

“Restricted Materials of IBM ”
All Rights Reserved
Licensed Materials - Property of IBM
©Copyright IBM Corp. 1987, 1989
LY28-1670-1
File No. S370-36

Program Product

**MVS/Extended Architecture
System Logic Library:
Dumping Services**

MVS/System Product:

JES3 Version 2 5665-291
JES2 Version 2 5740-XC6



Second Edition (September, 1989)

This is a major revision of, and obsoletes LY28-1670-0. See the Summary of Amendments following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Version 2 Release 2 of MVS/System Product program number 5665-291 and 5740-XC6 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product in this publication is not intended to state or imply that only IBM's product may be used. Any functionally equivalent product may be used instead. This statement does not expressly or implicitly waive any intellectual property right IBM may hold in any product mentioned herein.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 950, Poughkeepsie, N.Y. 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

The MVS/Extended Architecture System Logic Library is intended for people who debug or modify the MVS control program. It describes the logic of most MVS control program functions that are performed after master scheduler initialization completes. For detailed information about the MVS control program prior to this point, refer to MVS/Extended Architecture System Initialization Logic. For general information about the MVS control program and the relationships among the components that make up the MVS control program, refer to the MVS/Extended Architecture Overview. To obtain the names of publications that describe some of the components not in the System Logic Library, refer to the section Corequisite Reading in the Master Preface in MVS/Extended Architecture System Logic Library: Master Index.

TRADEMARKS

The following are trademarks of International Business Machines Corporation.

- MVS/DFP(TM)
- MVS/XA(TM)

HOW THE LIBRARY IS ORGANIZED

SET OF BOOKS

The System Logic Library consists of a set of books. Two of the books provide information that is relevant to the entire set of books:

1. The MVS/Extended Architecture System Logic Library: Master Index contains the master preface master index for the other books in the set.
2. The MVS/Extended Architecture System Logic Library: Module Descriptions contains module descriptions for all of the modules in the components documented in the System Logic Library.

Each of the other books (referred to as component books) in the set contains its own table of contents and index, and describes the logic of one of the components in the MVS control program.

ORGANIZATION OF THE COMPONENTS

Most component books contain information about one component in the MVS control program. However, some component books (such as System Logic Library: Initiator/Terminator) contain more than one component if the components are closely related, frequently referenced at the same time, and not so large that they require a book of their own.

A three or four character mnemonic is associated with each component book and is used in all diagram and page numbers in that book. For example, the mnemonic ASM is associated with the book MVS/Extended Architecture System Logic Library: Auxiliary Storage Management. All diagrams in this book are identified as Diagram ASM-n, and all pages as ASM-n, where n represents the specific diagram or page number. Whenever possible, the existing component acronym is used as the mnemonic for the component book. The Table of Book Titles in the Master Preface in MVS/Extended Architecture System Logic Library: Master Index lists the book titles, the components included in each book (if a book contains more than one component), the mnemonics for the books, and the order number for each book.

HOW TO USE THE LIBRARY

To help you use this library efficiently, the following topics cover

- How to find information using book titles and the master index
- What types of information are provided for each component
- How to obtain further information about other books in the System Logic Library

FINDING INFORMATION USING THE BOOK TITLES

As you become familiar with the book titles, MVS component names and mnemonics, and the book contents, you will be able to use the System Logic Library as you would an encyclopedia and go directly to the book that you need. We recommend that you group the books in alphabetical order for easy reference, or, if you are familiar with MVS, that you group the books by related functions.

The Table of Book Titles in the Master Preface in MVS/Extended Architecture System Logic Library: Master Index contains a list of book titles and mnemonics. It provides a quick reference to all the books, and their corresponding components, in the System Logic Library.

FINDING INFORMATION USING THE MASTER INDEX

If you are not sure which book contains the information you are looking for, you can locate the book and the page on which the information appears by using the master index in System Logic Library: Master Index. For the component books, the page number in an index entry consists of the mnemonic for the component and the page number.

For example:

ASM-12 refers to MVS/Extended Architecture System Logic Library: Auxiliary Storage Management, page ASM-12.

INFORMATION PROVIDED FOR MOST COMPONENTS

The following information is provided for most of the components described in the System Logic Library.

1. An introduction that summarizes the component's function
2. Control block overview figures that show significant fields and the chaining structure of the component's control blocks
3. Process flow figures that show control flow between the component's object modules
4. Module information that describes the functional organization of a program. This information can be in the form of:
 - Method-of-Operation diagrams and extended descriptions.
 - Automatically-generated prose. The automated module information is generated from the module prologue and the code itself. It consists of three parts: module description, module operation summary, and diagnostic aids.
5. Module descriptions that describe the operation of the modules (the module descriptions are contained in System Logic Library: Module Descriptions)

Some component books also include diagnostic techniques information following the Introduction.

FURTHER INFORMATION

For more information about the System Logic Library, including the order numbers of the books in the System Logic Library, see the Master Preface in MVS/Extended Architecture System Logic Library: Master Index.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

CONTENTS

Dumping Services DMP-1

Introduction DMP-3

Formatted and Unformatted - Abend Dumps DMP-3

Unformatted Dump - SVC Dump DMP-4

CHNGDUMP Operator Command DMP-4

Formatting Specific Control Blocks DMP-5

Data Management Control Block Formatter DMP-5

IOS Control Block Formatter DMP-5

Diagnostic Techniques DMP-7

SVC Dump Debugging Aids DMP-7

Important SVC Dump Entry Points DMP-7

SVC dump Error Conditions DMP-8

SYS1.LOGREC Entries Produced for SVC Dump Errors DMP-9

Control Blocks Used to Debug SVC Dump Errors DMP-14

SVC Dump Data Area (SDDAT) DMP-16

Resource Cleanup for SVC Dump DMP-16

Process Flow DMP-19

Method of Operation DMP-49

IEAVAD00 - SVC 51 Overview DMP-54

IEAVAD00 - SVC Dump Processing DMP-56

IEAVAD01 - Snap Dump Processing DMP-68

IEAVTABD - ABDUMP Processing DMP-82

IEAVTDSV - DUMPSRV Address Space Jobstep Task DMP-92

IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE)

Processor DMP-94

IEAVTSDR - SVC Dump/DUMPSRV Resource Manager DMP-110

IEAVTSDT - Resident Dump Task Processing (IEAVTSDT) DMP-116

IEAVTSDX - Schedule Dump Processing DMP-122

IEAVTSDX - Schedule Dump SRB Processing DMP-130

IEAVTSEP - SVC Post Exit Processor DMP-134

IEAVTSRB - SVC Dump SRB Routine DMP-136

IEAVTSSD - Summary Dump Processing DMP-152

IEAVTSSV - Suspend Summary Dump Processing DMP-160

IECDAFMT - Data Management Formatter Mainline DMP-164

IECIOFMT - IOS Formatter Mainline (IECIOFMT) DMP-168

IEECB867 - SVC Dump Reason Code Text Processor DMP-174

Index I-1

FIGURES

1. Abnormal End-of-Task DMP-20
2. ABEND/SNAP Dump Processing DMP-23
3. SVC Dump Overview DMP-25
4. Formatted Dump Module Flow DMP-28
5. Summary SVC Dump (SUMDUMP) Formatter for the AMDPRDMP Service Aid DMP-32
6. Synchronous Unformatted Dump Module Flow DMP-33
7. Scheduled Unformatted Dump Module Flow. DMP-37
8. Master Trace Dump Exit Processing DMP-45
9. RTM Services Module Flow DMP-46
10. Data Management Control Block Formatter DMP-47
11. IOS Control Block Formatter DMP-48
12. Key to Hipo Logic Diagrams DMP-49
13. Key to the Logic Diagrams DMP-51

SUMMARY OF AMENDMENTS

Summary of Amendments
for LY28-1670-1
for MVS/System Product Version 2 Release 2.3

This major revision contains changes to support MVS/System Product Version 2 Release 2.

The following changes have been made:

- MVS/Extended Architecture Data Facility Product (MVS/XA DFP) Version 3 Release 1.0, which introduces the storage management subsystem (SMS). SMS provides new function for data and storage management.
- Modifications to the Diagnostic Techniques section, including new and changed information on:
 - SVC dump debugging aids
- Changed information in the Process Flow section.
- The following changed modules in the Method of Operation section:
IEAVAD01
IEAVTABD
- The Preface has been updated to include the new title for the MVS/XA System Logic Library: Master Index and the deletion of the index from MVS/XA System Logic Library: Module Descriptions.
- Minor technical and editorial changes throughout the publication.

Summary of Amendments
for LY28-1670-0
MVS/System Product Version 2 Release 2.0

This publication is new for MVS/System Product Version 2 Release 2.0. It contains information that was reorganized from the Dumping Services section in MVS/XA System Logic Library Volume 6, LY28-1227-2, which applies to MVS/XA System Product Version 2 Release 1.7.

This publication contains changes to support MVS/System Product Version 2 Release 2.0. The changes include:

- The addition of information on the SVC dump data area, SDDAT.
- The following new modules:
IEAVTSRB
IEECB867
- The following changed modules:
IEAVTSDI
IEAVTSDR
- Minor technical and editorial changes throughout the publication.

INTRODUCTION

SVC 51 produces the following types of dumps:

- Formatted system dumps (SYSABEND/SYSUDUMP).
- Unformatted system dumps (SVC Dump).
- Unformatted abend dumps (SYSMDUMP).

FORMATTED AND UNFORMATTED - ABEND DUMPS

SNAP/ABDUMP is a subcomponent of the recovery termination management (RTM) component.

The SNAP routine is invoked by a SNAP macro instruction, which a user program can issue at any time to obtain a dynamic dump. The SNAP macro instruction, whose expansion contains an SVC 51, causes the SVC second level interrupt handler (SLIH) to call the SVC Dump routine. The SVC Dump routine checks the SNAP parameter list to determine whether a SNAP macro instruction has been issued. If so, the SVC Dump routine passes control to the SNAP routine.

The SNAP macro instruction can also be issued by an abend dump during abnormal termination. In this case, the ABDUMP routine is invoked, causing either a SYSABEND or SYSUDUMP formatted dump to be produced. (The ABDUMP routine can also cause a SYSMDUMP unformatted dump to be produced.) The ABDUMP routine causes the type of dump specified on the corresponding DD card to be produced. Thus, SNAP processing can provide either a formatted abnormal dump or a formatted dynamic dump. The dump produced by ABDUMP is governed by a combination of either:

- The default dump options as specified in the SYS1.PARMLIB member for the specified dump type
- The options specified on the CHNGDUMP operator command
- The options specified by the SNAP macro instruction in the ABEND, CALLRTM, or SETRP macro.

Figure 2 on page DMP-23 shows how ABDUMP determines the content of the dump to be taken based on the requested dump options.

The SYS1.PARMLIB default dump options for a SYSABEND dump consist of the major control blocks belonging to the task, enqueue control blocks, LSQA (local system queue area), programs and dynamically obtained storage, the GTF and trace table entries, and summary dump. For a SYSUDUMP dump, the default consists of only the summary dump. See SPL: Initialization and Tuning for further information. The default SYS1.PARMLIB dump options for SYSMDUMP are NUC, SQA, LSQA, RGN, SUM, SWA, and TRT.

These default options reside in the SYS1.PARMLIB members IEAABD00 for SYSABEND, in IEADMR00 for SYSMDUMP, and in IEADMP00 for SYSUDUMP. ABDUMP processing merges these options with those specified on the request.

If a dynamic dump is requested (the SNAP macro is issued by a user program), the storage areas to be dumped are specified by the operands of the SNAP macro. (See SPL: Supervisor Services and Macro Instructions for information on how to obtain a dump.)

The use of the SNAP routine is restricted to tasks that do not have job-step tasks within their subtask structure at entry to SNAP processing. If a task has a subtask that is a job-step task, control is returned immediately to the caller. Storage areas (storage, save areas, and the like) are dumped by SNAP in the current addressing mode of the issuer of the SNAP macro instruction. Storage areas dumped by ABDUMP are in the PSW addressing mode at the time of the error. Addresses are therefore 24 bits or 31 bits, depending on the current addressing mode.

UNFORMATTED DUMP - SVC DUMP

The SVC Dump service provides a quick, unformatted dump of virtual storage directly to a data set. To use SVC Dump, callers must have APF (authorized program facility) authorization, or be in control program key. The SDUMP macro instruction calls SVC Dump processing either by SVC 51 or branch entry.

The SVC 51 routine dumps the contents of virtual storage from the address space in which the request occurred—operating under the caller's task—or it initiates the dump of another address space by posting the permanent dump task in the destination address space being dumped by scheduling an SRB and operating under the dump task.

CHNGDUMP OPERATOR COMMAND

The CHNGDUMP operator command allows the operator to change the system dump options. (See System Commands, for a complete description of the CHNGDUMP, DUMPDS, DISPLAY DUMP, and DUMP operator commands and their uses.)

FORMATTING SPECIFIC CONTROL BLOCKS

This section describes how specific control blocks are formatted.

DATA MANAGEMENT CONTROL BLOCK FORMATTER

This formatter is called as a TCB exit from AMDPRDMP when the DM verb is used, or by SNAP/ABDUMP, when the DM option is specified. It formats the data management control blocks associated with each TCB. The control blocks that are formatted are:

- DEB — The data extent block is formatted when the DEB chain is available and SNAP/ABDUMP is the caller.
- DCB — The data control block is formatted when the associated DEB formatted successfully, the DCB is available, and SNAP/ABDUMP is the caller. AMDPRDMP calls this formatter if the associated task was terminated.
- IOB — The input/output block is formatted when the DCB formatter returns a zero return code and the IOBs are available.
- ICB — The interrupt control block is formatted when the DCB formatter returns a zero return code, the access method is SAM or BPAM with chained scheduling, and the ICBs are available.
- LCB — The line control block is formatted for the TCAM line group access method when the DCB contains a zero return code and the LCBs are available.

IOS CONTROL BLOCK FORMATTER

This formatter is called as a TCB exit from AMDPRDMP when the IO verb is used, or by SNAP/ABDUMP when the IO option is specified. It formats the IOS control blocks associated with each TCB. The control blocks that are formatted are:

- EXCPD (XDBA) — The EXCP debugging area is formatted whenever available and when SNAP/ABDUMP is the caller. This area is also formatted for an abended task when AMDPRDMP is the caller.
- UCB — The unit control block is formatted for each DEB in the DEB chain whenever available and when SNAP/ABDUMP is the caller. The UCB is also formatted for a terminated task when AMDPRDMP is the caller.

DIAGNOSTIC TECHNIQUES

SVC DUMP DEBUGGING AIDS

The SVC dump function is invoked when the SDUMP macro is issued. SVC dump produces dumps of system errors on a SYS1.DUMPxx or user-defined data set. SVC dump also produces abend dumps requested by SYSMDUMP DD statements.

Items that are important for you to understand when debugging errors in SVC dump processing are described in the following topics:

- Important SVC dump entry points
- SVC dump error conditions
- SYS1.LOGREC entries produced for SVC dump errors
- Control blocks used to debug SVC dump errors
- Resource cleanup for SVC dump

IMPORTANT SVC DUMP ENTRY POINTS

The BRANCH= parameter on the SDUMP macro determines the SVC dump entry points and mainline processing to be used.

BRANCH=YES Option

Entry point IEAVTSDX is used for branch-entry SVC dumps. IEAVTSDX creates a summary dump in a real storage buffer (if the SMDUMP option is requested on the SDUMP macro), schedules an SRB to invoke task (IEAVTSDT) processing to start the dump, and then returns control to the caller.

The branch entry option is requested by many FRR routines and some ESTAE routines. This option is also requested when ACTION=SVCD is specified on the SLIP command.

This option can only be used by routines running logically disabled (PSW disablement, locked, SRB Mode, or Super Bits Set).

BRANCH=NO Option

Entry point IEAVAD00 is used for the SVC entry to SVC dumps. The dump becomes a scheduled dump if any of the following schedule dump options occur in the parameter list:

- The ASID or ASIDLST option is specified.
- LISTA is requested.
- TYPE=XMEM or TYPE=XMEME is specified.
- SUBPLST is specified and contains an ASID other than the current ASID.

These options force a scheduled dump because they request data in an address space other than the current address space. For scheduled dump requests, IEAVAD00 calls IEAVTSDX (through routine IEAVTSGI), which schedules (or causes to be scheduled) one or more SRBs to invoke the dump task (IEAVTSDT) processing for the requested address spaces. After scheduling the SRBs, IEAVTSDX returns to IEAVAD00, which then returns to the caller. A synchronous dump request is an SVC-entered SDUMP which does not contain any of the scheduled dump options. For synchronous dump requests, IEAVAD00 processes the dump and then returns to the caller.

The SVC entry is requested by many ESTAE routines. It is also requested by the DUMP command (as a scheduled dump), and by the abend dump processor (IEAVTABD) for SYSMDUMP DD statements (as a synchronous dump).

Determining the SVC Entry Point

The title line produced for SVC dumps that are formatted by the print dump service aid indicates whether a scheduled or synchronous SVC dump was requested as follows:

- If "MODULE IEAVTSDT" is in the title line, the dump was scheduled (SDUMP BRANCH=YES or SDUMP BRANCH=NO with one or more of the schedule dump options).
- If the title line contains "MODULE SVCDUMP", the dump was synchronous (SDUMP BRANCH=NO with none of the schedule dump options).
- If the title line contains "MODULE modname", the dump was synchronous SDUMP or SYSMDUMP and the module name "modname" was found in a CDE or LPDE entry.
- If another module name is in the title line, then the dump was synchronous. SVC dump was able to determine which load module invoked it and insert the name in the title.

SVC DUMP ERROR CONDITIONS

If the SVC dump function encounters an unexpected error during its processing, it produces a software SYS1.LOGREC record and, if possible, continues taking the dump.

Expected program checks can occur when SVC dump is checking whether a virtual page that is to be dumped is valid and assigned. These program checks do not result in SYS1.LOGREC entries.

SVC dump issues abend 133 if it detects an unauthorized caller or abend 233 for an invalid input parameter. In these cases, LOGREC entries are not created and retry is not attempted.

SYS1.LOGREC ENTRIES PRODUCED FOR SVC DUMP ERRORS

The best starting place for debugging SVC dump problems is the SYS1.LOGREC entries contained in the in-storage SYS1.LOGREC buffer or in the SYS1.LOGREC data set, because a dump of the SVC dump problem is generally not available. (SVC dump does not take a dump of its own problems.)

Many SVC dump problems can be debugged from the SYS1.LOGREC entries alone. However, more complex problems may require a stand-alone dump that can be taken when a SLIP trap with ACTION=WAIT has been encountered. These problems include loops and failures to free critical system resources.

Following is a description of the fixed data entries and the variable data entries that SVC dump places in the system diagnostic work area (SDWA) for recording on SYS1.LOGREC:

Fixed Data Entries for Recording on SYS1.LOGREC

- | | | |
|-----------|---|---|
| SDWAMODN | - | Load module name (generally IGC0005A, which is the SVC 51 load module in SYS1.LPALIB). |
| SDWAC SCT | - | CSECT (microfiche) name in error, which can be any SVC dump module name. |
| SDWAREXN | - | Name of the CSECT (microfiche) that contains the recovery routines that handles the error. |
| SDWARRL | - | Recovery routine name, which is given as a label. This label can be found in the CSECT given by SDWAREXN. |
| SDWASC | - | Phrase that describes the general area of SDUMP that experienced the problem. |

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

The following table shows the label of the recovery routine, the microfiche name of the containing CSECT, and a description of the recovery processing.

Label	CSECT	Description
DTESTAE1	IEAVTSDT	<p>ESTAE routine for scheduled SVC dumps that are executing under the dump task (IEAVTSDT) in the requested address space. IEAVTSDT also establishes SDESTAEX which can percolate to DTESTAE1. DTESTAE1 records the problem and does the following to clean up:</p> <ol style="list-style-type: none"> 1. Frees the RSB. 2. Page-releases the virtual buffer. 3. Sets the system dispatchable. 4. Sets the address space swappable. 5. Enables for asynchronous interrupts. 6. Sets the tasks in the address space dispatchable. 7. When dump processing completes, posts the SDUMP caller. 8. When dump processing completes, unlocks SDUMP.
ESTAEXIT	IEEMB879	<p>ESTAE routine for the master trace exit routine invoked by SVC dump module IEAVTSDU. This ESTAE is established by IEEMB879.</p>
SCHFRR	IEAVTSDX	<p>FRR routine for branch entry to SDUMP. SCHFRR covers module IEAVTSDX from entry to exit but does not cover any of the SRBs. SCHFRR records all errors except X'133' and X'233' abends, and does the following to clean up:</p> <ol style="list-style-type: none"> 1. Frees the RSB. 2. Frees the system trace buffer if it was obtained. 3. Unlocks SDUMP.
SDESTAEX	IEAVAD00	<p>ESTAE routine for mainline SVC dump processing. This ESTAE is established by IEAVAD00 and IEAVTSDT. While percolating, SDESTAEX records errors in SDUMP processing and does the following to clean up:</p> <ol style="list-style-type: none"> 1. Frees the RSB. 2. Page-releases the virtual buffer in DUMPSRV. 3. If the dump is finished, calls IEAVTSDC. If not, calls IEAVTSDE entry IN IEAVTSDC. 4. If the dump is finished, posts callers ECB and unlocks SDUMP.
SDFRRRTN	IEAVAD00	<p>FRR routine for mainline SVC dump processing. This FRR is established by SVC dump modules when a lock is held and a retry is needed in the locked state. While percolating, SDFRRRTN records errors in SDUMP processing and does the following to clean up:</p> <ol style="list-style-type: none"> 1. Frees the RSB. 2. If the dump is finished, unlocks the SDUMP.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

Label	CSECT	Description
SRBFRR	IEEMB880	FRR routine for IEEMB880 which is the SRB routine that does data moves for IEEMB879. IEEMB879 is the master trace exit for SVC dump. This FRR is established by IEEMB880.
SUMFRR	IEAVTSSD	FRR routine for summary dump processing invoked for branch-entered SVC dumps. This FRR is established by IEAVTSSD. SUMFRR does recording and retries if allowed. No cleanup is performed by SUMFRR.
SRBFRR1	IEAVTSRB	FRR routine for the SRBs scheduled to post the dump tasks. This routine, or SRBFRR2, records the error; SRBFRR1 does the following to clean up: 1. Sets the system dispatchable. 2. Sets the address space swappable. 3. Enables for asynchronous interrupts. 4. Frees the RSB. 5. Resets the SDUMP caller. 6. Sets tasks in the address space dispatchable. 7. Frees the copy of the system trace tables. 8. If the SDUMP is completed, posts the caller and unlocks SDUMP.

