory 120
 in IKTVTGET MO diagram 20
 in installation exit summary 182
 PO description 77
 IKTIDSX1
 in control flow 71
 in directory 120
 in IKTOMIDS MO diagram 28
 in installation exit summary 182
 PO description 78
 IKTIDSX2
 in control flow 69
 in directory 120
 in IKTIMIDS MO diagram 27
 in installation exit summary 182
 PO description 78
 IKTIDSX3
 in control flow 69
 in directory 120
 in IKTIMIDS MO diagram 26
 in installation exit summary 182
 PO description 79
 IKTIDSX4
 in control flow 68
 in directory 120
 in IKTVTGET MO diagram 20
 in installation exit summary 182
 PO description 79
 IKTIDSX5
 in control flow 69
 in directory 120
 in MO diagram 34
 PO description 80
 IKTIOM
 in control flow 66

in directory 119
 MO diagram 9
 PO description 81
 IKTIMFRR
 in directory 120
 in IKTTIMRT MO diagram 31
 in IKTTIMRT PO description 95
 IKTIMIDS
 in control flow 69
 in directory 120
 MO diagram 26
 PO description 81
 IKTIMLU2
 in control flow 69
 in directory 120
 MO diagram 34
 PO description 82
 IKTIMX1
 in directory 170
 in installation exit summary 182
 in TCAS VTAM Interface MO diagram 141
 PO description 167
 IKTIMX2
 in control flow 66
 in directory 119
 in IKTXINIT MO diagram 8
 in installation exit summary 182
 PO description 83
 IKTIOPRR
 in control flow 69,70
 in directory 120
 MO diagram 43
 PO description 84
 IKTIOM00
 in directory 120
 IKTIOM01
 in directory 120
 IKTIOM02
 in directory 120
 IKTIOM03
 in directory 120
 IKTIST00
 in directory 120
 MO diagram 45
 PO description 84
 IKTLOGFF
 in control flow 67
 in directory 119
 MO diagram 13
 PO description 84
 IKTLOGR
 in control flow 66
 in directory 119
 MO diagram 12
 PO description 86
 IKTLTERM
 in control flow 67
 in directory 119
 MO diagram 14
 PO description 87
 IKTMSGS
 in directory 121
 PO description 88
 IKTOMFRR
 in directory 120
 in IKTOMIDS MO diagram 28
 in IKTOMIDS PO description 88
 in IKTTOMRT MO diagram 32
 in IKTTOMRT PO description 96
 IKTOMIDS
 in control flow 71
 in directory 120
 MO diagram 28
 PO description 88
 IKTOMLU2
 in control flow 71
 in directory 120
 MO diagram 36
 PO description 89
 IKTQMEV
 in directory 120
 in IKTQMIN MO diagram 48
 in IKTQMNGR PO description 90
 in IKTQMOUT MO diagram 51
 IKTQMF1
 in directory 120
 in IKTQMIN MO diagram 48
 in IKTQMNGR PO description 90
 in IKTQMOUT MO diagram 51
 IKTQMF2
 in directory 120
 in IKTQMIN MO diagram 48
 in IKTQMNGR PO description 90
 in IKTQMOUT MO diagram 51
 IKTQMIN
 (see also IKTQMNGR)
 in control flow 68,69
 in directory 120
 MO diagram 46
 IKTQMNGR
 (see also IKTQMIN, IKTQMOUT)
 in control flow 68,69,71,72
 in directory 120
 PO description 90
 IKTQMOUT
 (see also IKTQMNGR)
 in control flow 68,69,71
 in directory 120
 MO diagram 49
 IKTRPLXT
 in control flow 66
 in directory 119
 MO diagram 10
 PO description 92
 IKTRTX1
 in control flow 71
 in directory 120
 in IKTTOMRT MO diagram 32
 in installation exit summary 182
 PO description 92
 IKTRTX2
 in control flow 69
 in directory 120
 in IKTTIMRT MO diagram 30
 in installation exit summary 182
 PO description 92
 IKTRTX3
 in control flow 69
 in directory 120
 in IKTTIMRT MO diagram 31
 in installation exit summary 182
 PO description 94
 IKTRTX4
 in control flow 68
 in directory 120
 in IKTVTGET MO diagram 20

in installation exit summary 182
 PO description 94
 IKTTIMRT
 in control flow 69
 in directory 120
 MO diagram 30
 PO description 95
 IKTTOMJR
 in control flow 71
 in directory 120
 MO diagram 29
 PO description 96
 IKTTOMRT
 in control flow 71
 in directory 120
 MO diagram 32
 PO description 96
 IKTVTGET
 in control flow 68
 in directory 120
 MO diagram 20
 PO description 97
 IKTVTPUT
 in control flow 68
 in directory 120
 MO diagram 22
 PO description 98
 IKTXINIT
 in control flow 66
 in directory 119
 MO diagram 8
 PO description 99
 IKTXLOG
 in control flow 66
 in directory 119
 MO diagram 11
 PO description 100
 IKTXMFRR
 in directory 120
 in IKTXMTPT MO diagram 24
 in IKTXMTPT PO description 101
 IKTXMTPT
 in control flow 68
 in directory 120
 MO diagram 24
 PO description 101
 IKT0009D
 in control flow 72
 in directory 121
 MO diagram 52
 PO description 102
 IKT0940A
 in control flow 72
 in directory 121
 MO diagram 57
 PO description 103
 IKT0940C
 in control flow 72
 in directory 121
 MO diagram 58
 PO description 104
 IKT0940D
 in control flow 72
 in directory 121
 MO diagram 59
 PO description 105
 IKT0940E
 in control flow 72
 in directory 121
 MO diagram 60
 PO description 106
 IKT0940F
 in control flow 72
 in directory 121
 MO diagram 61
 PO description 106
 IKT09401
 in control flow 72
 in directory 121
 MO diagram 53
 PO description 108
 IKT09404
 in control flow 72
 in directory 121
 MO diagram 54
 PO description 109
 IKT09405
 in control flow 72
 in directory 121
 MO diagram 55
 PO description 110
 IKT09409
 in control flow 72
 in directory 121
 MO diagram 56
 PO description 111
 IKT09412
 in control flow 72
 in directory 121
 MO diagram 62
 PO description 112
 IKT09413
 in control flow 72
 in directory 121
 MO diagram 63
 PO description 113
 IKT09414
 in control flow 72
 in directory 121
 MO diagram 64
 PO description 114
 IKT3270I
 in control flow 69,70
 in directory 120
 MO diagram 41
 PO description 115
 IKT32700
 in control flow 71
 in directory 120
 MO diagram 42
 PO description 116
 IKT93EST
 in control flow 69
 in directory 120
 MO diagram 25
 PO description 116
 initialization
 of ACB 8
 of ASCB 142
 of EXLST 8
 of logon buffer 141,142
 of NIB 8,142
 of pointers to ASCII translation
 tables 9
 of RPL 142
 of terminal input manager 9