Variable Data Entries for Recording on SYS1.LOGREC

The variable data that SVC dump places in the SDWAVRA field of the SDWA for recording on SYS1.LOGREC is:

- The 24-byte recovery routine parameter area if DTESTAE1, SDESTAEX, SDFRRRTN, or SUMFRR is the recovery routine name in SDWARRL. This area contains bits that indicate the resources held, other status bits, the retry address, the register update block (RUB) address, and the address of the SVC dump data area (ERRSDDAT at X'0'). The contents of the parameter area are mapped by the IHASDERR macro. The common name of the work area is ERRWORK.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

The ERRWORK data is mapped as follows:

Offset	Length	Name	Description
0(0)	24	ERRWORK	Mapping for ESTAE/FRR work area, used by modules called by IEAVAD00 and IEAVTSDT
0(0)	4	ERRSDDAT	Address of SDDAT for reg 7
4(4)	4	ERRRTDAT	Retry data
4(4)	4	ERRRETRY	Retry address used by the error recovery routine. If zero, then percolation is requested.
8(8)	4	ERRRUBAD	Address of the RUB used to reload registers for retry.
12(C)	4	*	Reserved
16(10)	1	ERRFLGS1	First flag byte:
1...	ERRSDFIX	IEAVTSDG is page fixed
.1..	ERRPGSDL	IEAVTSDL is page fixed
..1.	ERRSSEFX	IEAVTSSE is page fixed
...1	ERRVSMFX	IEAVTVSM is page fixed
.... 1..	ERRVSMTB	VSMLIST is page fixed
.... .1..	ERRSDBIT	1 = SVC DUMP is serialized (RTCTSDPL)
.... ..1.	ERRBFBIT	1 = 4K buffer is serialized (CVTSDBF)
.... ...1	*	Reserved
17(11)	1	ERRFLGS2	Second flag byte:
1...	ERRLLOCK	Local lock is held
.1..	ERRSLOCK	SALLOC lock is held
..1.	ERRCMSLK	CMS lock is held
...1	ERRDLOCK	Dispatcher lock is held
.... 1..	ERRCLOCK	CPU lock is held
.... .1..	ERRVPAGE	VSM page lock is held
.... ..1.	ERRVFIX	VSM fix lock is held
.... ...1	ERRSMSER	RSM serialization held
18(12)	1	ERRFLGS3	Third flag byte:
1...	ERRFRR	FRR protection is active
.1..	ERRESTAE	ESTAE protection is active
..1.	ERRSYSMD	1 = SYSMDUMP recovery 0 = SDUMP recovery
.... ..1.	ERRINIO	Error during I/O, no retry
.... 1..	ERRENDMP	Terminating error indicator
.... .1..	ERRSDENQ	ENQ performed on SDUMP resource
.... ..1.	ERRTRMNT	All dump tasks should DEQ and quit because of terminating error
.... ...1	ERRENSW	Switch to end-of-loop processing
19(13)	1	ERRFLGS4	Fourth flag byte:
1...	ERRSRBSD	SRB being scheduled
.1...	ERRCSDT	IEAVTSDT called this module
..1.	ERRCUDMP	Indicate this is current active dump
...1	ERRNORCD	Indicate recording not needed
.... 1..	ERRCDOUT	At least 1 RCD in dump data set
.... .1..	ERRTCBND	TCBs in address space not dispatched
.... ..1.	ERRGTFGO	Hook macro issued for GTF
.... ...1	ERRSSEXM	Module IEAVTSSE changed XM modes
20(14)	1	ERRFLGS5	Fifth flag byte:
1...	ERRSDCEN	SDC has been entered and should not attempt cleanup again
.111	1111	*	Reserved
21(15)	1	ERRSAVE	General save space
22(16)	2	*	Reserved

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

- For ESTAEXIT (IEEMB879) and SRBFRR (IEEMB880), the variable data consists of as much of the SVC dump exit parameter list (SDEXPARM) as will fit in the SDWAVRA.

The SDEXPARM data is mapped (by the IHASDEXP macro) as follows:

Offset	Length	Name	Description
0(0)	4	SDEXECB	ECB used to synchronize SVC dump processing.
4(4)	4	SDEXASCB	ASCB address used to post the ECB.
8(8)	2	SDEXWAID	ASID used to post the ECB.
10(A)	2	-	Reserved.
12(C)	4	SDEXBFAD	Buffer address for the data moved by the exit.
16(10)	4	SDEXBFLN	Length of the data buffer
20(14)	4	SDEXORAD	Address of the output routine used to process the buffer.
24(18)		SDEXKEYS	Two one-byte keys — for first and second 2K of storage in the buffer.
26(1A)	2	SDEXASID	ASID of the data that was last moved to the buffer.
28(1C)	4	SDEXNDAD	Address of the next data to move to the buffer.
32(20)	5	SDEXCDAD	From address of the data that was last moved to the buffer.
36(24)	4	SDEXBTTA	Start address of the master trace table (if the master trace exit is executing).
40(28)	4	SDEXTTLN	Length of the master trace table (if the master trace exit is executing).
44(2C)	72	SDEXSAVE	Save area for exit use.

The following routines enter the data indicated into the variable recording area in key/data format:

Label	CSECT	Description
DTESTAE1	IEAVTSDT	<ul style="list-style-type: none"> • 6-word ESTAE parameter area (see ERRWORK). • The data from RTCT: RTCCSDI, RTCTSDF, and RTCTSDF3. See RTCT (mapping macro) for details. • DSTAT/DAE dump status flags mapped by ADYDSTAT.
SCHFRR	IEAVTSDX	<ul style="list-style-type: none"> • 6-word FRR parameter area. • SDRSN—first word of SDRSN from RTSDRSN. See SDRSN (mapping macro IHASDRSN) for details. • DSTAT/DAE dump status flags mapped by ADYDSTAT. • Current TCB address (PSATOLD). • IHASDUMP/parameter list address from RTCTSDPL parameter list. • IHARTSD/RTSD address from RTCTRTSD. • RTSDXMST/contents of RTSDXMST (caller's control registers). • RTSDXPSW contents of RTSDXPSW/caller's PSW. • ASCB address of address space whose CML is held, from field RTSDCMLA. • RTSDCMSV/ save area for SVC dump CMSETS.

Label	CSECT	Description
SDESTAEX	IEAVAD00	<ul style="list-style-type: none"> • 6-word ESTAE work area (see ERRWORK). • RTCTSD3/RTCT ASID array (16 one-word entries). • SDRSN-contents of SDWSDRSN mapped by IHASDRSN. • DSTAT/DAE dump status flags mapped by ADYDSTAT. • IHASDUMP/parameter list address (SDWSDPL) parameter list.
SDFRRRTN	IEAVAD00	<ul style="list-style-type: none"> • 6-word ESTAE work area (see ERRWORK). • RTCTSD3/RTCT ASID array (16 one-word entries). • SDRSN-contents of SDWSDRSN mapped by IHASDRSN. • DSTAT/DAE dumps status flags mapped by ADYDSTAT. • IHASDUMP/parameter list address (SDWSDPL)/parameter list.
SUMFRR	IEAVTSSD	<ul style="list-style-type: none"> • 6-word FRR area mapped by IHASUMFR. • SDRSN-1st word of SDRSN from RTSDRSN. • Current TCB address (PSATOLD). • Home ASID (ASCBASID). • SMWKRFBAD/88 characters starting at SMWKRFBAD. See IHASMWK for details of the data. • SMWKNFRQ/76 characters starting at SMWKNFRQ. See IHASMWK for details of the data.
SRBFRR2 SRBFRR1	IEAVTSRB	<ul style="list-style-type: none"> • 6-word parameter area. • RTCTSD3/RTCT ASID array (16 one-word entries). • SDRSN/ contents of SDWSDRSN mapped by IHASDRSN. • DSTAT/DAE dumps status flags mapped by ADYDSTAT. • IHASDUMP/parameter list address parameter list.

CONTROL BLOCKS USED TO DEBUG SVC DUMP ERRORS

The following control blocks contain key information that can be used to debug problems in SVC dump routines.

- Address space control block (ASCB)
- DUMPSRV control block (DSVCB)
- Recovery termination control table (RTCT)
- RTCT extension for SDUMP (RTSD)
- SVC dump DIE control block (SDDIE)
- SVC dump work area (SDWORK)
- Summary dump work area (SMWK)
- SVC dump data areas (SDDAT)

Address Space Control Block (ASCB)

The ASCB contains the address of the TCB for the SVC dump task (IEAVTSDT) in the ASCBDUMP field (at offset X'60'). In this TCB, the TCBEVSVC bit (low-order bit at X'CC') is set on while the SVC dump task is executing. The ASCB is mapped by the IHAASCB macro.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

DUMPSRV Control Block (DSVCB)

The DSVCB is pointed to by RTCTDSV (X'2C') and is mapped by IHADSVCB. The DSVCB controls functions in the DUMPSRV address space. It is partitioned into sections which control separate functions. There is a section for address space control, dump data set control, and post dump exit processing.

Recovery Termination Control Table (RTCT)

The RTCT is pointed to by the CVTRTMCT field (at X'23C') in the CVT. It contains SVC dump information including status bits, and an array that contains information for the address spaces to be dumped. The RTCT is mapped by the IHARTCT macro.

The RTCT has a pointer (RTCTSDDS X'24') to the dump data set queue (SDDSQ). Each element in the queue is mapped by the SDDSQ, which is in mapping macro IHASDDSQ. See this mapping for any problems with selecting dump data sets or in using the DUMPDS command which modifies this queue.

RTCT Extension for SDUMP (RTSD)

The RTSD is pointed to by RTCTRTSD (X'16C') and is mapped by IHARTSD. The RTSD contains working storage and flags for SDUMP and an area which is used to save a copy of most of the caller's parameters. The caller's parameter list, storage lists, dump header, and other parameters are saved to allow the caller to release storage (using IEAVGM00) which may have been obtained to call SDUMP.

SVC Dump DIE Control Block (SDDIE)

The SDDIE contains all the storage used by the SVC dump timer DIE routine, IEAVTSDB. The SDDIE protects critical SVC dump resources. There are two SDDIE control blocks in the system. Both SDDIE control blocks are mapped by IHASDDIE. The SDDIE that protects the SVC dump real storage buffer is pointed to by RTCTDIRS (X'60') in the RTCT. The SDDIE that protects the system/task non-dispatchability is pointed to by RTCTDIND (X'50') in the RTCT. This SDDIE also protects the stopped unit of work during suspend summary dump processing.

SVC Dump Work Area (SDWORK)

The SDWORK is pointed to by the RTCTSDWK field (at X'DC') in the RTCT. It contains most of the reentrant storage used by SVC dump, including pointers to the I/O buffers and pointers that contains the 4104-byte SVC dump records before they are written to the dump data set. The SDWORK work area is mapped by the IHASDWRK macro.

Summary Dump Work Area (SMWK)

The SMWK is pointed to by the RTCTSDSW field (at X'B4') in the RTCT and contains fields used when a summary SVC dump was requested or defaulted (via the SUMDUMP option on the SDUMP macro). It includes counter fields that show how many real frames are used for the real storage buffer that holds the summary dump created for branch-entry callers of SVC dump. The count of real frames held (field SMWKFRHD at X'9A') is zeroed after the summary dump is written to the dump data set and the frames returned to RSM. The SMWK work area is mapped by the IHASMWK macro.

SVC DUMP DATA AREA (SDDAT)

The SDDAT is pointed to by the RTCTSDAT field at (X'64') in the RTCT. It contains pointers to all SVC dump primary data areas. This includes pointers to storage for dump SRBs, parameter lists for external services and working storage for SVC dump modules. The SDDAT data area is mapped by the IHASDDAT macro.

RESOURCE CLEANUP FOR SVC DUMP

At the end of an SVC dump, module IEAVTSDC (SDUMP cleanup routine) is given control and performs the following operations when necessary:

1. Sets the system dispatchable.
2. Sets the address space swappable.
3. Sets the tasks in the address space dispatchable.
4. Page-frees any modules or control blocks which were fixed by SDUMP.
5. Frees the copy of the saved system trace table.
6. Writes an end of file on the dump data set.
7. Issues message IEA911E for complete or partial dumps.
8. Issues or deletes message IEA994E or IEA994A for all dump data sets that are full.
9. Initiates post exit processing.

SVC dump also has a termination resource manager (IEAVTSDR) pointed to by CVTSDRM (X'21C'). IEAVTSDR receives control for all task and memory terminations. IEAVTSDR examines a resource manager parameter list (RMPL) to determine if any processing is required. IEAVTSDR handles the following situations:

- Memory terminations for address spaces involved in an SVC dump.
- Task terminations for tasks involved in an SVC dump.
- Memory termination of the DUMPSRV address space.

IEAVTSDR performs the following functions when necessary:

1. Frees the real storage buffer.
2. Sets the system dispatchable.
3. Sets the tasks in the address space dispatchable.
4. Frees the copied system trace table.
5. Resets the stopped caller of SDUMP.
6. Page-releases the virtual storage buffer.
7. Posts the SVC dump caller.
8. Unlocks SVC dump by zeroing the high-order bit of CVTSDBF (in the CVT) and zeroing the RTCTSDPL (in the RTCT).
9. Cleans up the DUMPSRV address space:
 - Frees storage on the post exit processing queue.
 - Removes entries from the SDDSQ that were added with DUMPDS.
 - Restarts the DUMPSRV address space.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

SVC dump also has a timer DIE processor (IEAVTSDB), which is pointed to by RTCTSOBB (X'64') in the RTCT. IEAVTSDB protects the following critical SVC dump resources:

- The real storage buffer
- System non-dispatchability
- Task non-dispatchability
- The stopped unit of work during suspend summary dump processing

IEAVTSDB performs the following operations when it detects that SVC dump processing has halted processing or has failed:

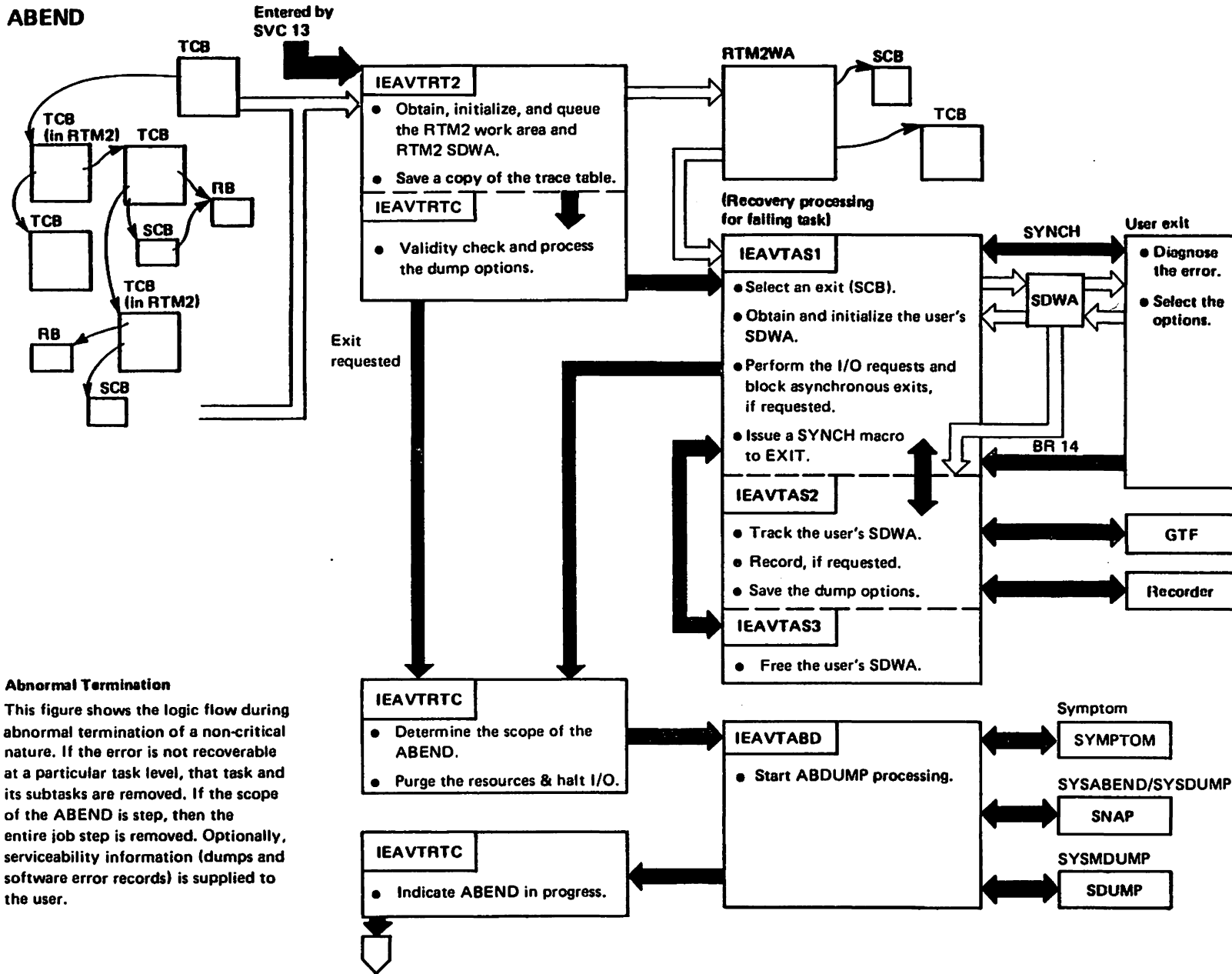
- Returns held real storage buffer frames to the system.
- Resets the system dispatchability.
- Resets the tasks of the address spaces (involved in the SVC Dump) dispatchability.
- Resets the stopped SVC dump caller's unit of work during suspend summary dump processing.

PROCESS FLOW

The figures in this section provide a functional overview for the following DMP services:

- Abnormal end-of-task
- ABEND/SNAP dump processing
- SVC dump overview
- Formatted dump module flow
- Summary SVC dump (SUMDUMP) formatter for the AMDPRDMP service aid
- Synchronous unformatted dump module flow
- Scheduled unformatted dump module flow
- Master trace dump exit processing
- RTM services module flow
- Data management control block formatter
- IOS control block formatter

ABEND



Abnormal Termination

This figure shows the logic flow during abnormal termination of a non-critical nature. If the error is not recoverable at a particular task level, that task and its subtasks are removed. If the scope of the ABEND is step, then the entire job step is removed. Optionally, serviceability information (dumps and software error records) is supplied to the user.

Figure 1 (Part 1 of 2). Abnormal End-of-Task

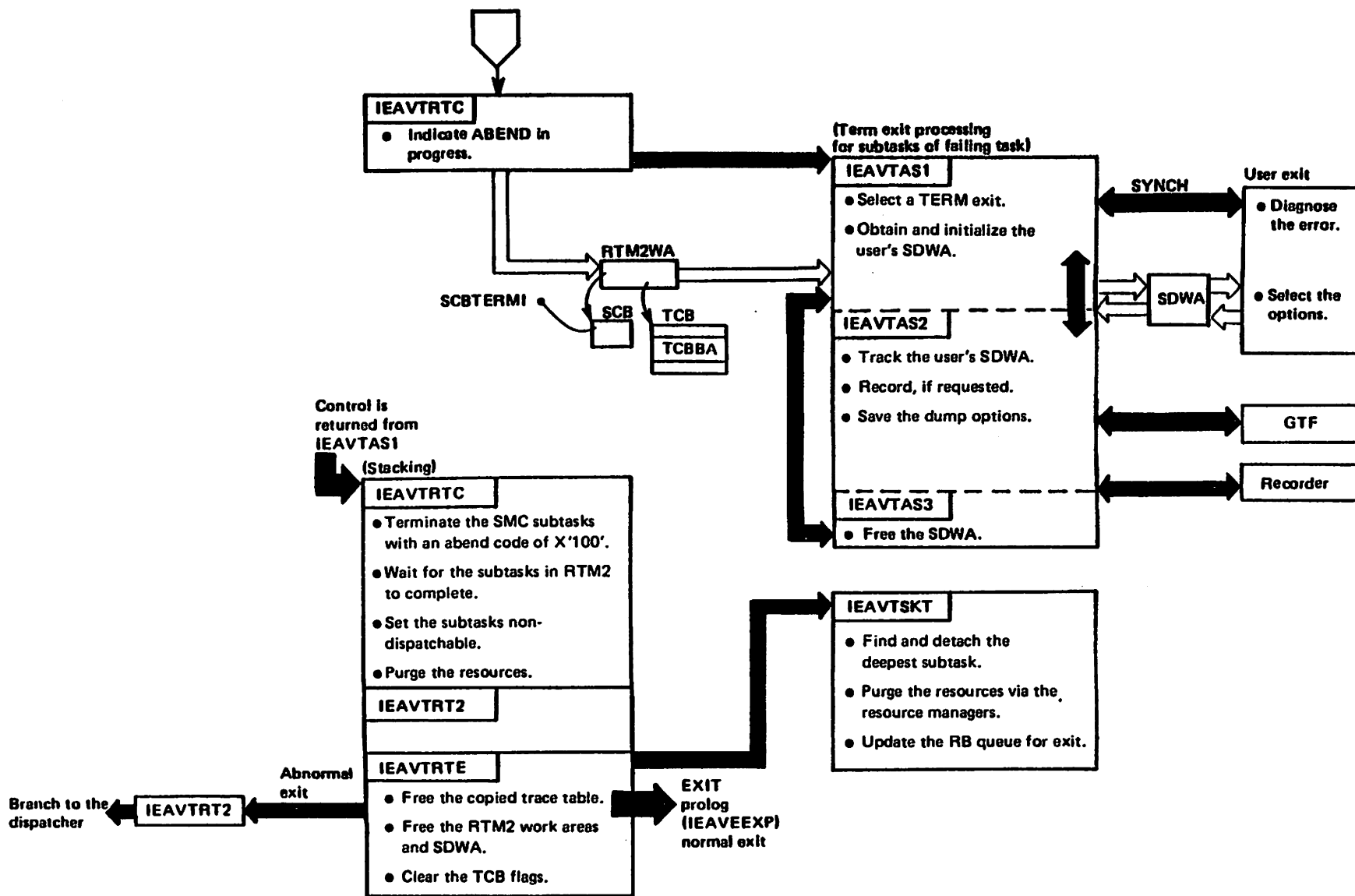
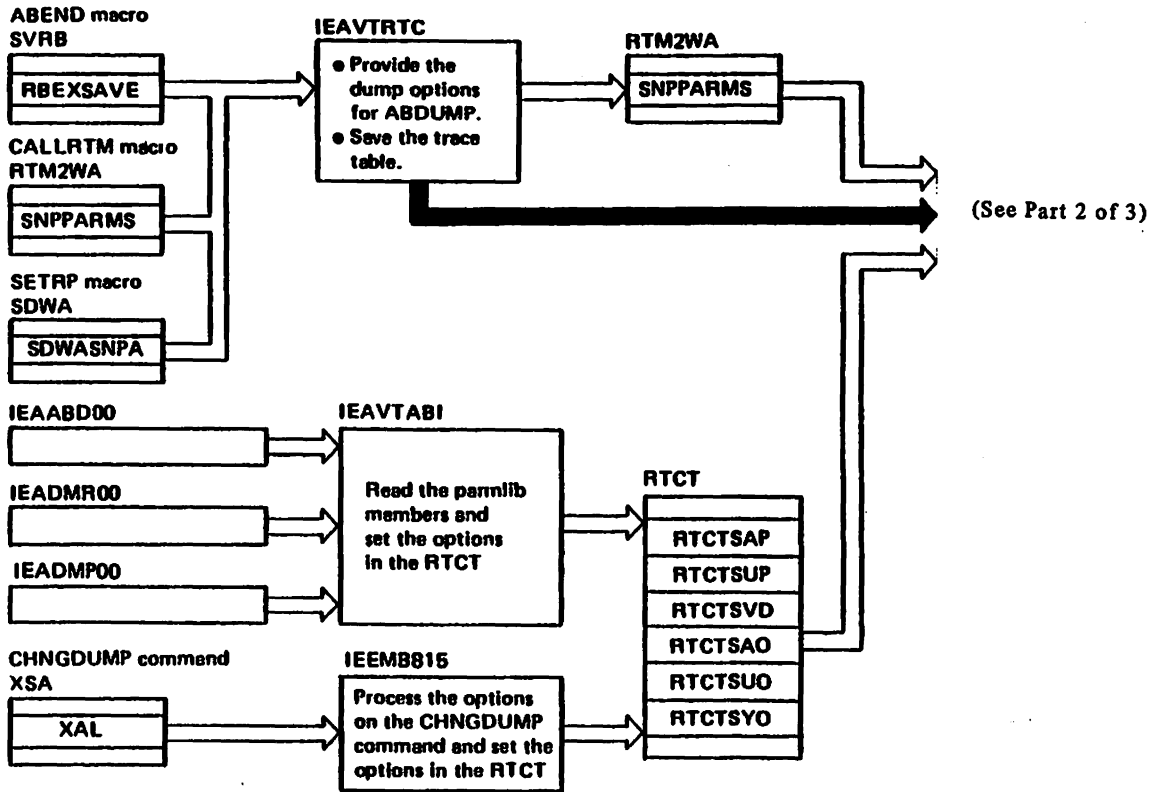


Figure 1 (Part 2 of 2). Abnormal End-of-Task



Note: SDDAT the SDUMP data area contains pointers to the following fields:

- ID
- SDPL
- SDWRK
- SDMOD
- SDCON
- RNGTB
- RNGTE
- OBPFIX
- OBUFF
- EXBUF
- CPYHD
- SDTAB
- SDEXP
- SMWK
- ERRWK
- TRSN
- TRCT
- LVLWK
- DSPD
- MDSN
- MDECB
- MSYMP

This figure provides an overview of all data areas related to ABEND/SNAP dumps, the sources from which the dump options are obtained, the key modules involved and the complete scheme of data flow. It ties together the function of system initialization requestor's dump options and operator intervention.

Figure 2 (Part 1 of 3). ABEND/SNAP Dump Processing

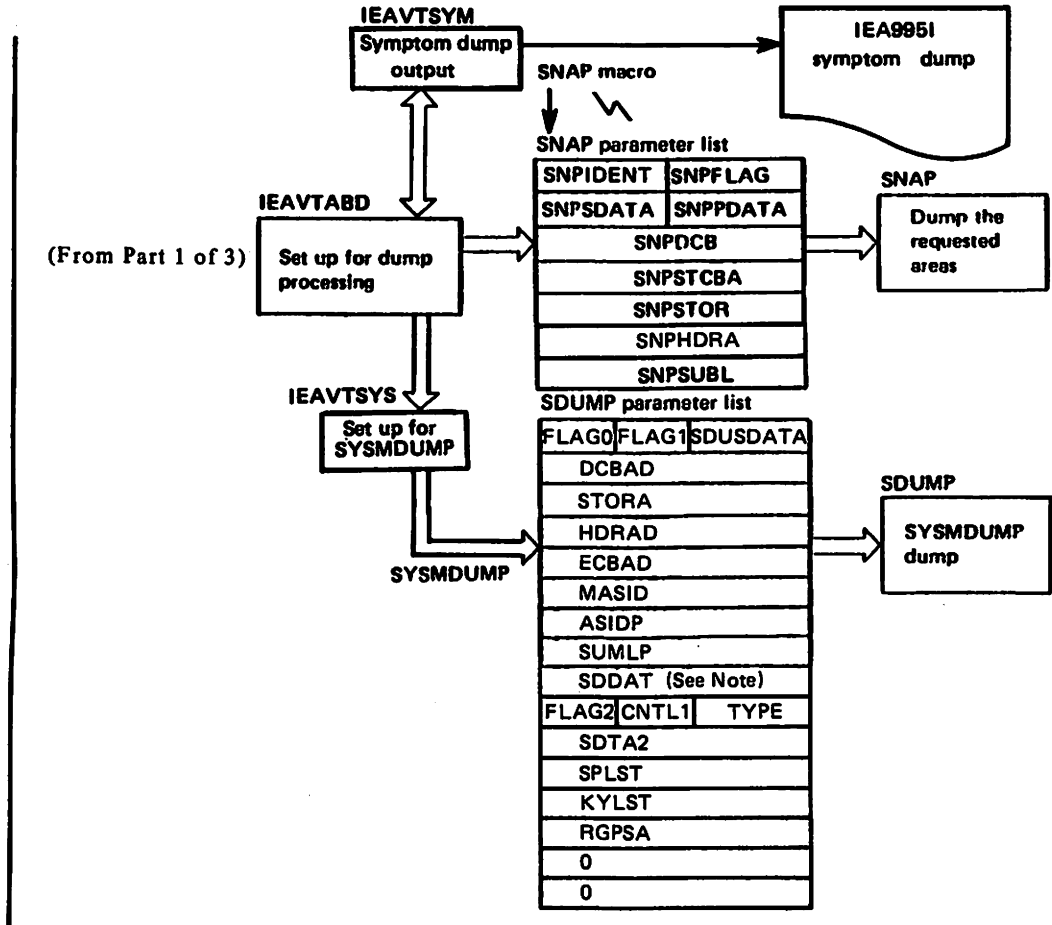


Figure 2 (Part 2 of 3). ABEND/SNAP Dump Processing

Dump options for ABEND dumps are determined from:

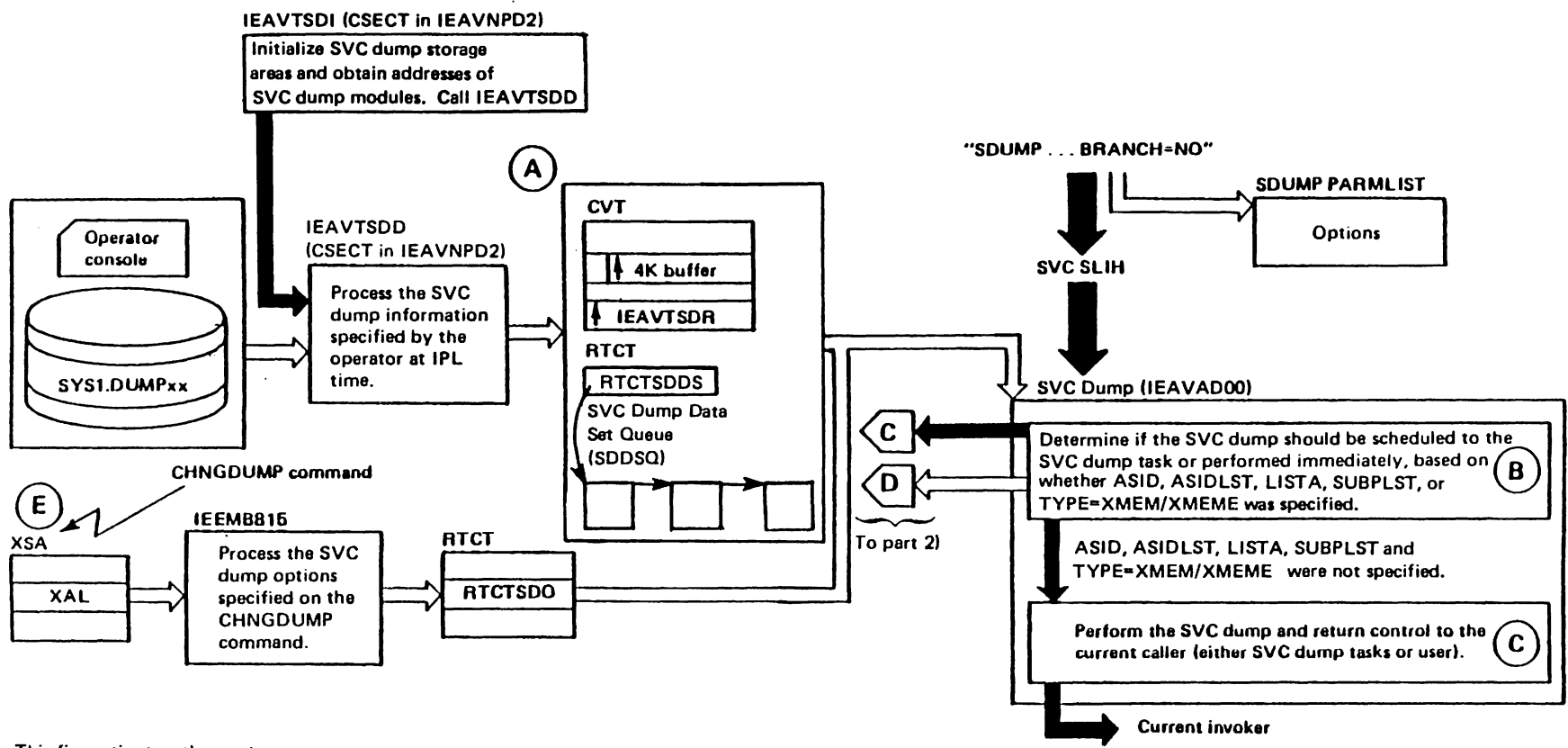
- 1) The options specified on the ABEND, CALLRTM or SETRP macro;
- 2) The options specified in SYS1.PARMLIB members IEAABD00 (SYSABEND options), IEADMP00 (SYSUDUMP options), and IEADMR00 (SYSMDUMP options).
- 3) Options specified on the CHNGDUMP command.

When dump options are requested using more than one of the above, the order of selection is as follows:

1. CHNGDUMP options with OVER requests completely override any other request.
2. CHNGDUMP options with ADD requested are merged with options specified on ABEND, CALLRTM, or SETRP macros and this result is merged with options specified in IEAABD00, IEADMP00, or IEADMR00.
3. Lacking CHNGDUMP options, the options specified on the ABEND, CALLRTM or SETRP macros are merged with the options in IEAABD00, IEADMP00, or IEADMR00.
4. If no options were specified on the ABEND, CALLRTM or SETRP macros, the options specified in IEAABD00, IEADMP00 or IEADMR00 are used, If no options are specified via CHNGDUMP, ABEND, CALLRTM, SETRP, IEAABD00, IEADMR00, or IEADMP00, and symptom dump is not suppressed, a symptom dump is produced. If no options are specified and symptom dump is suppressed, no dump is taken.

For ABDUMPS the requestor (via ABEND, CALLRTM, and SETRP) and installation (via SYS1.PARMLIB members IEAABD00, IEADMP00, and IEADMR00) are given the ability to tailor dumps to the needs of the installation and the individual maintenance requirements of each error type. In addition, the CHNGDUMP command provides a means of temporarily overriding options specified by the requestor and/or installation.

Figure 2 (Part 3 of 3). ABEND/SNAP Dump Processing



This figure ties together system initialization, option modification by operator, multiple address space processing, and multiple tasks as parts of the SVC DUMP process. The process of how a dump is initiated for a task in a single address space (or in multiple address spaces different from the requestor's) is explained by the control flow and data flow shown here.