- of terminal output manager 9
- of TSB 142
- of TSBX 142
- of TVWA 8
- of TWAR 138
- of VTAM control blocks 8
- routines
 - IKTIOM 9
 - IKTLOGR 12
 - IKTRPLXT 10
 - IKTXINIT 8
 - IKTXLOG 11
- initialization/termination routines
 - introduction 1
 - module names 119
- input queue
 - allocation 8
 - in VTIOC control block overview 124
- input queue element 128
 - input queue manager 46
 - (see also IKTQMIN)
- input queue manager parameter list 125
- installation-written exit routines 181
- inter-terminal communication, setting 157
- IPARMS (input queue manager parameter list) 125
- keyboard, unlocking 26
- legend for HIPO figures 5
- line numbering, automatic
 - halting 59
 - restarting 60
 - sending 28,33
 - setting 56
- line size, setting 57
- local terminal handling queue
 - creation 143
 - in TCAS control block overview 172
 - processing 140
- logical unit, types of 3
- Logical Unit Status command 31
- logoff processing 14,17,18
 - extended logoff 13
- logon buffer
 - address 11
 - allocation and initialization 141,142
 - in VTIOC control block overview 124
- LOGON initialization routine (see IKJEFLA)
- LOGON/LOGOFF verification routine (see IKJEFLA)
- logon processing 141
 - extended logon 11
 - logon reconnect 12
- logon reconnect routine 12
 - (see also IKTLOGR)
- LOGON scheduling routine (see IKJEFLB)
- look-aside buffer
 - allocation 8
 - in VTIOC control block overview 124
- lost terminal exit scheduler 45
- LOSTERM exit routine 14
 - (see also IKTLTERM)
- LUS (Logical Unit Status) command 31
- LU0 3
- LU1 3
- LU2 3
- main task, TCAS 138
- messages
 - TCAS 176
 - VTIOC 132
- MODIFY command 144
- MODIFY queue
 - creation 144
 - in TCAS control block overview 172
 - processing 144
- NIB (node information block)
 - allocation and initialization 8
 - in VTIOC control block overview 124
 - null RU routine 29
- OPARMS (output queue manager parameter list) 126
- opening, of ACB 8,140
- OPNDST macro
 - checked 10
 - issued 8,14,141
- OPNDST RPL asynchronous exit routine 10
 - (see also IKTRPLXT)
- output queue
 - allocation 8
 - in VTIOC control block overview 124
- output queue element 129
- output queue manager 46
 - (see also IKTQMOUT)
- output queue manager parameter list 126
- parse/scan interface routine (see IKJEFLA)
- QTIP 29 equivalent routine (IKTAY83)
 - in control flow 68
 - in directory 119
 - in PO description 76
 - MO diagram 18
- queue
 - adding elements 46,49
 - clearing 53
 - removing elements 46,49
 - updating elements 46,49
- queue element
 - input format 128
 - output format 129
- queue manager
 - introduction 1
 - module names 120
- queue manager look-aside buffer
 - allocation 8
 - in VTIOC control block overview 124
- queue verification 48,51
- R/TM
 - in TCAS control flow 146,147
 - in VTIOC control flow 68
- RECEIVE macro
 - in introduction 1
 - issued 26,30
- reconnection of TSO/VTAM terminal 12,14
- RPL (request parameter list)
 - allocation and initialization 8

in VTIOC control block overview 124
 RTAUTOPT routine 60
 (see also IKT0940E)
 RTM1
 in TCAS User Interface MO diagram 143
 RTM2
 in VTIOC control flow 67

 SEND macro
 in introduction 1
 issued 28,29,32
 SPAUTOPT routine 59
 (see also IKT0940D)
 START command 138
 started task control
 in IKTCAS00 PO description 149
 in TCAS control flow 146
 in TCAS Main Task MO diagram 138
 STAUTOCP routine 58
 (see also IKT0940C)
 STAUTOLN routine 56
 (see also IKT09409)
 STBREAK routine 54
 (see also IKT09404)
 STC (started task control)
 in IKTCAS00 PO description 149
 in TCAS control flow 146
 in TCAS Main Task MO diagram 136
 STCOM routine 55
 (see also IKT09405)
 STPSMODE routine 62
 (see also IKT09412)
 STLINENO routine 63
 (see also IKT09413)
 STOP command 144
 STOP queue
 creation 144
 in TCAS control block overview 172
 processing 144
 STSIZE routine 57
 (see also IKT0940A)
 STTHPMD routine 64
 (see also IKT09414)
 STTRAN routine 61
 (see also IKT0940F)
 SVC 93 (see TGET/TPUT routines)
 SVC 94 (see terminal control macro
 routines)
 synchronous queue
 creation 139
 in TCAS control block overview 172
 processing 140,142,144
 system-initiated cancel 14,142

 target address space TPUT with ASID
 routine 24
 (see also IKTXHTPT)
 TCAS console communication 144
 TCAS termination routine 16
 (see also IKTAY81)
 TCAS main task 138
 TCAS user interface 142
 TCAS VTAM interface 140
 TCAST (TCAS table)
 allocation and initialization 138
 freeing 16,17