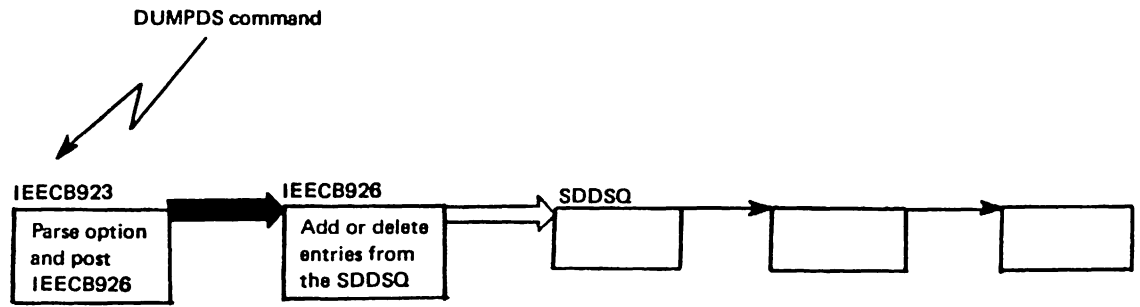


Figure 3 (Part 1 of 3). SVC Dump Overview

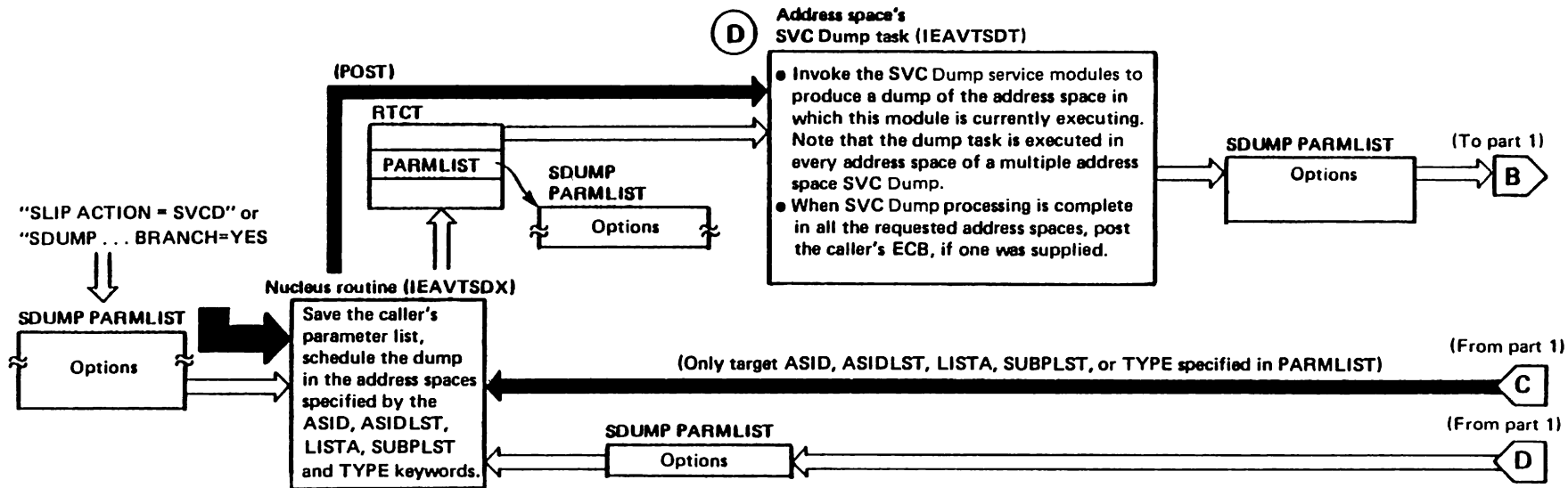


Figure 3 (Part 2 of 3). SVC Dump Overview

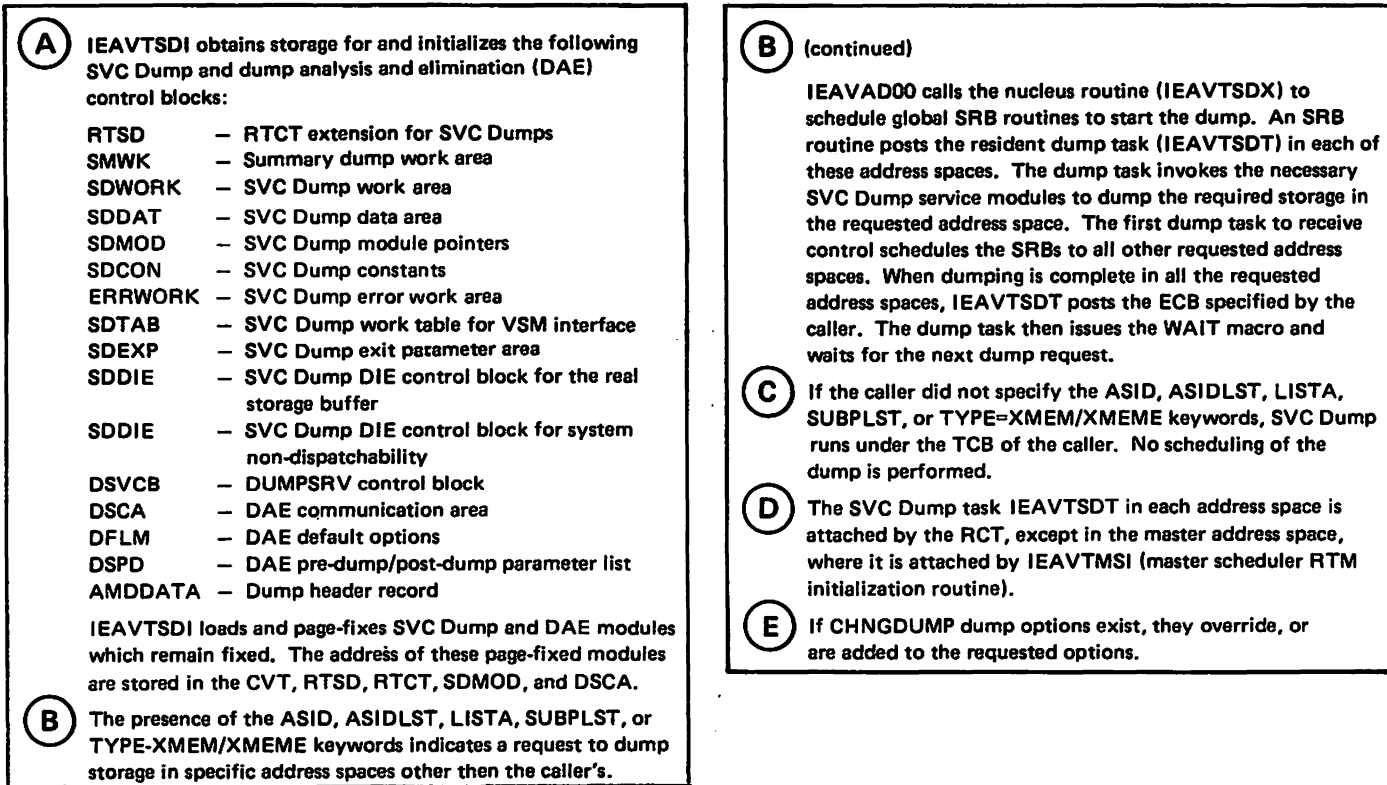


Figure 3 (Part 3 of 3). SVC Dump Overview

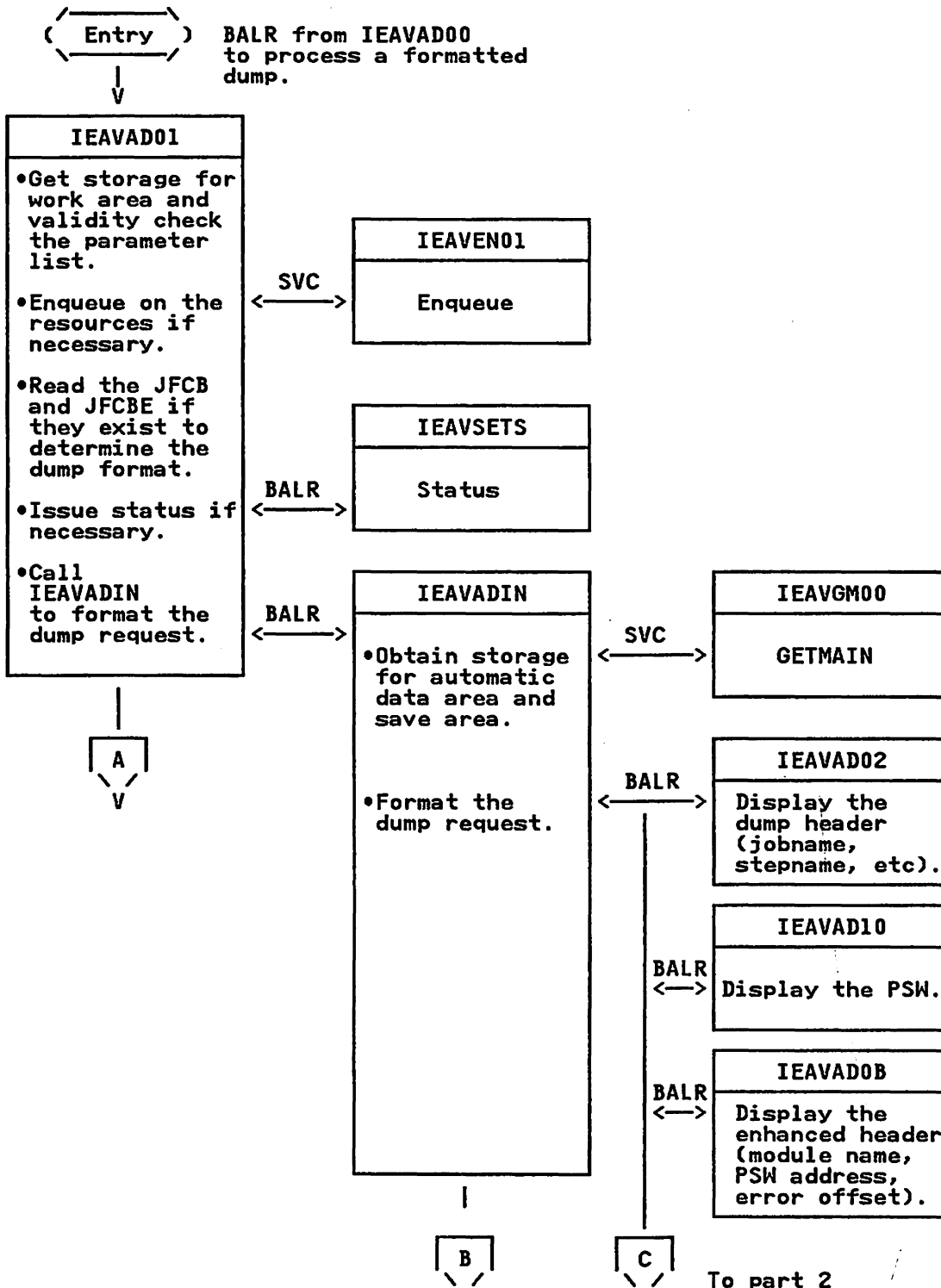


Figure 4 (Part 1 of 4). Formatted Dump Module Flow

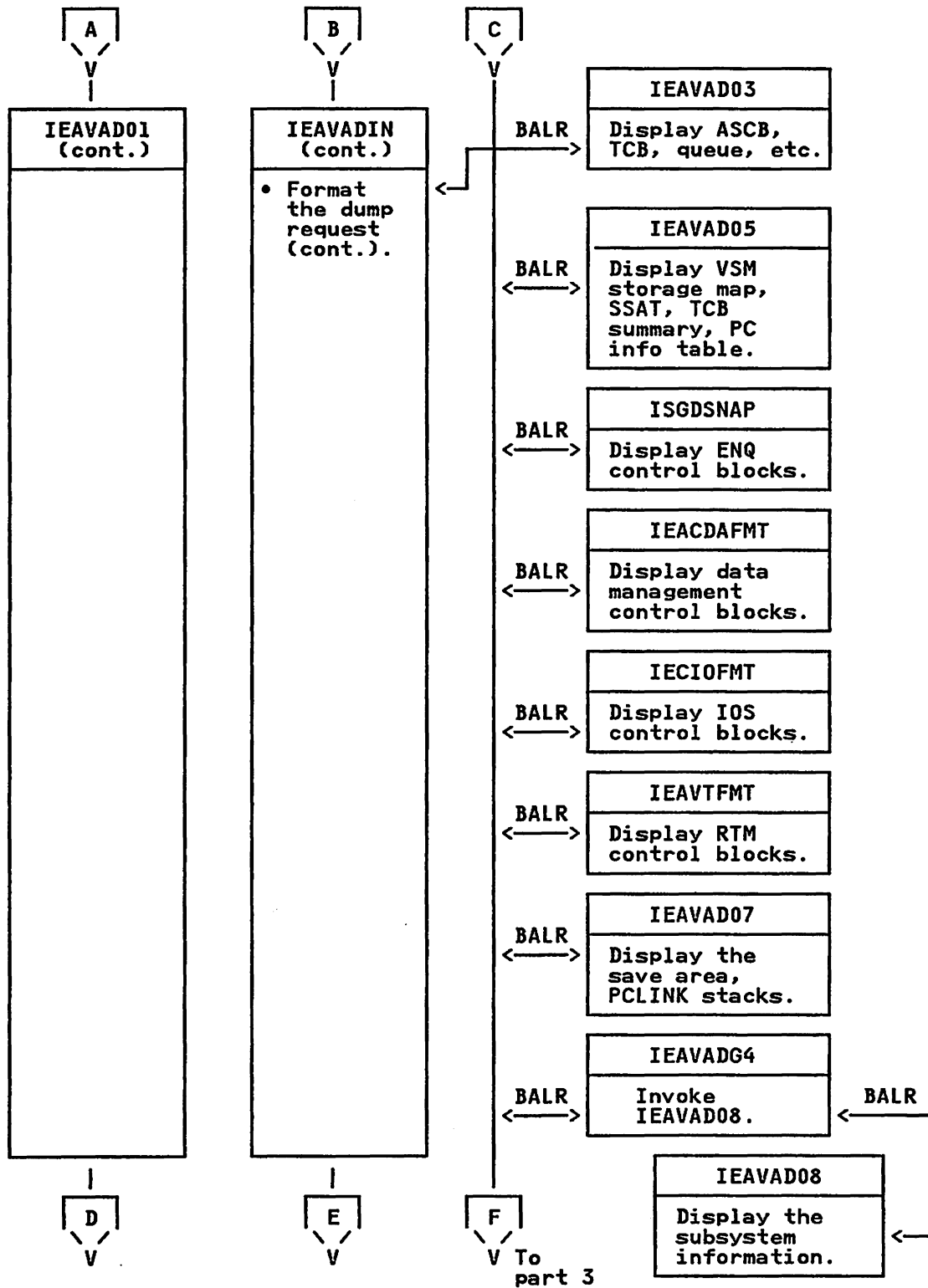


Figure 4 (Part 2 of 4). Formatted Dump Module Flow

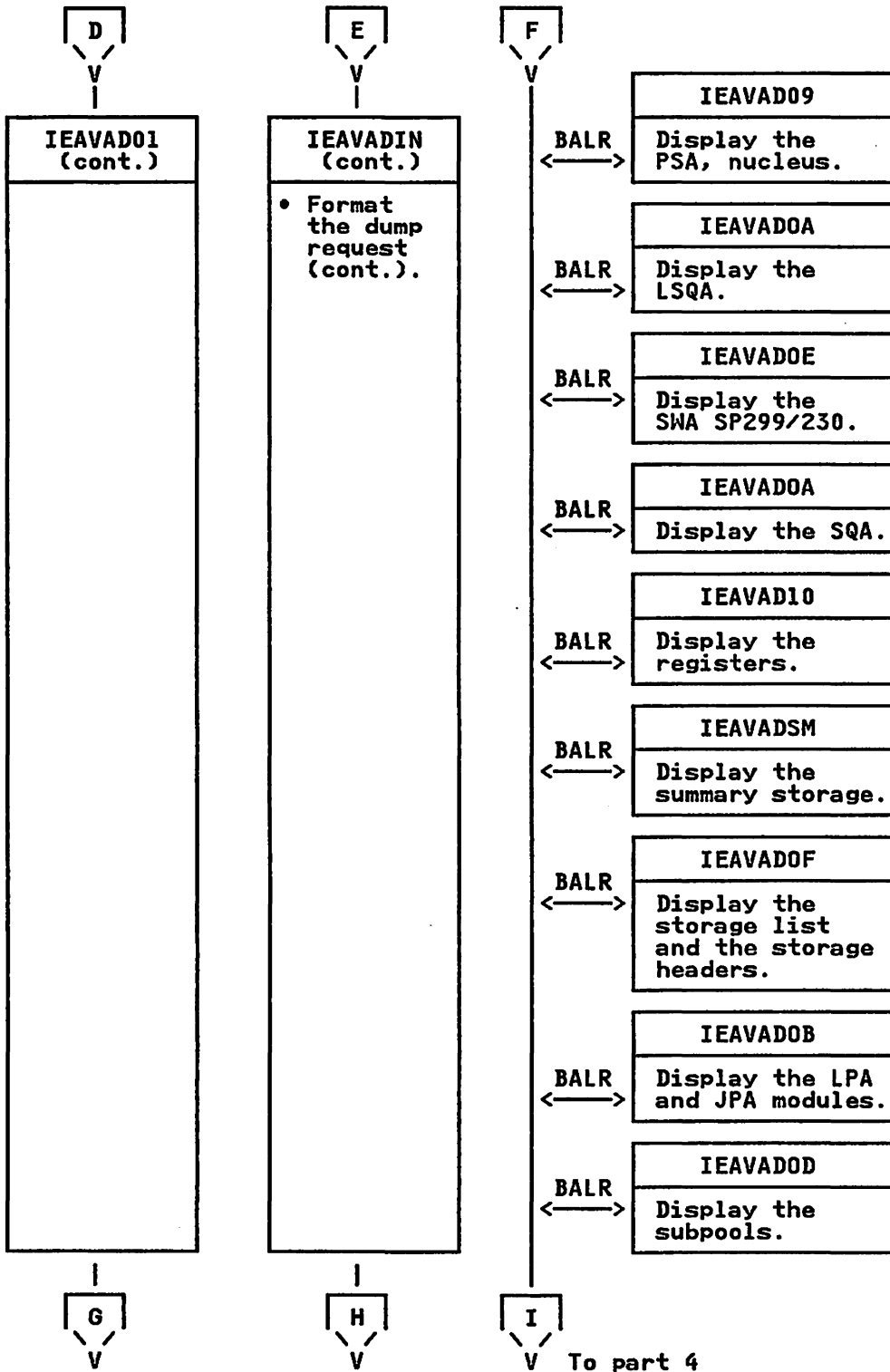


Figure 4 (Part 3 of 4). Formatted Dump Module Flow

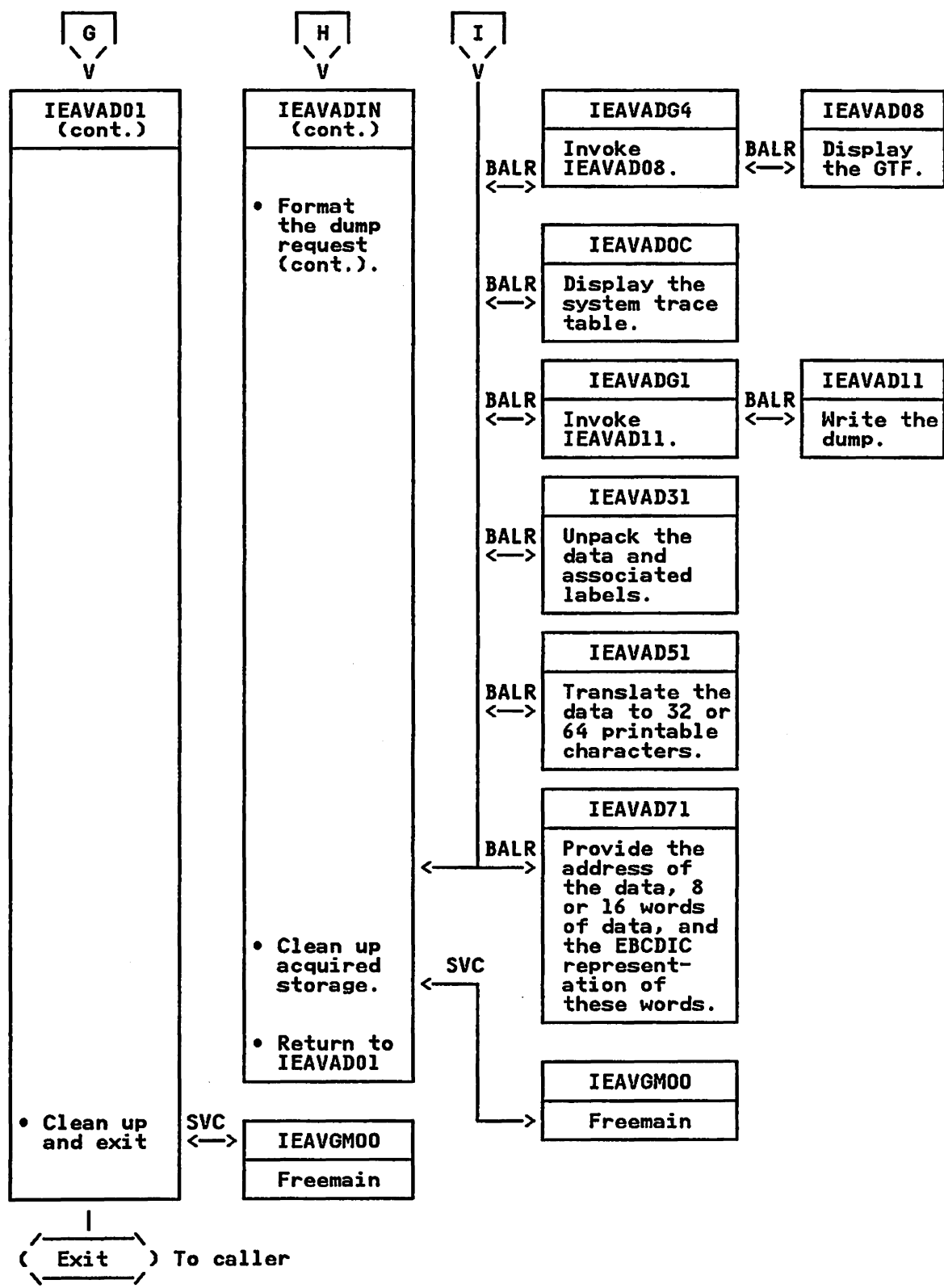


Figure 4 (Part 4 of 4). Formatted Dump Module Flow

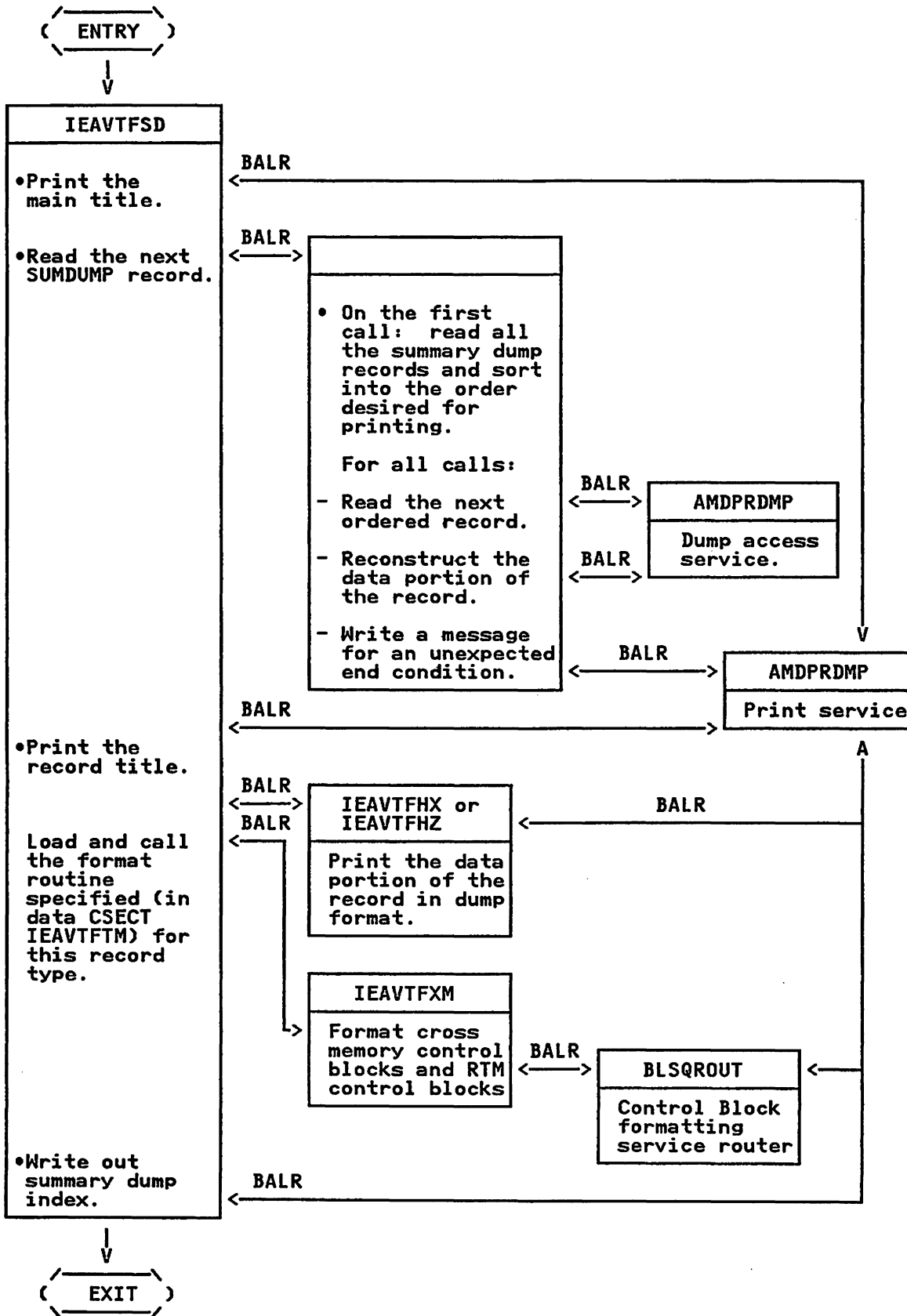


Figure 5. Summary SVC Dump (SUMDUMP) Formatter for the AMDPRDMP Service Aid

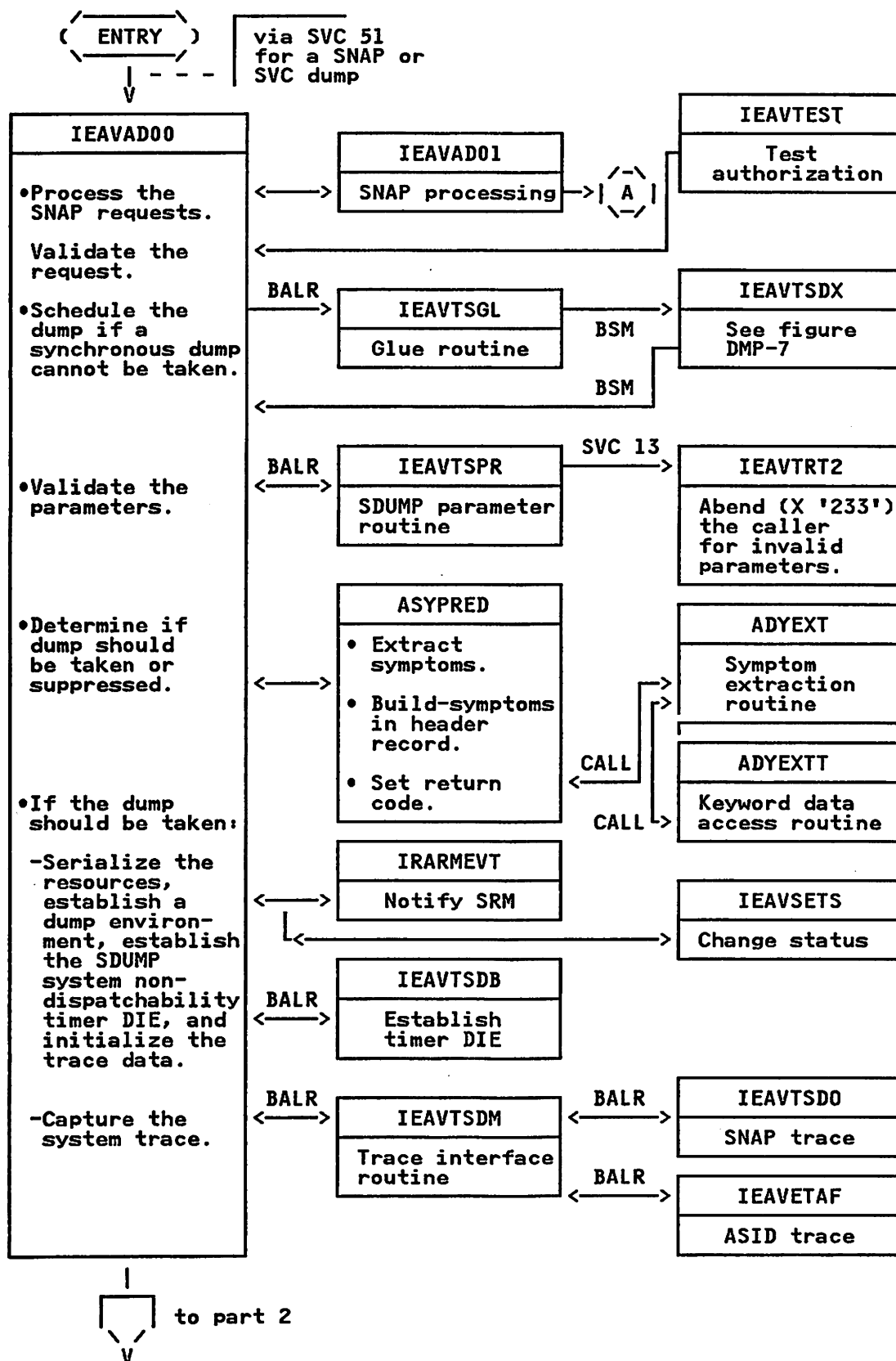


Figure 6 (Part 1 of 4). Synchronous Unformatted Dump Module Flow

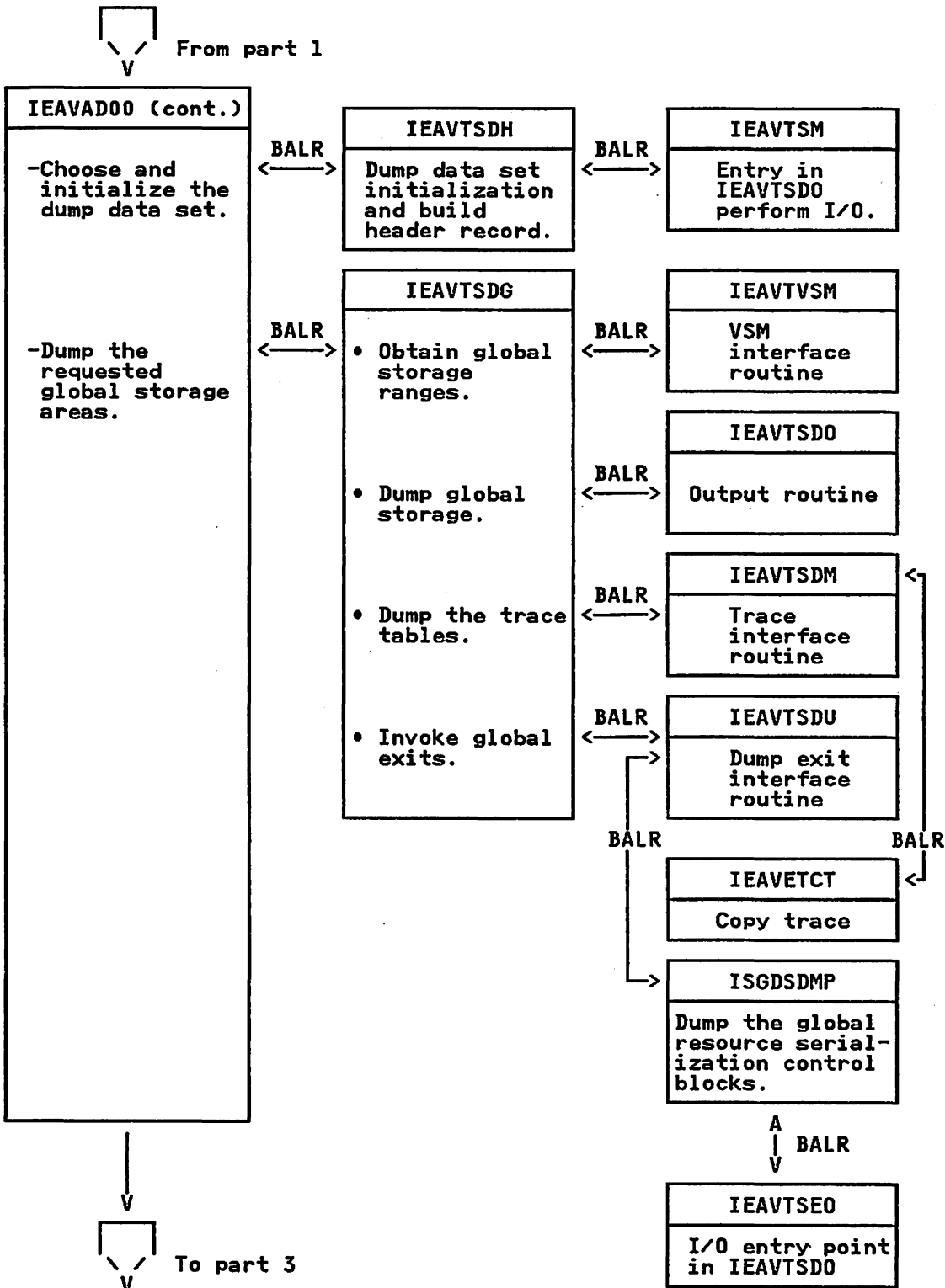


Figure 6 (Part 2 of 4). Synchronous Unformatted Dump Module Flow

From part 2

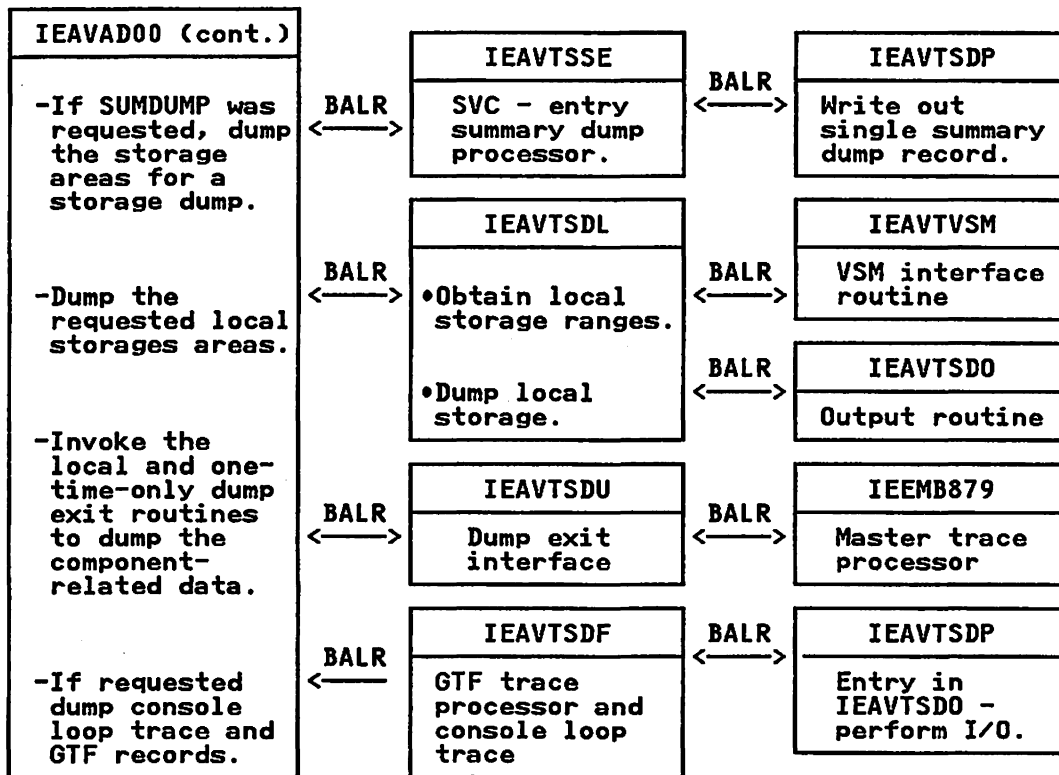


Figure 6 (Part 3 of 4). Synchronous Unformatted Dump Module Flow

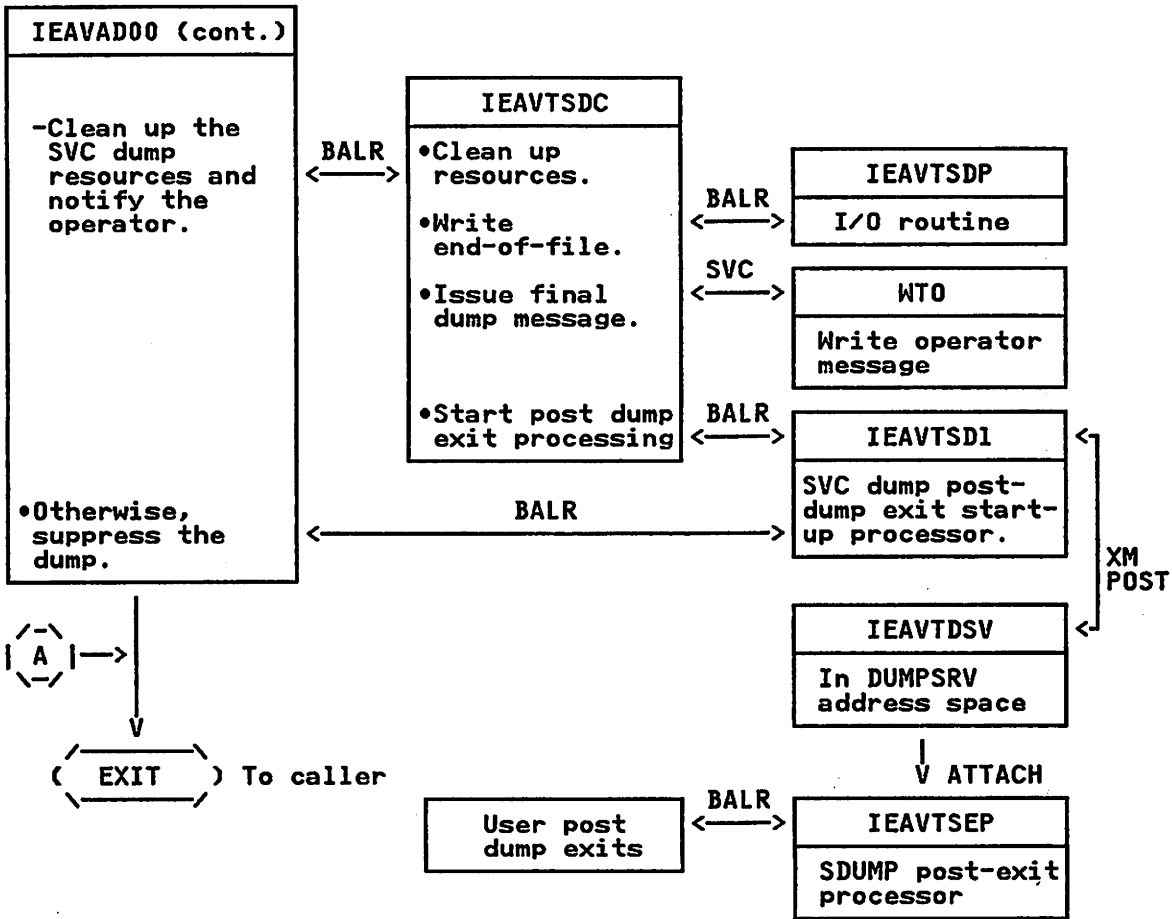


Figure 6 (Part 4 of 4). Synchronous Unformatted Dump Module Flow

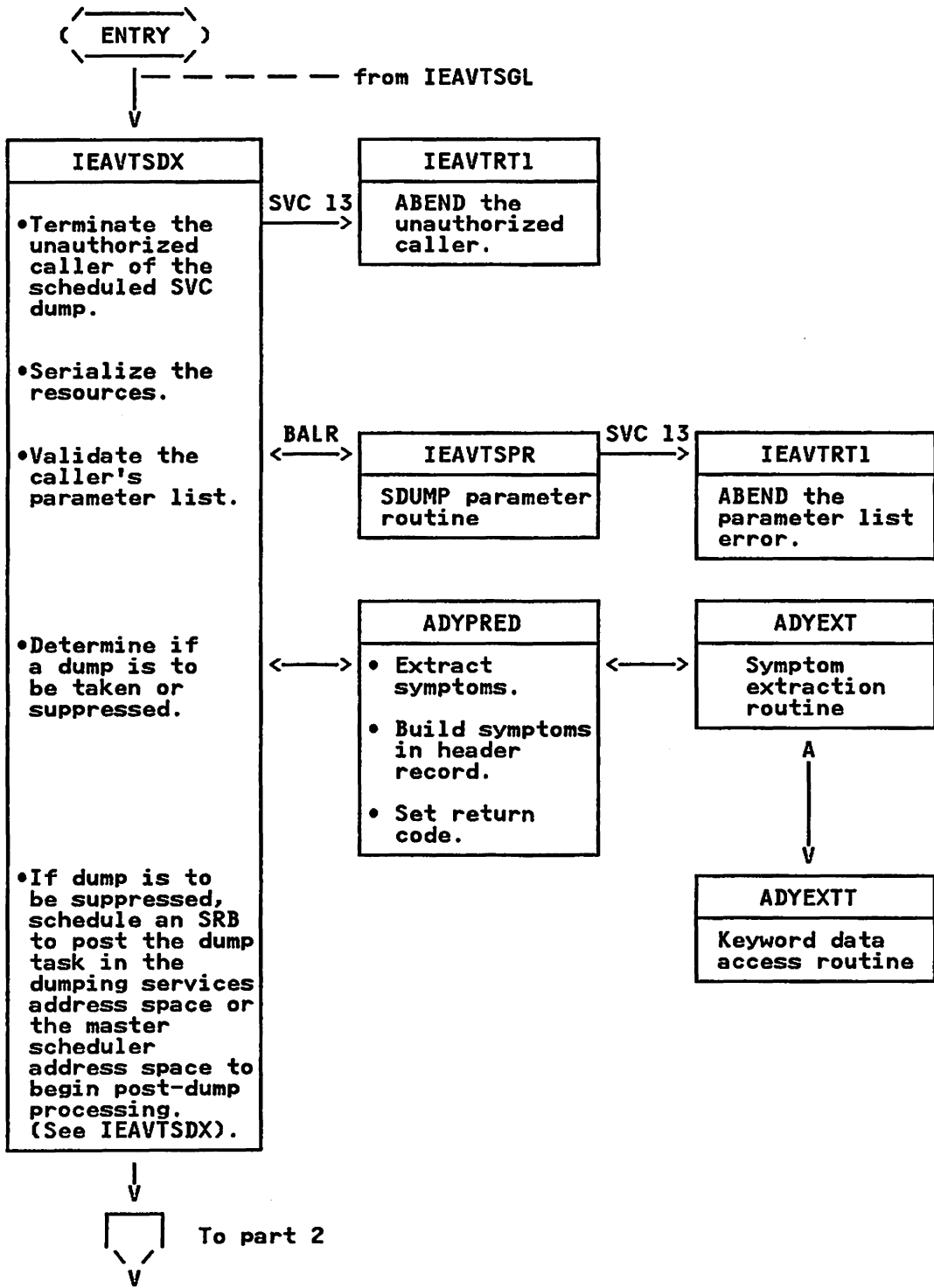


Figure 7 (Part 1 of 8). Scheduled Unformatted Dump Module Flow.

From part 1

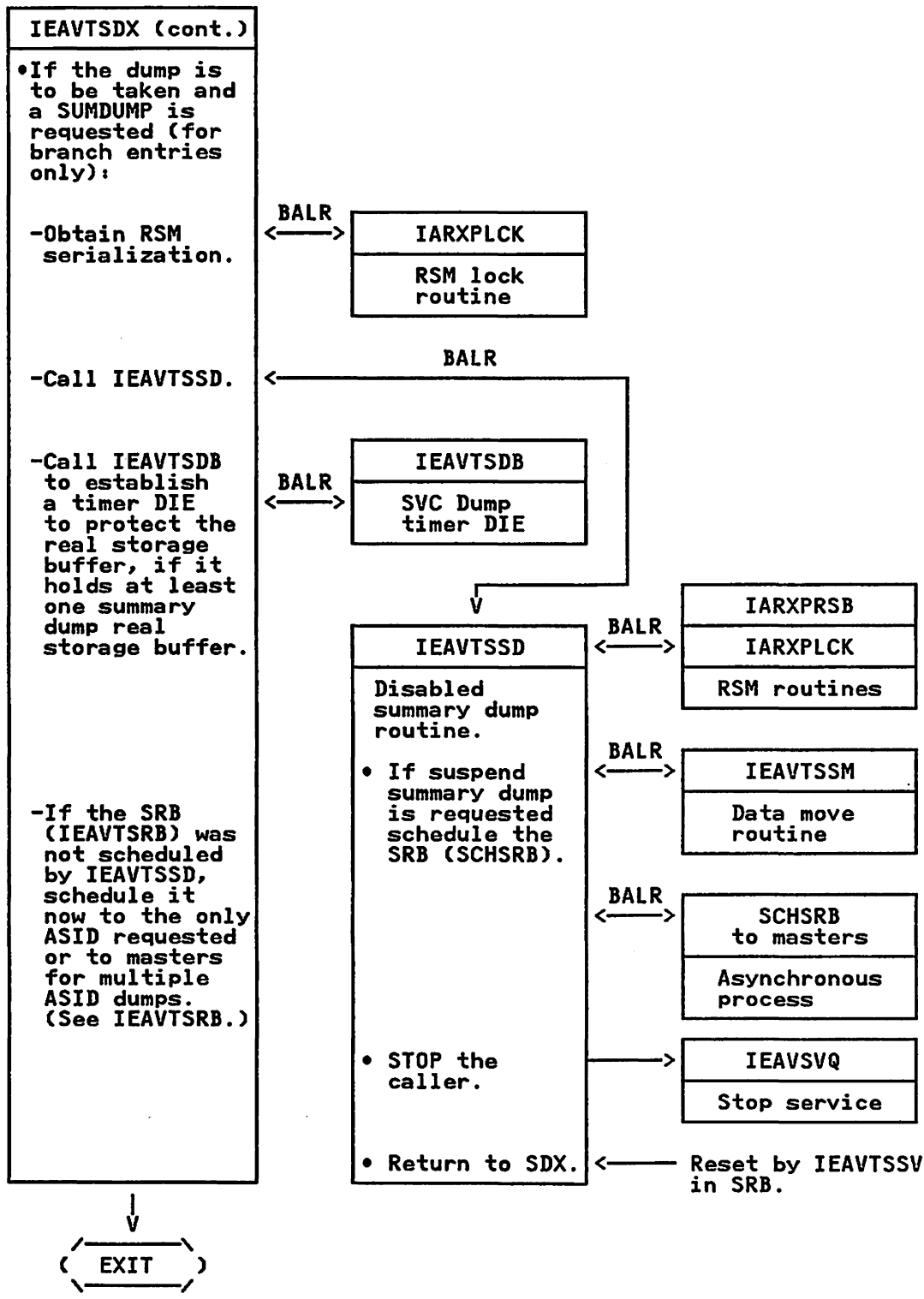


Figure 7 (Part 2 of 8). Scheduled Unformatted Dump Module Flow.

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

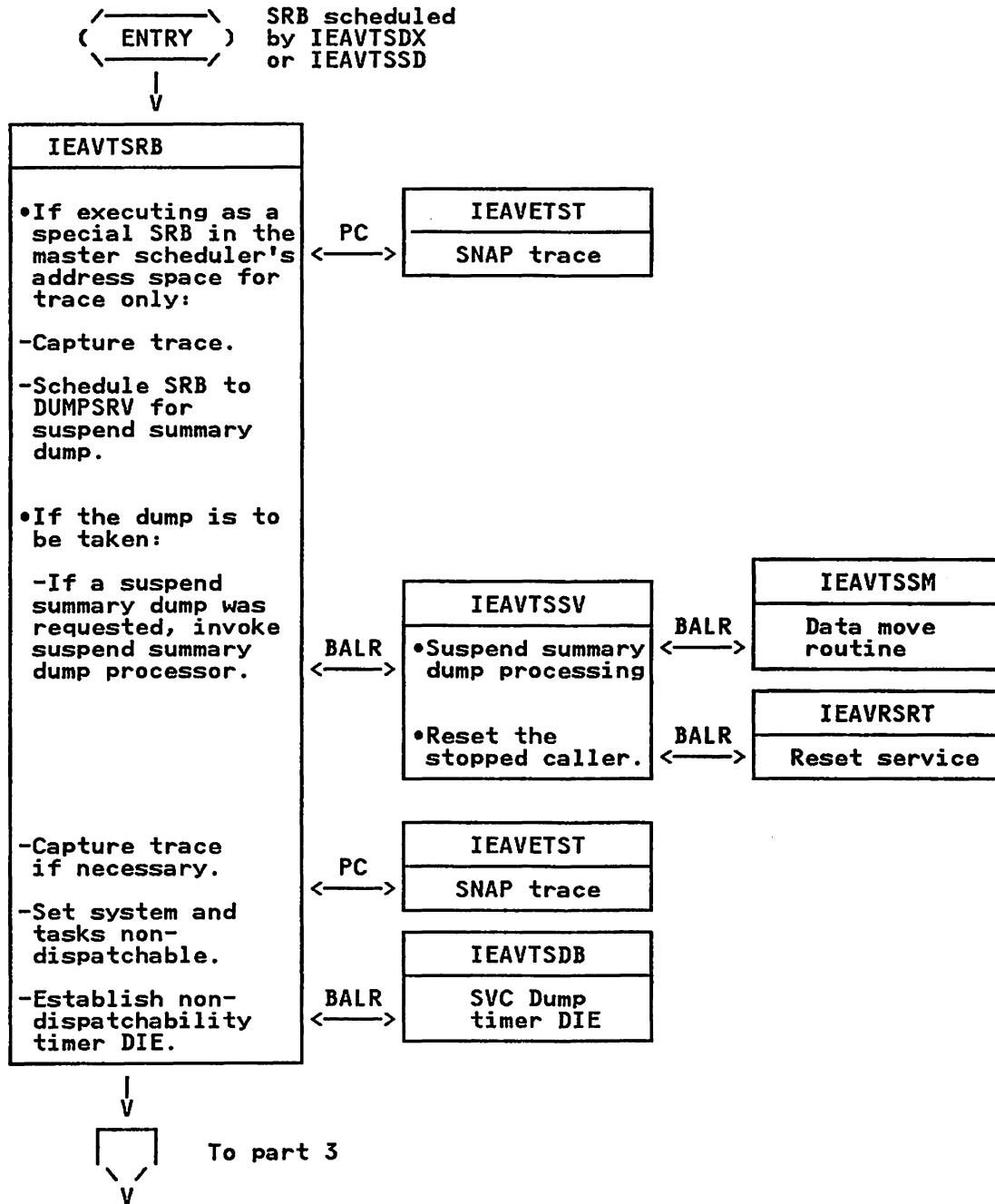


Figure 7 (Part 3 of 8). Scheduled Unformatted Dump Module Flow.

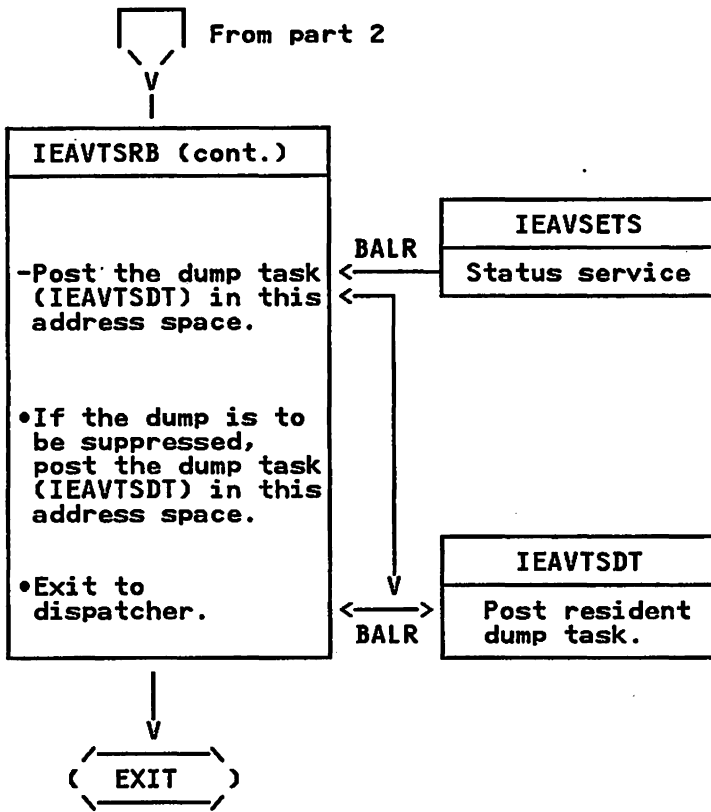


Figure 7 (Part 4 of 8). Scheduled Unformatted Dump Module Flow.

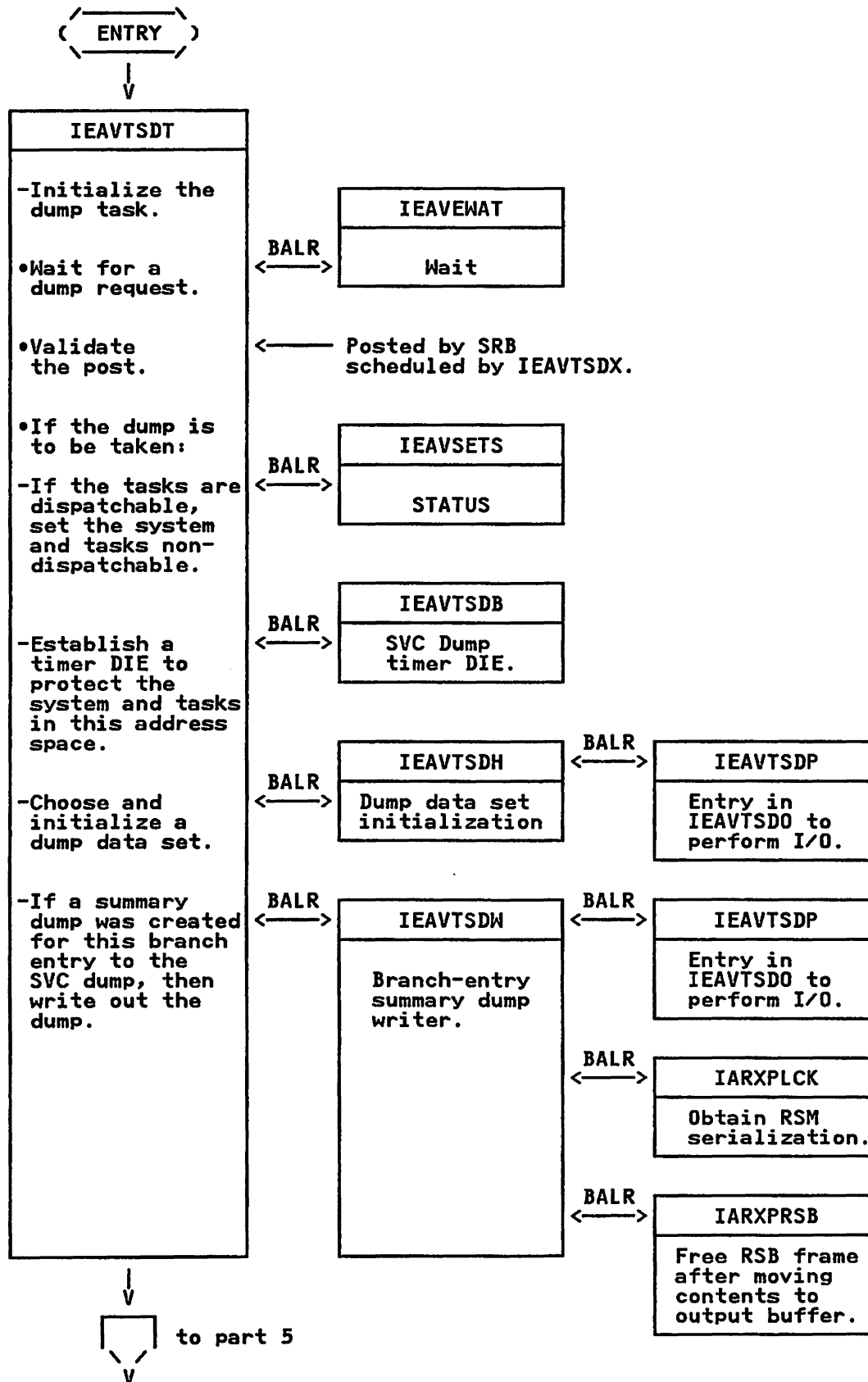


Figure 7 (Part 5 of 8). Scheduled Unformatted Dump Module Flow.

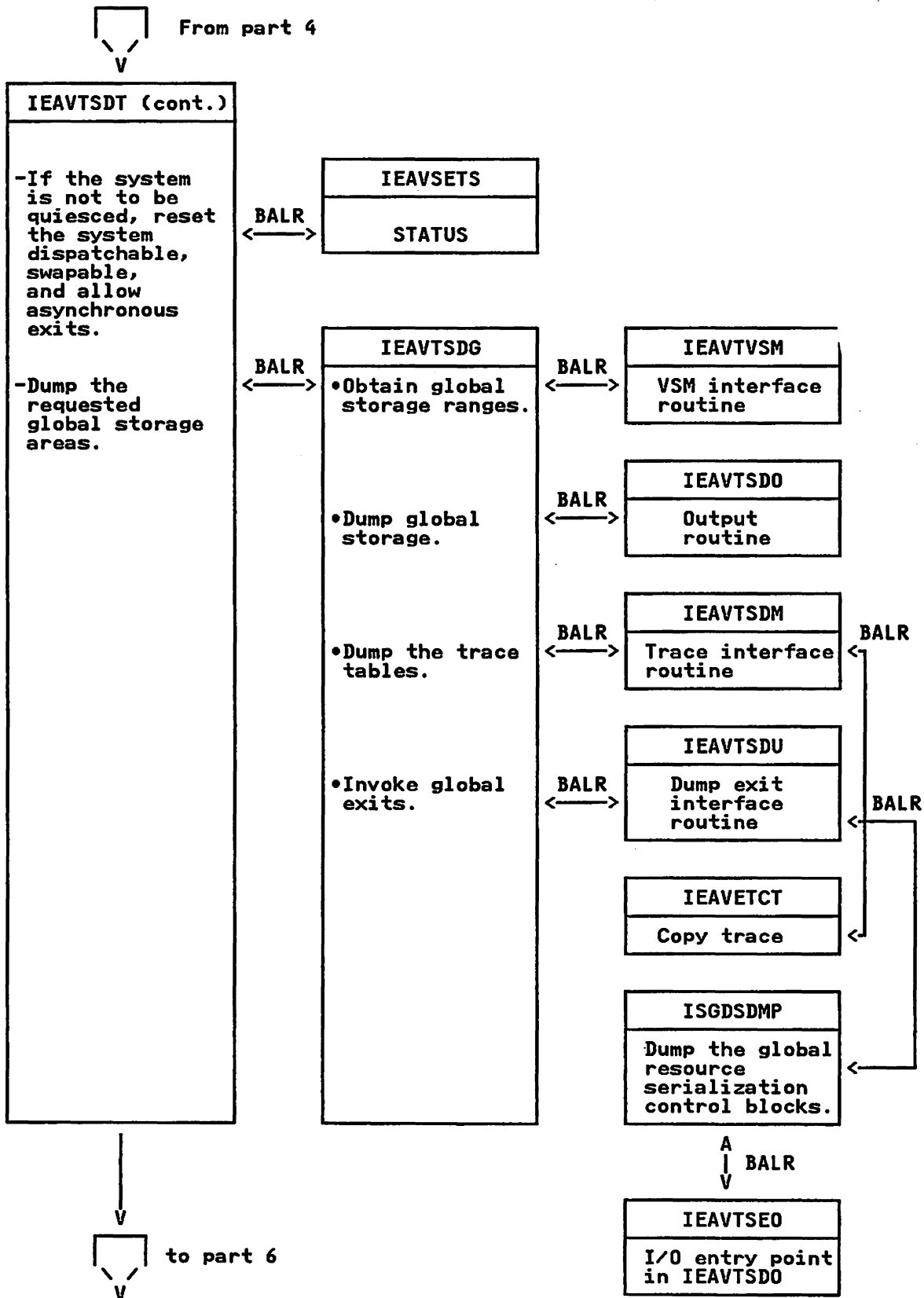


Figure 7 (Part 6 of 8). Scheduled Unformatted Dump Module Flow.

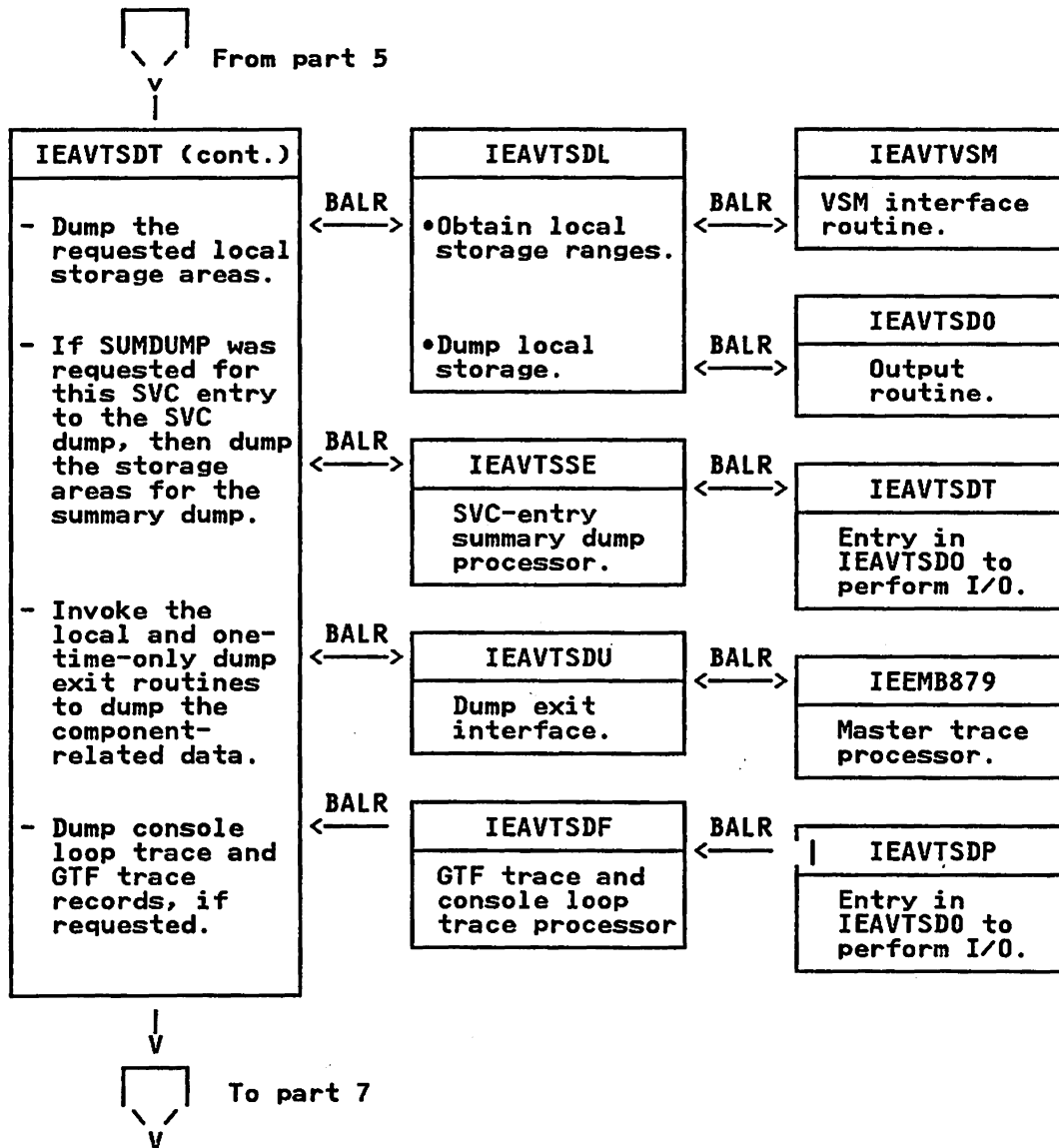


Figure 7 (Part 7 of 8). Scheduled Unformatted Dump Module Flow.