 in TCAS control block overview 172
 in VTIOC control block overview 124
 TCB
 in VTIOC control block overview 124
 TCLEARQ routine 53
 (see also IKT09401)
 terminal control, transfer of 8,140
 terminal control macro routines
 introduction 1
 module names 120
 terminal I/O managers
 introduction 1
 module names 120
 terminal input manager
 (see also IKTIMIDS, IKTIMLU2, IKTTIMRT)
 introduction 1
 terminal output manager
 (see also IKTOMIDS, IKTOMLU2, IKTTOMRT)
 introduction 1
 terminal output manager null RU routine 29
 (see also IKTTOMJR)
 terminal user address space termination
 routine 17
 (see also IKTAY82)
 terminals, types supported 2
 termination
 of TCAS 16,144
 of TSO/TCAM user address space 18
 of TSO/VTAM user address space 17,142
 TGET routine 20
 (see also IKTVTGET)
 TGET/TPUT routines
 introduction 1
 module names 120
 TIM (terminal input manager)
 (see also IKTIMIDS, IKTTIMRT)
 introduction 1
 SRB allocation and initialization 9
 TIM work area
 allocation 9
 in VTIOC control block overview 124
 TIM/TOM list
 freeing 16,17
 TOM (terminal output manager)
 (see also IKTOMIDS, IKTTOMRT)
 introduction 1
 SRB allocation and initialization 9
 TOM work area
 allocation 9
 in VTIOC control block overview 124
 TPEND queue
 creation
 in TCAS control block overview 172
 processing 140
 TPUT with ASID routine 23,24
 (see also IKTASTPT, IKTXHTPT)
 TPUT without ASID routine 22
 (see also IKTVTPUT)
 transfer of terminal control 8,140
 translation tables
 with output data 28,32
 with STTRAN macro 61
 TSB (terminal status block)
 allocation and initialization 142
 freeing 17
 in TCAS control block overview 172
 in VTIOC control block overview 124
 updating 12

TSBX (TSB extension)
 allocation and initialization 142
 freeing 17
 in VTIOC control block overview 124
 updating 12
 TSOKEY00 member of parmlib
 BUFRSIZE value 8
 TVCS (TSO/VTAM CSA area)
 allocation and initialization 23
 freeing 23
 in VTIOC control block overview 124
 TVWA (TSO/VTAM work area)
 allocation of storage for 8
 in VTIOC control block overview 124
 initialization 8,9
 TWAR (TCAS work area)
 allocation and initialization 138
 in TCAS control block overview 172

 unlocking keyboard 26
 user interface, TCAS 142

 visual table of contents
 TCAS 137
 VTIOC 6
 VTAM
 in TCAS control flow 147
 in VTIOC control flow 67

 VTAM interface, TCAS 140
 VTIOC initialization routine 8
 (see also IKTXINIT)

 work element
 description 173
 format 174

 3270 LU0 terminal input manager 26
 (see also IKTIMIDS)
 introduction 1
 3270 LU2 terminal input manager 34
 (see also IKTIMLU2)
 3270 LU0 terminal output manager 28
 (see also IKTOMIDS)
 introduction 1
 3270 LU2 terminal output manager 35
 (see also IKTOMLU2)
 3767/3770 terminal input manager 30
 (see also IKTTIMRT)
 introduction 1
 3767/3770 terminal output manager 32
 (see also IKTTOMRT)
 introduction 1



SY27-7269-1

OS/V/S2 MVS VTIOC and TCAS Logic (File No. S370-30) Printed in U.S.A. SY27-7269-1



VTAM Terminal I/O
Coordinator (VTIOC)
Terminal Control Address Space (TCAS)

SY27-7269-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

How did you use this publication?

- As an introduction
- As a reference manual
- For another purpose (explain) _____
- As a text (student)
- As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

Cut or Fold Along Line

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address:

IBM branch office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

SY27-7269-1

Reader's Comment Form

Cut or Fold Along Line

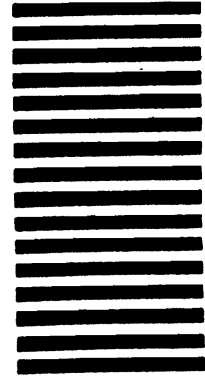
Fold

Fold



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 63T
Neighborhood Road
Kingston, New York 12401

Fold

Fold

OS/VS2 MVS VTIOC and TCAS Logic (File No. S370-30) Printed in U.S.A. SY27-7269-1



SY27-7269-1