From part 6
 V

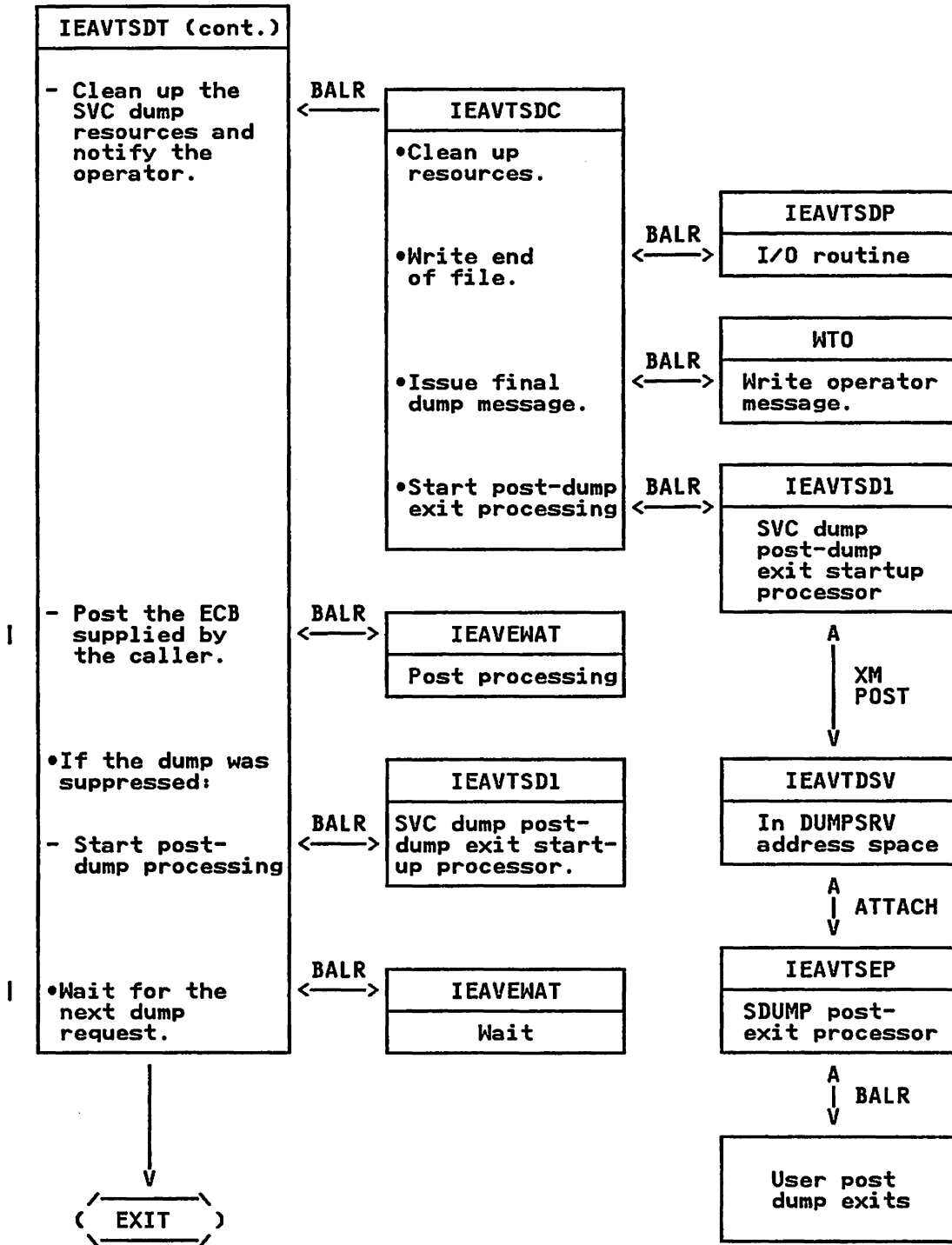


Figure 7 (Part 8 of 8). Scheduled Unformatted Dump Module Flow.

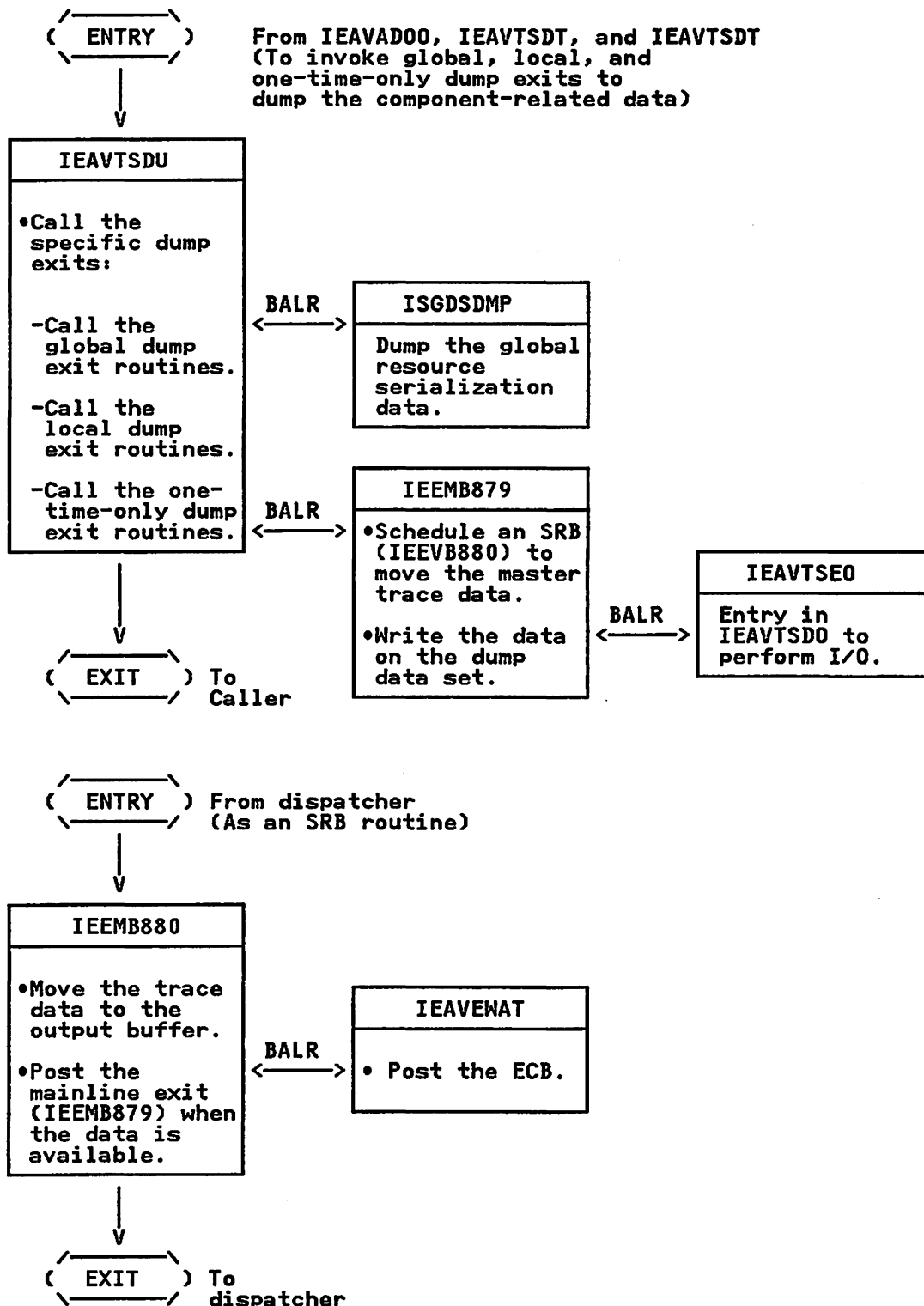


Figure 8. Master Trace Dump Exit Processing

CHNGDUMP routine

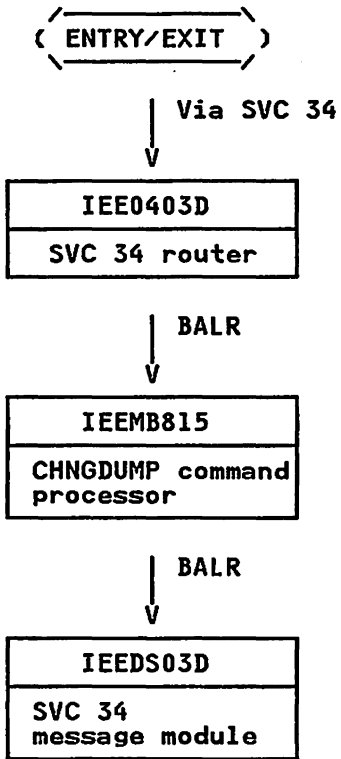


Figure 9. RTM Services Module Flow

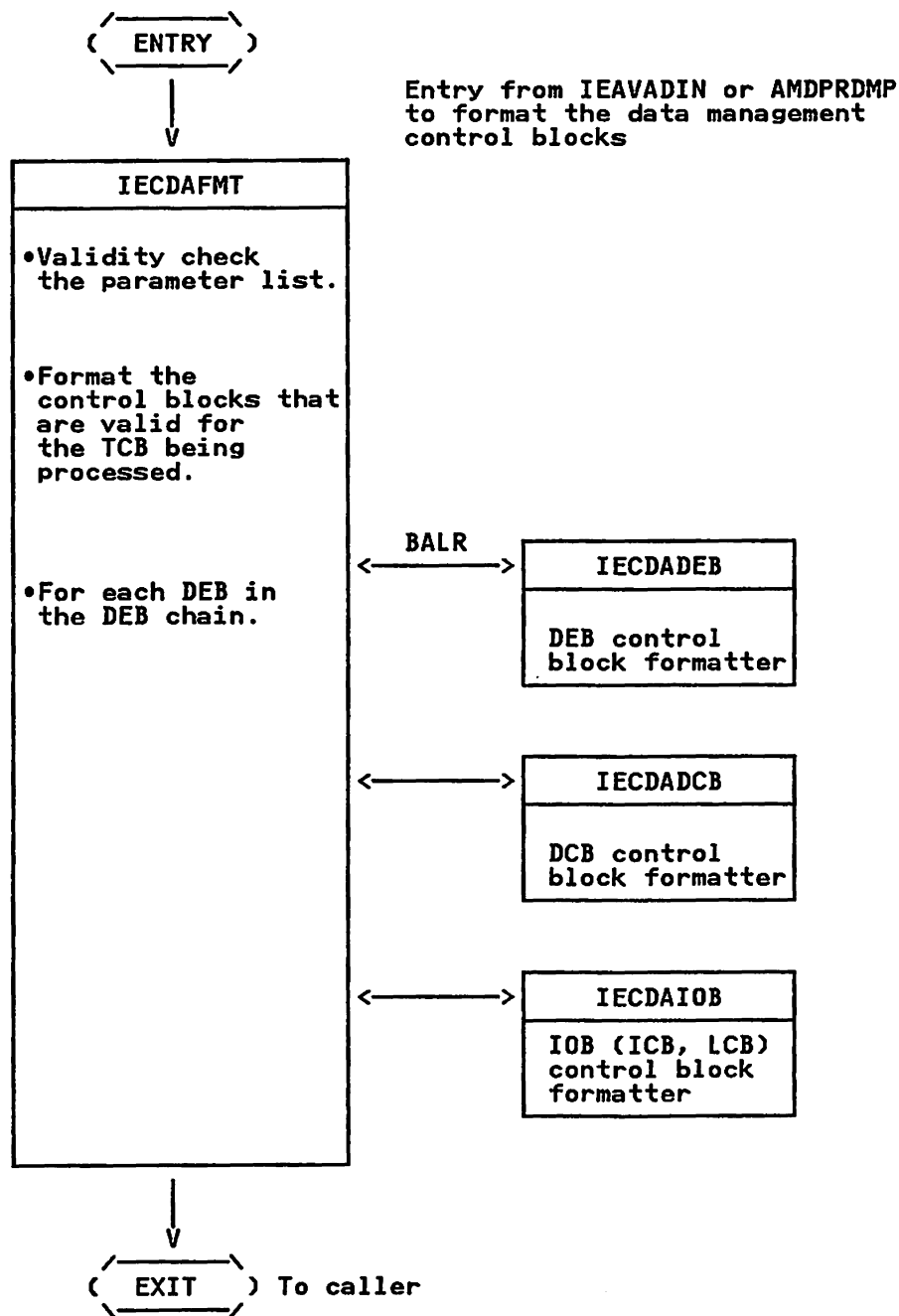


Figure 10. Data Management Control Block Formatter

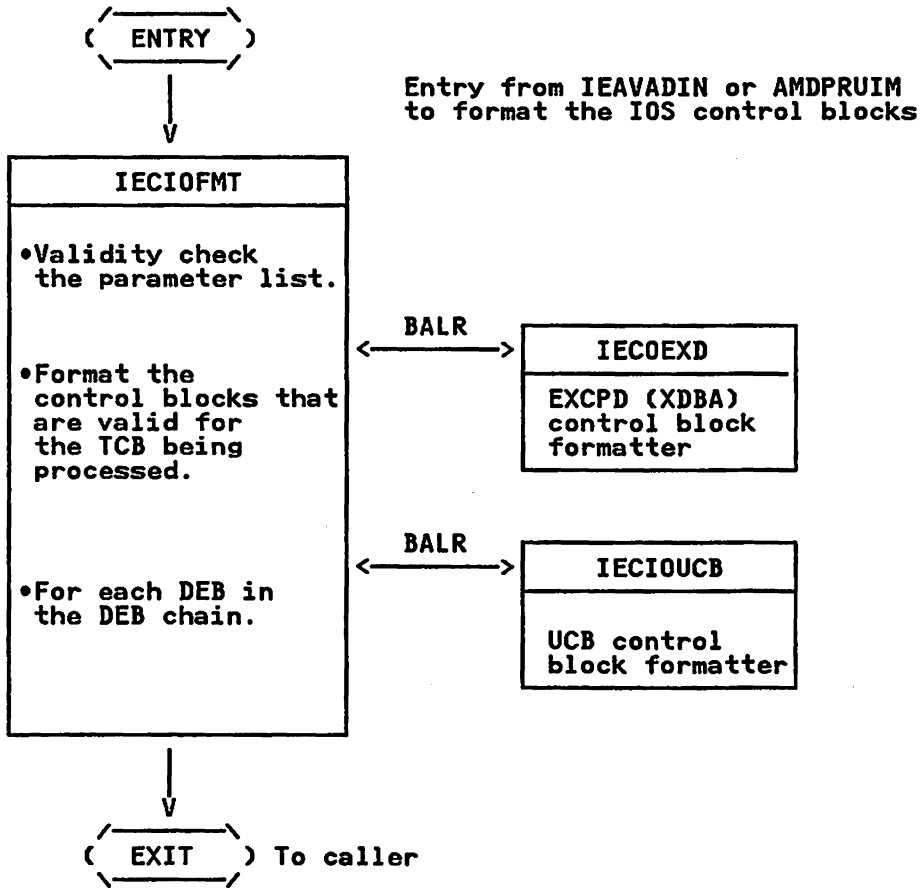


Figure 11. IOS Control Block Formatter

METHOD OF OPERATION

This section contains logic diagrams for the modules in this component.

The modules are in alphabetic order, and the diagrams use either hipo format or prologue format.

The following figure shows the symbols used in hipo format logic diagrams. The relative size and the order of fields in control block illustrations do not always represent the actual size and format of the control block.

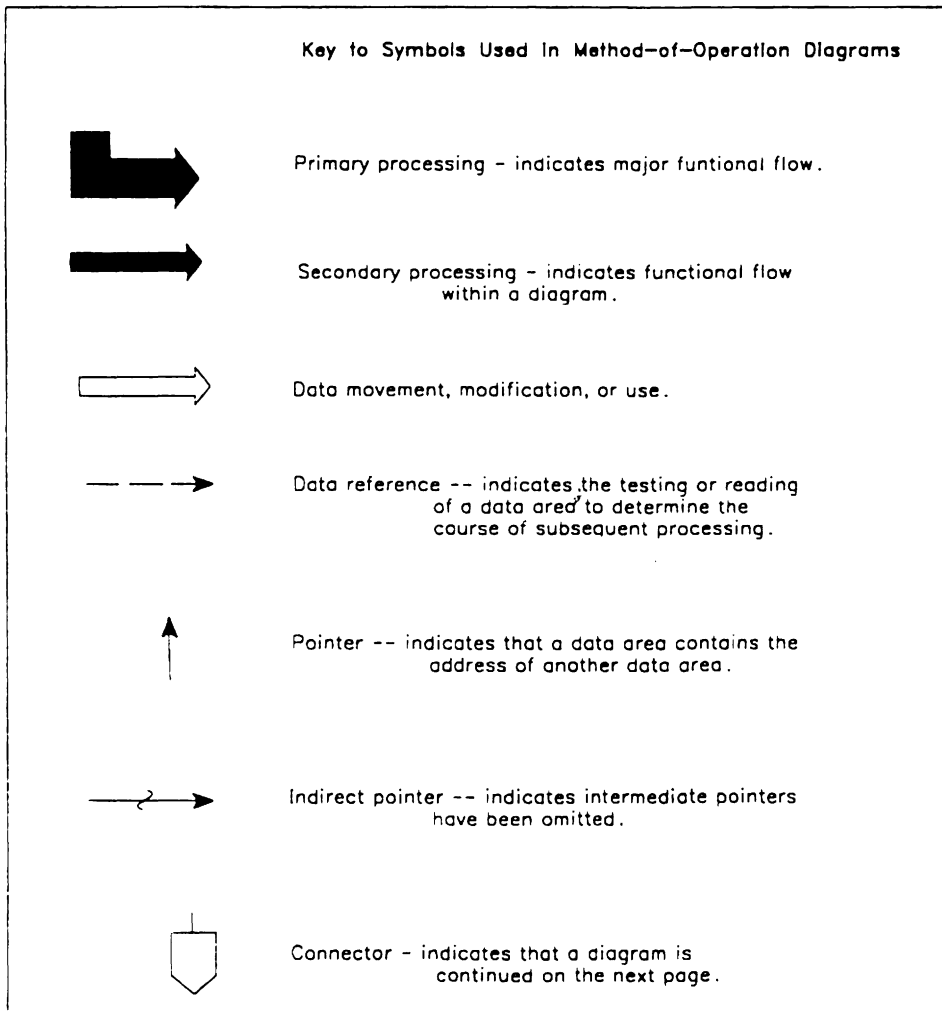


Figure 12. Key to Hipo Logic Diagrams

The prolog format diagrams contain detailed information that is broken down into three different headings. The three headings and the topics they document are:

Module Description, which includes:

- Descriptive name
- Function (of the entire module)
- Entry point names, which includes:
 - Purpose (of the entry point)
 - Linkage
 - Callers
 - Input
 - Output
 - Exit normal
 - Exit error, if any
- External references, which includes:
 - Routines
 - Data areas, if any
 - Control blocks
- Tables
- Serialization

Note: Brief DMP module descriptions are also included in MVS/Extended Architecture System Logic Library: Module Descriptions, which contains module descriptions for all the MVS/Extended Architecture components described in the System Logic Library.

Module Operation, which includes:

- Operation, which explains how the module performs its function.
- Recovery operation, which explains how the module performs any recovery.

Diagnostic aids, which provide information useful for debugging program problems; this includes:

- Entry point names
- Messages
- Abend codes
- Wait state codes
- Return codes for each entry point. Within each entry point, return codes might be further categorized by exit-normal and exit-error.
- Entry register contents for each entry point
- Exit register contents for each entry point

Logic Diagram, which illustrates the processing of the module, the input it uses, the output it produces, and the flow of control. Some modules do not have a logic diagram because the processing is sufficiently explained in the module description, the module operation, and the diagnostic aids sections. Figure 13 on page DMP-51 illustrates the graphic symbols and format used in the logic diagrams.

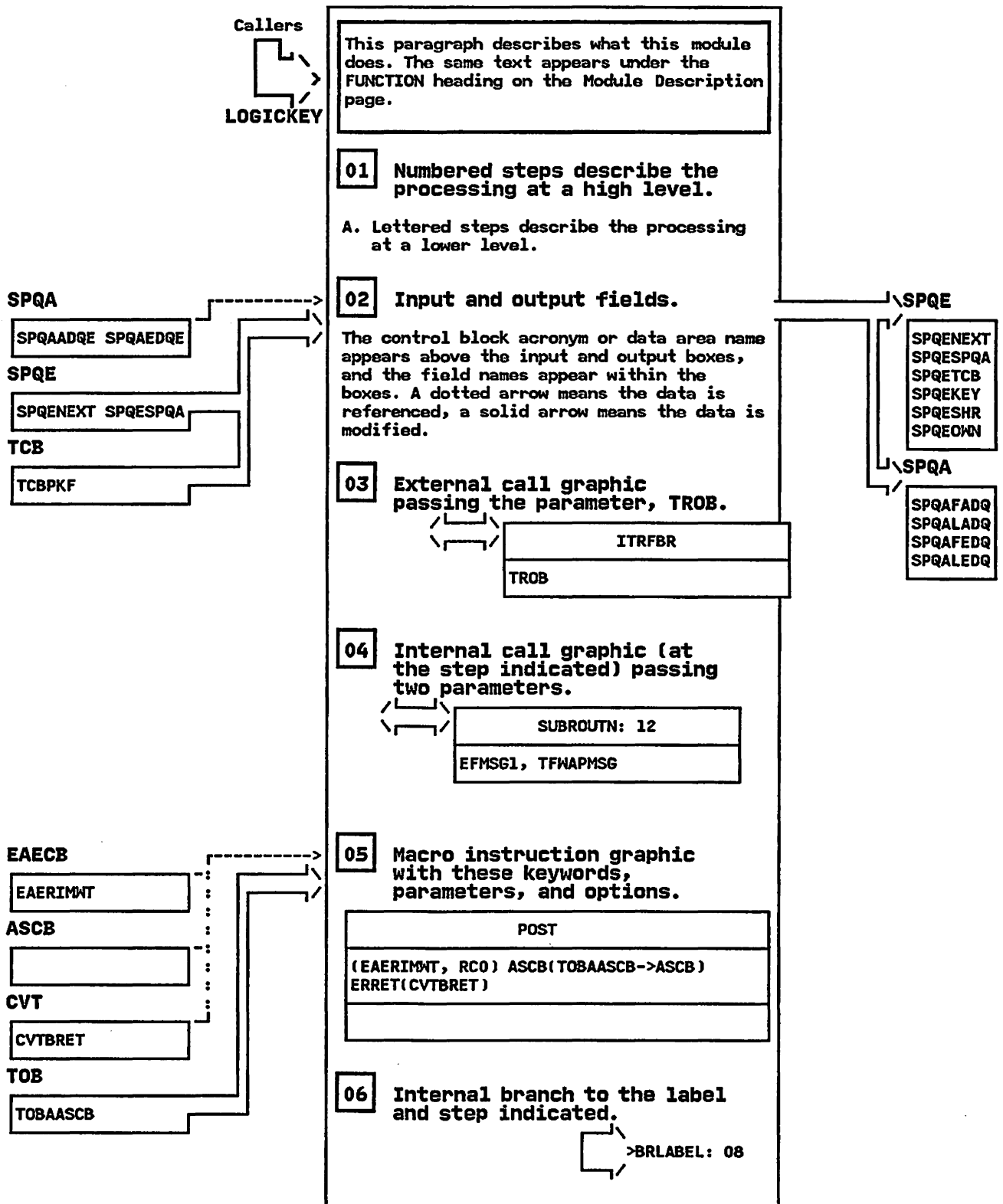


Figure 13. Key to the Logic Diagrams (Part 1 of 2)

LOGICKEY - Key to the Logic Diagrams

STEP 07

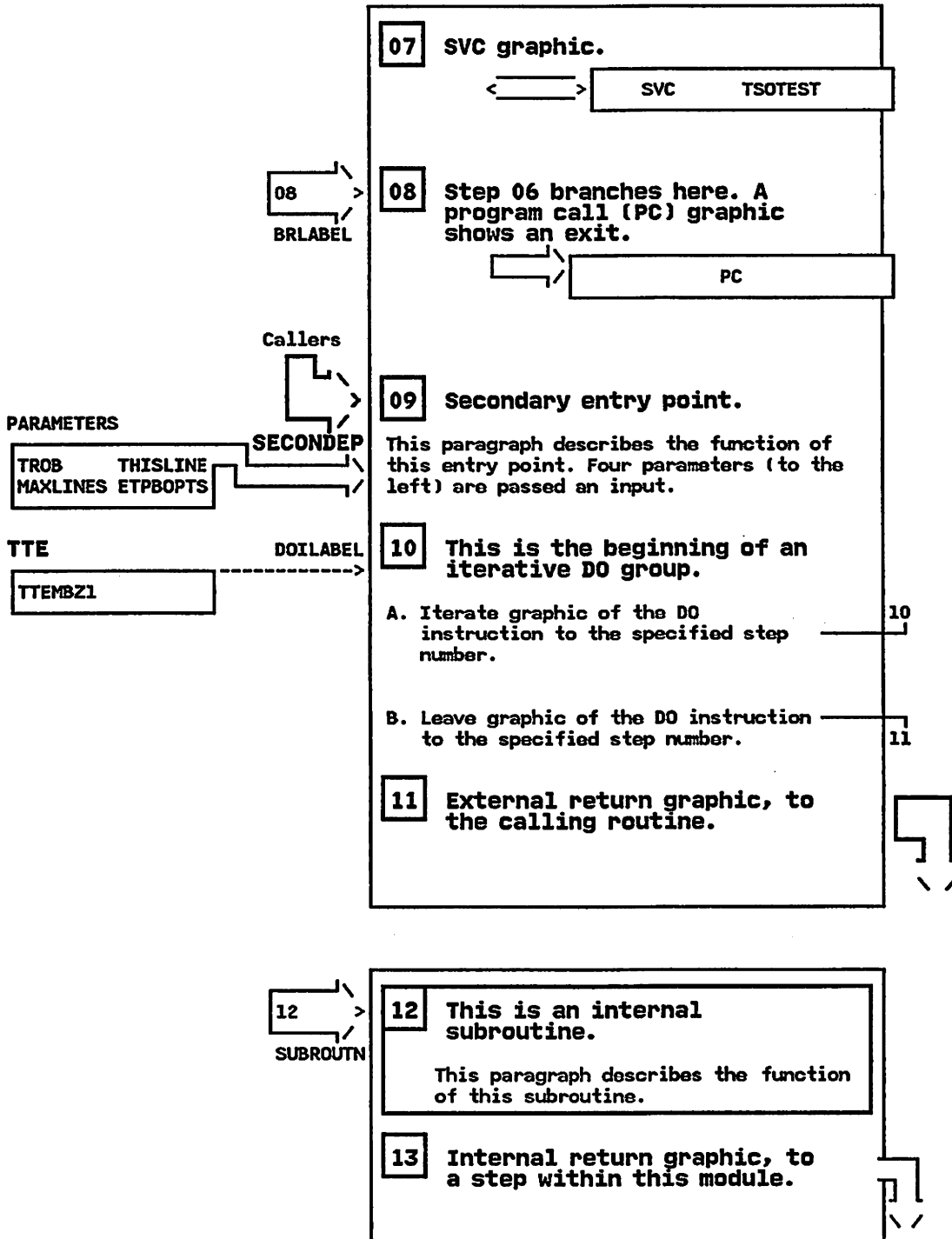
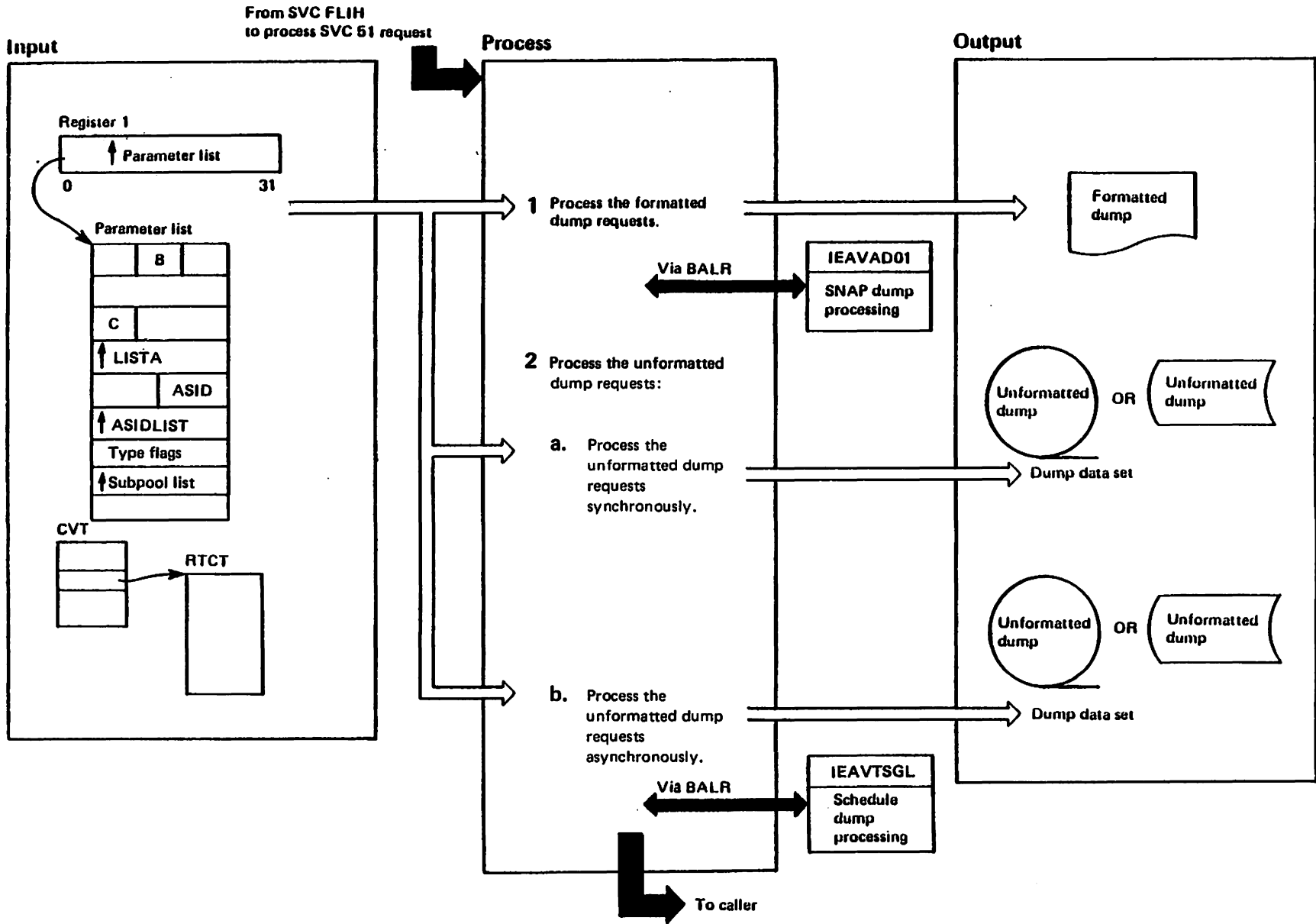


Figure 13. Key to the Logic Diagrams (Part 2 of 2)

This page intentionally left blank.

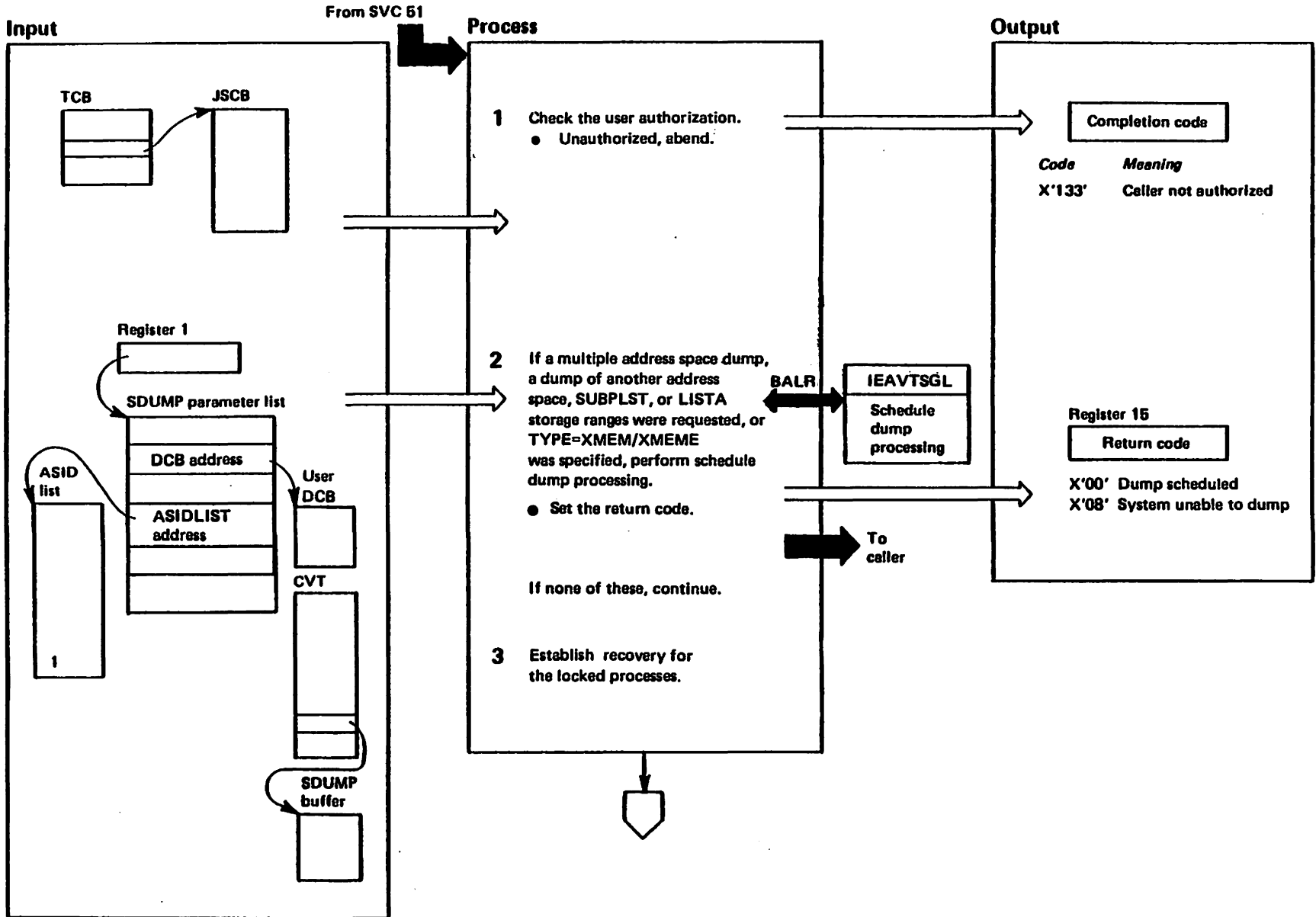
IEAVAD00 - SVC 51 Overview (Part 1 of 2)



IEAVAD00 - SVC 51 Overview (Part 2 of 2)

Extended Description	Module	Label	Extended Description	Module	Label
<p>An SVC 51 instruction provides linkage to both the SNAP function and to the SVCDUMP function. Both functions require that register 1 point to a parameter list. The SVC FLIH enters SVC51 processing holding the local lock in 31 bit addressing mode. The distinction between a SNAP and an SVCDUMP parameter list is in byte 1 of the first word: if X'80', the parameter list is for SVCDUMP; any other value in byte 1 indicates that the parameter list is for SNAP.</p> <p>If SVC 51 processing finds a SNAP parameter list it calls SNAP processing. Otherwise, it calls SVCDUMP processing.</p>			<p>ASIDLST - A list of ASIDs to be dumped. LISTA - A list of ASIDs and related storage to be dumped. TYPE=XMEM - Dump the home, primary, and secondary ASIDs related to the caller at the time SDUMP is entered. TYPE=XMEME - Dump the home, primary, and secondary ASIDs related to the caller at the time of error. SUBPLST - A list of ASIDs and related subpools to be dumped. If the only ASID in this list is the caller's, this option will not force a scheduled dump.</p>		
<p>1 SVC 51 identified a SNAP parameter list. SVC 51 produces a formatted, synchronous dump of the requested storage area. The M.O. diagram Snap Dump Processing (IEAVAD01) describes this process in detail.</p>	IEAVAD00	SDTOP			
<p>2 SVC 51 identified an SVC dump request. Callers of SVC dump must be authorized by APF and have a control program key. SVC 51 processing abnormally terminates unauthorized callers with a completion code of X'133'.</p>	IEAVAD00	SDTOP	<p>a. If none of the above options are specified, a synchronous dump of the caller's address space is taken. The M.O. diagram SVC dump processing (IEAVAD00) explains the synchronous dump processing section of SVC dump. A synchronous dump is always taken when SVC 51 is issued by ABDUMP to process SYSMDUMPs.</p>	IEAVAD00	SDTOP
<p>SVC dump provides an unformatted storage dump of virtual storage areas for one to 15 address spaces. Two separate processes are used to provide these dumps. Synchronous dump processing (also called SVC dump processing) provides a synchronous, unformatted dump of the caller's address space.</p> <p>The options specified in the SDUMP parameter list determine whether a synchronous or scheduled dump is taken. The determining options are:</p>			<p>b. If any of the above options are specified, an asynchronous dump of all requested address spaces is taken. Issuing an SVC for an asynchronous dump request results in a call to the schedule dump routine (IEAVTSDX) through interface with module IEAVTSGL. Asynchronous dump requests are also made by branching directly to the schedule dump processing routine using the BRANCH=YES keyword on the SDUMP macro (SVC dump branch entry). Interface to IEAVTSDX is again through the module IEAVTSGL.</p>	IEAVTSGL	
<p>ASID - A single ASID specifying the address space to dump.</p>			<p>The M.O. diagram Schedule Dump Processing (IEAVTSDX) describes this process in detail.</p>		

IEAVAD00 - SVC Dump Processing (Part 1 of 12)

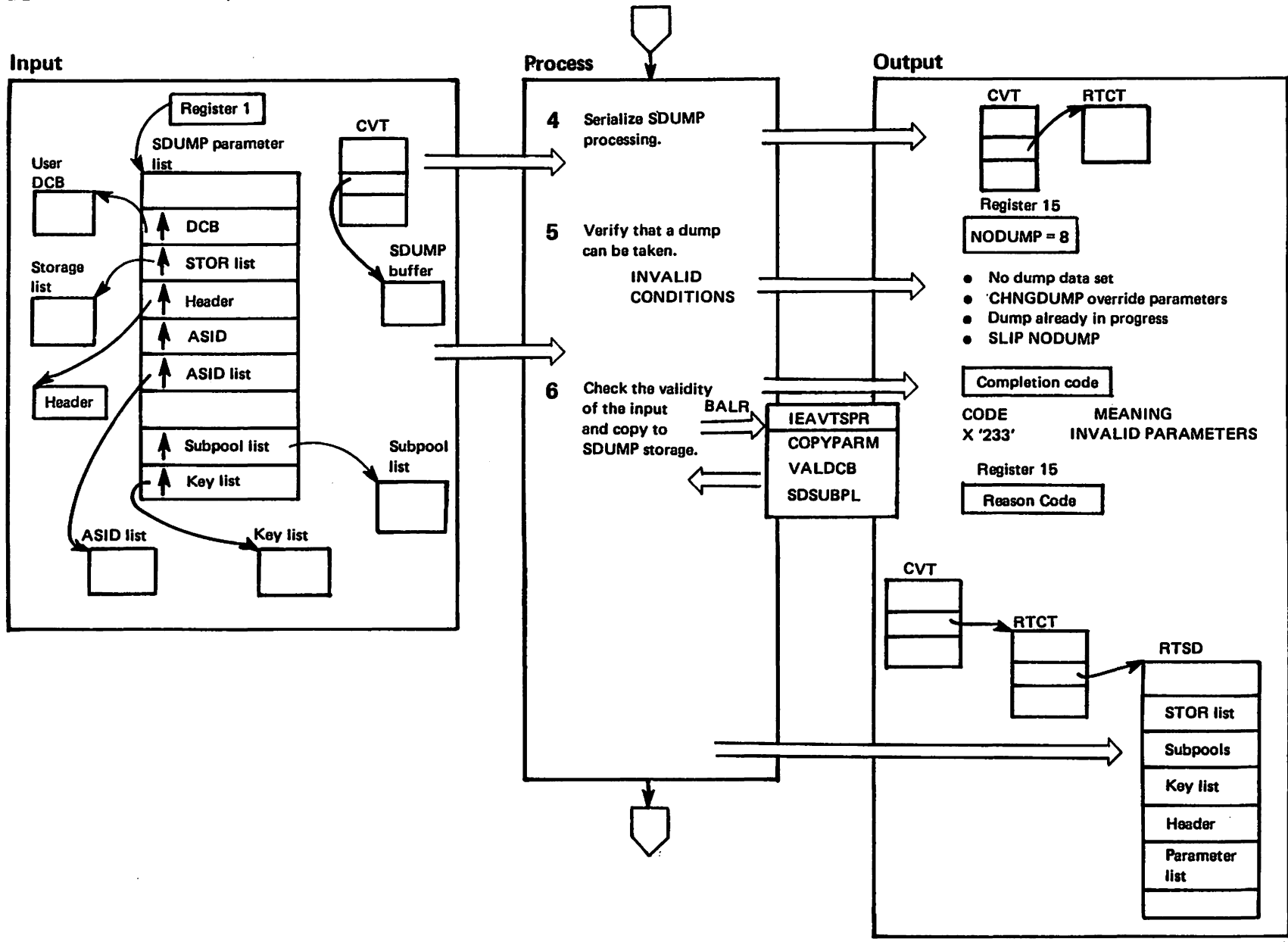


IEAVAD00 - SVC Dump Processing (Part 2 of 12)

Extended Description	Module	Label
<p>SVC dump processing creates a synchronous unformatted dump of virtual storage and writes it on a data set. SVC dump receives control from SVC 51 when the parameter list addressed in register 1 indicates an unformatted dump.</p>		
<p>1 SVC dump uses TESTAUTH to verify the authorization of the caller. Callers of SVC dump must be APF authorized and running in keys 0-7. IEAVAD00 abends callers who are not authorized to use SVC dump with a system code of X'133'.</p>	<p>IEAVAD00</p>	<p>TOPSEG</p>
<p>2 If SVC 51 requests a schedule dump, IEAVAD00 invokes module IEAVTSDX through the addressing mode interface module IEAVTSGL. IEAVAD00 schedules a dump for the following:</p> <ul style="list-style-type: none"> ● A multiple address space dump ● A dump of an address space other than the current address space ● A request for LISTA storage ranges ● A request for TYPE=XMEN or TYPE=XMEN ● A request for SUBPLST with an ASID other than the current ASID 	<p>IEAVAD00 IEAVTSDX IEAVTSGL</p>	<p>TOPSEG</p>
<p>Upon return from IEAVTSDX, SVC dump returns control to the caller with a return code stating whether or not the dump is scheduled.</p>		
<p>3 IEAVAD00 uses SETFRR to establish a functional recovery routine (FRR) exit. This exit provides recovery for SVC dump processing, which runs disabled or locked.</p>	<p>IEAVAD00</p>	<p>TOPSEG SDFRRRTN</p>

IEAVAD00 - SVC Dump Processing (Part 3 of 12)

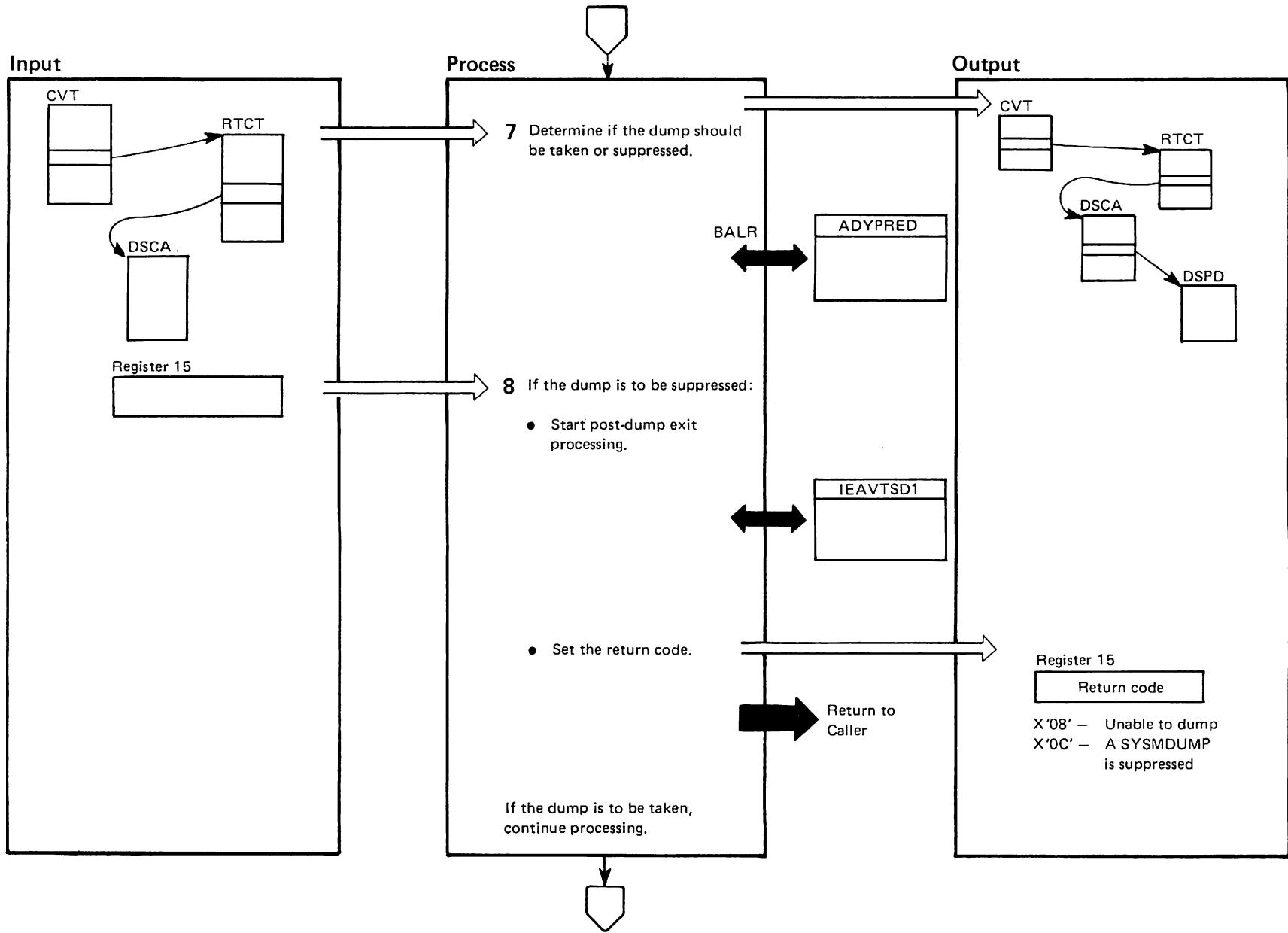
DMP-58 MVS/XA SLL: Dumping Ser LY28-1670-1 (c) Copyright IBM Corp. 1987, 1989



IEAVAD00 - SVC Dump Processing (Part 4 of 12)

Extended Description	Module	Label	Extended Description	Module	Label
<p>4 IEAVAD00 serializes SDUMP to prevent the system from trying to take concurrent SDUMPs. IEAVAD00 fills in the RTCTSDPL work with the address of the SDUMP parameter list and turns on the high-order bit of the CVTSDBF work. If either of these two serialization procedures fails, the dump is cancelled due to an SDUMP already in progress. For SYSMDUMP requests, IEAVAD00 does not check if SDUMP is already in progress. Multiple SYSMDUMPs are allowed because the SYSMDUMP process does not use or modify any SDUMP control blocks.</p> <p>5 IEAVAD00 examines the RTCT for the CHNGDUMP overriding parameter which specifies that no SVC dumps are to be taken. IEAVAD00 checks to verify that at least one dump data set has been defined, and checks system control blocks to make sure this dump is not being suppressed by the SLIP NODUMP process. If no dump can be taken, the caller is given a return code of 8 in register 15 and a reason code is stored in the SDWA.</p> <p>Note: For SYSMDUMP requests, SNAP/ABDUMP processing has already examined and merged the CHNGDUMP options.</p> <p>6 IEAVAD00 checks the validity of the parameter list and areas pointed to by the parameter list. It calls the SDUMP parameter list analysis routine (IEAVTSPR) to perform the following functions in the COPYPARM subroutine:</p> <ul style="list-style-type: none"> ● Clear the parameter area of the RTSD ● Copy the SDUMP parameter list to the RTSD ● Copy the header to the RTSD ● Copy the user's storage list to the RTSD if it is less than 484 bytes in length ● Merge the CHNGDUMP options with the options requested in the SDUMP parameter list according to the CHNGDUMP mode 			<p>If the caller supplied a DCB, IEAVAD00 calls the SDUMP parameter list analysis routine (IEAVTSPR subroutine VALDCB) to validate the user's DCB. The DCB must represent an open data set on tape or DASD. If the caller specified a DASD data set, the data set must be on a supported device with a track size of at least 4104 bytes.</p> <p>If the caller supplied a subpool list (SUBPLST keyword specified), IEAVAD00 calls the SDUMP parameter list analysis routine (IEAVTSPR subroutine SDSUBPL) to sort the list into global and local subpools. The sorted lists are put in the correct format for later use by VSM, and stored in the RTSD. For SYSMDUMP requests, subpools are processed in a similar fashion and stored into the work area passed by ABDUMP as part of the SYSMDUMP interface.</p> <p>Callers with SDUMP parameter list errors are given ABEND X'233'. IEAVAD00 returns a reason code in register 15, which describes the parameter in error. See <i>Message Library: System Codes</i> for a list of reason codes.</p>		

IEAVAD00 - SVC Dump Processing (Part 5 of 12)



IEAVAD00 - SVC Dump Processing (Part 6 of 12)

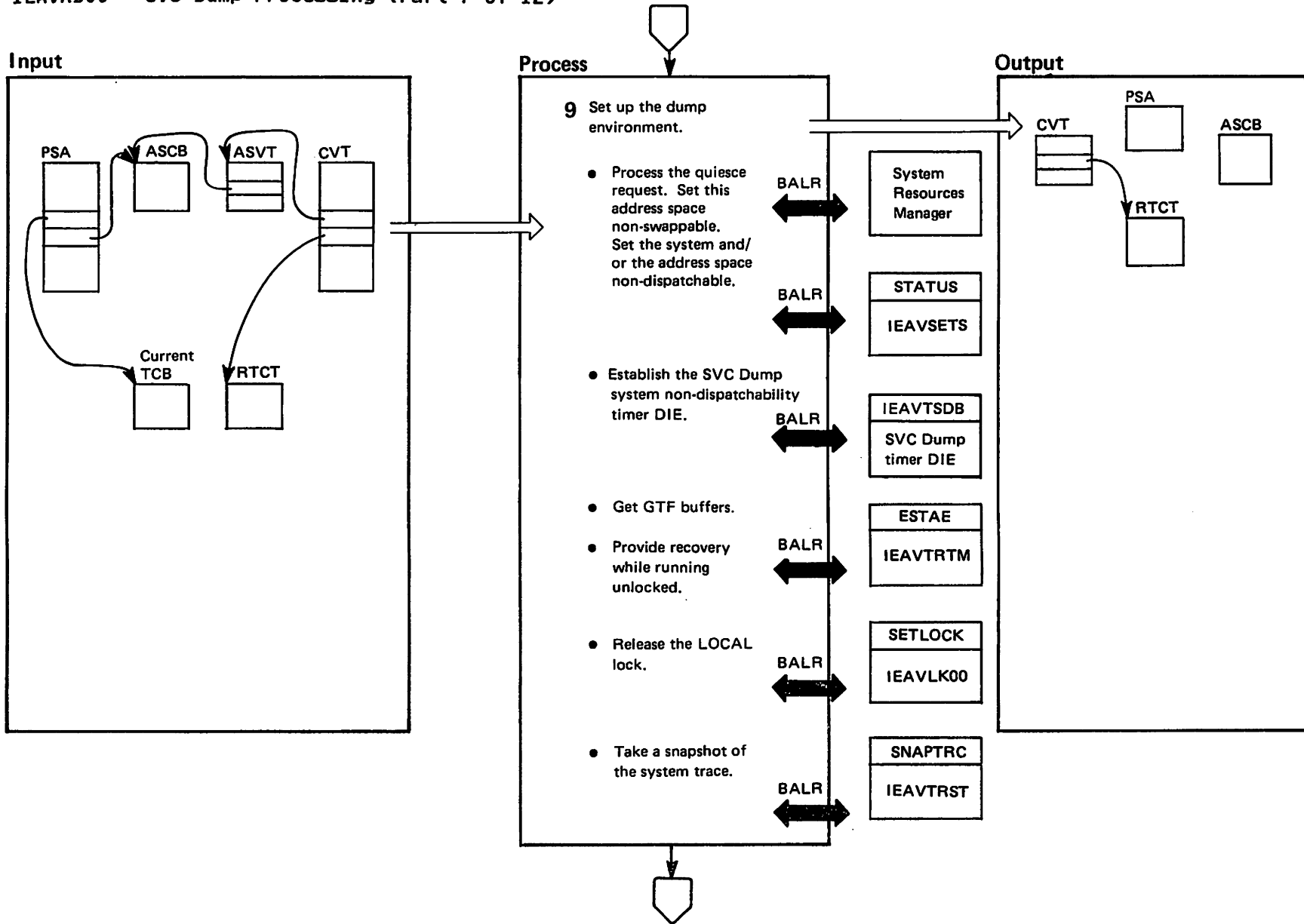
Extended Description	Module	Label
----------------------	--------	-------

7 IEAVAD00 initializes the pre-dump parameter list (DSPD). IEAVAD00 passes the DSPD to ADYPRED, the DAE pre-dump processor. ADYPRED determines if this dump should be suppressed and builds the DAE portion of the dump header record. When ADYPRED returns control, IEAVAD00 checks the return code in register 15 to determine if the dump should be suppressed.

8 If the dump should be suppressed, IEAVAD00 saves the dump suppression status flags and the dump suppression reason code (X'0B' in SDWASDRC) in the caller's SDWA. IEAVAD00 then calls IEAVTSD1 to begin post-dump exit processing. IEAVTSD1 obtains storage and copies the dump header record into this storage. IEAVTSD1 posts IEAVTDSV and passes a copy of the dump header record. IEAVTDSV begins post-dump exit processing. IEAVTSD1 then returns to IEAVAD00. IEAVAD00 unlocks SVC dump and sets a no dump return code (X'08' for an SVC dump, X'0C' for a SYSMDUMP) in register 15. IEAVAD00 returns to the caller.

If the dump is to be taken, processing continues with step 9.

IEAVAD00 - SVC Dump Processing (Part 7 of 12)



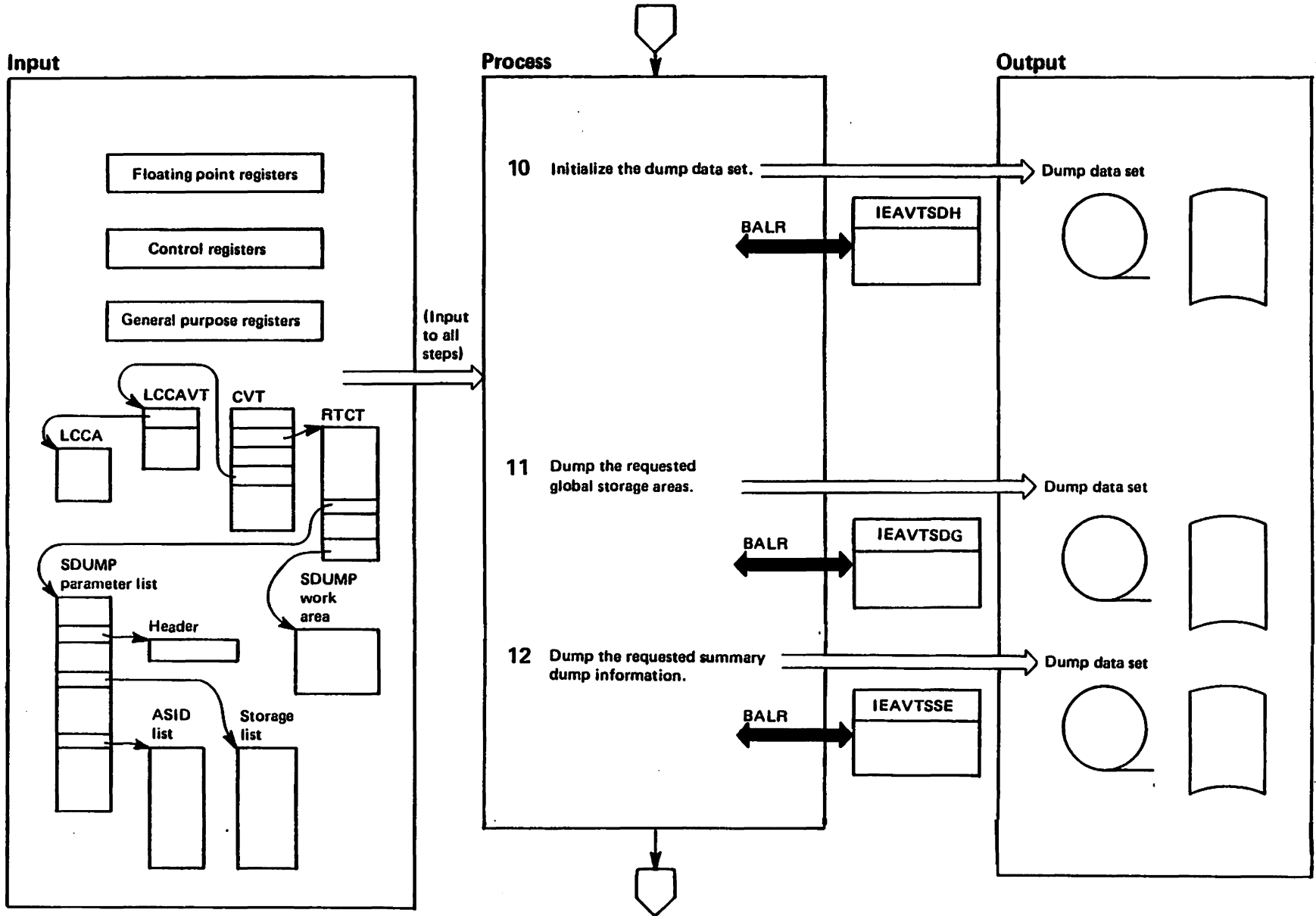
IEAVAD00 - SVC Dump Processing (Part 8 of 12)

Module

9 If the QUIESCE=YES and CSA or SQA operands were requested, SVC Dump informs the system resource manager (SRM) not to swap the current address space and branch enters the STATUS routine to set the system non-dispatchable. IEAVAD00 then calls IEAVTSDB to establish the system non-dispatchability timer DIE to protect the system and tasks of this address space from being left set non-dispatchable due to an unrecoverable error in SVC Dump. The STATUS routine is also branch entered to set all of the tasks in the address space, except the region control task, communications tasks, and the current task, non-dispatchable. If GTF is active and the TRT option was requested, GTF is invoked to save the current GTF buffers for the dump.

Note: GTF buffers are not dumped for SYSMDUMPs. An ESTAE exit is initialized to provide recovery for SVC dump processing, which runs unlocked. The current FRR entry is deleted and the local lock released. If an RTM2 trace table is not available, the SNAPTRC macro is issued. This causes the system trace routine (IEAVTRST) to save a copy of the system trace tables in a buffer that resides in private storage in the trace address space.

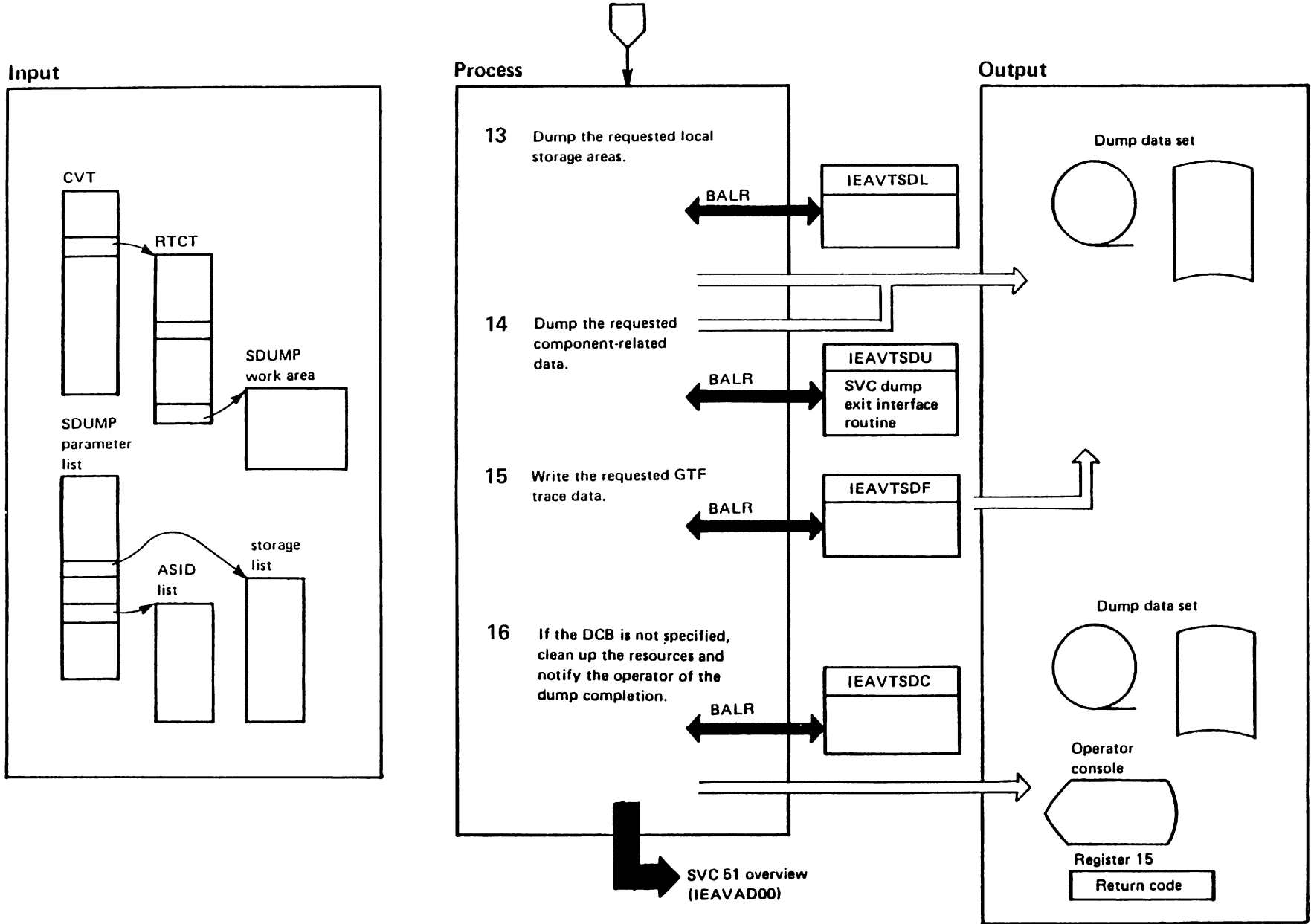
IEAVAD00 - SVC Dump Processing (Part 9 of 12)



IEAVAD00 - SVC Dump Processing (Part 10 of 12)

Extended Description	Module	Label	Extended Description	Module	Label
<p>10 SVC dump chooses an available dump data set. If all dump data sets are full, IEAVAD00 sets an indicator in the RTCT so that IEAVTSDC will issue message IEA994A to the operator. IEAVAD00 returns control to the caller with a return code of 8 in register 15. IEAVTSDC only issues this message the first time SVC dump finds all dump data sets full. It is not issued again until after SVC dump has found an available dump data set and then used all of the available dump data sets. This dump data set checking and message processing is not performed for SYSDUMP requests.</p> <p>If the SDUMP 4K buffer is requested and locked, SVC dump writes it on the dump data set. If IEAVTSDH is running on a processor with an active MSSF and the processor is not a part of a complex with a VM machine, IEAVTSDH uses SVC 122 to obtain the console loop trace data. If the console loop trace data has not been read before, IEAVTSDH calls IEAVTSDP to dump the console loop trace data on the dump data set. SVC dump writes the header record, including the TOD clock value, all of the register values, and the address of the caller, on the dump data set.</p> <p>IEAVTSDH builds an 8K address range table in the SDUMP workarea pointed to by CVTSDBF. This table is used by the rest of SVC dump processing to describe the storage areas to be dumped. For SYSDUMP requests, a storage area acquired by ABDUMP is used instead of the SDUMP workarea. IEAVTSDH adds addresses of the following storage areas to the address range table: ASVT, ASCB for all requested address spaces, CVT, PSA of the current processor, and SCVT, GDA, PCCAVT, all PCCAs and UCBs, IOCOM and other IOS data areas, NUCMAP, TOB, and TRVT, LPA directory, ASTEs, three pages around the error PSW if the PSW points to global storage, and the address range table.</p>	IEAVTSDH	SDHDR	<p>11 IEAVTSDG adds the address ranges for the global storage areas requested in the RTCT and the SDUMP parameter list to the address range table. IEAVTSDG dumps the following areas in the order listed:</p> <ol style="list-style-type: none"> 1. SQA 2. CSA 3. Trace tables 4. PSA's 5. Storage from requested global subpools 6. Global storage requested in the caller's storage list 7. Nucleus 8. Storage dumped by any requested global dump exits (e.g. GRSQ). <p>IEAVTSDG calls module IEAVTSDO to move the areas referenced in the address range table to the dump data set. IEAVTSDG sets the system dispatchable, if it was set non-dispatchable, by a branch entry call to STATUS.</p> <p>12 If a summary dump was requested, IEAVTSSE calls module IEAVTSDO at entry point IEAVTSDP to move the following areas to the dump data set:</p> <ul style="list-style-type: none"> • Storage addresses specified with the SUMLIST or SUMLSTA parameter on the SDUMP macro. • Any RTM2 work areas (RTM2WA) associated with the failing task, 4K of storage before and after every valid, unique address in the program status word, and registers for the failure that were saved in each RTM2WA. 	IEAVTSDG	SDGLOBAL
	IEAVTSDH	SDFULL		IEAVTSDO	SDOUTPUT
	IEAVTSDP			IEAVTSDG	SETSDISP
	IEAVTSDH	SDHGOBL		IEAVTSSE	OUTPUTP
				IEAVTSDO	IEAVTSDP
				IEAVTSSE	SUMLISTP
					RTM2P

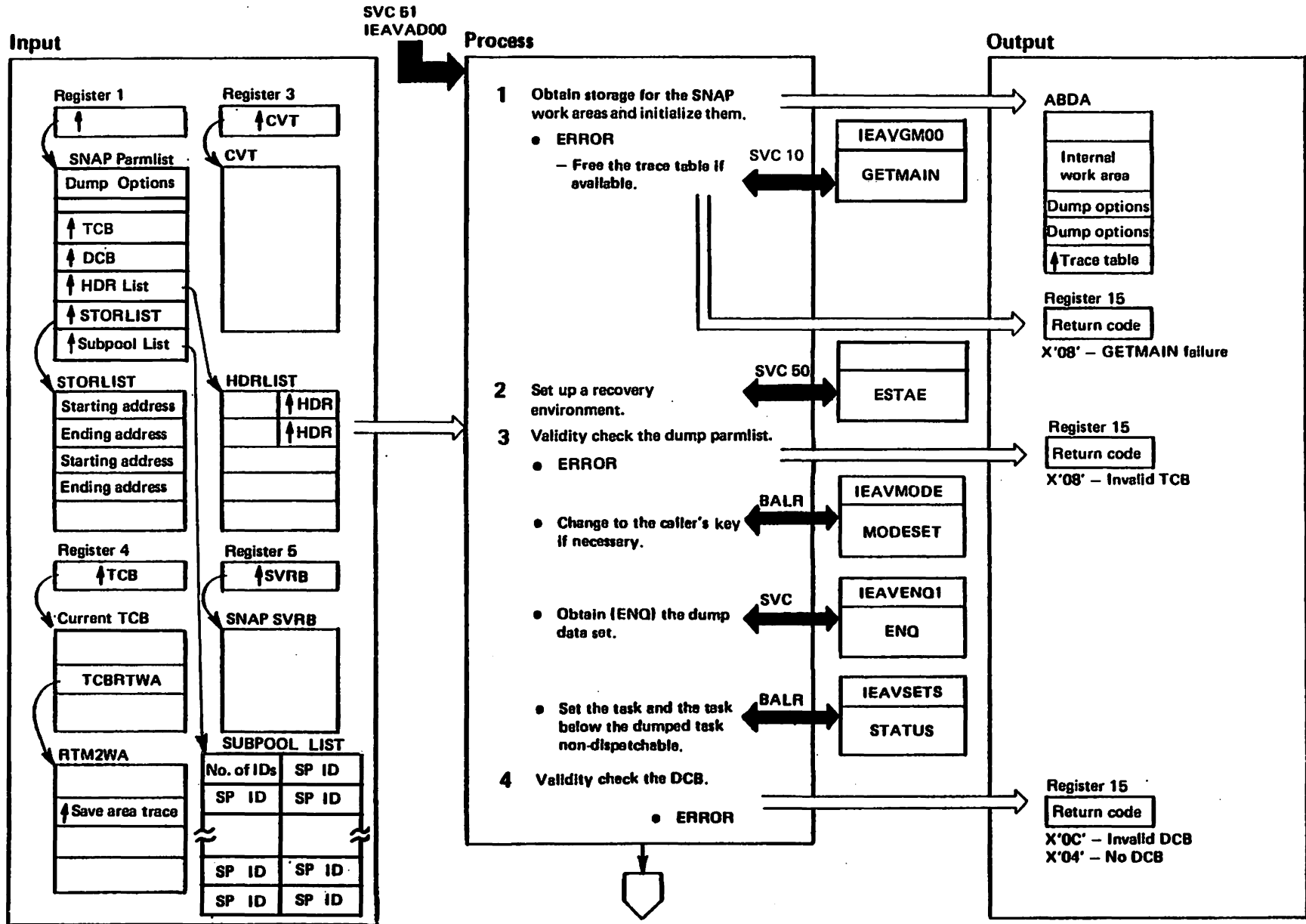
IEAVAD00 - SVC Dump Processing (Part 11 of 12)



IEAVAD00 - SVC Dump Processing (Part 12 of 12)

Extended Description	Module	Label	Extended Description	Module	Label
<p>13 IEAVTSDL adds the address ranges for the local storage areas requested in the RTCT and the SDUMP parameter list to the address range table. IEAVTSDL dumps the following areas in the order listed:</p> <ol style="list-style-type: none"> 1. Cross memory control tables: LT, ET, AT, and PCLINK stacks. 2. Three pages around the error PSW if the PSW points to local storage. 3. Local storage requested in the caller's storage list 4. Storage from requested local subpools. 5. LSQA 6. SWA 7. RGN 8. LPA 9. Storage dumped by any requested local exits. <p>IEAVTSDL calls IEAVTSDO to move the local areas referenced in the address range table to the dump data set.</p>	IEAVTSDL	SDLOCAL	<p>issues this message to the operator. (IEAVTSDC does not issue this message if any tape dump data sets are used. If tape dump data sets are being used, IEAVTSDC issues message IEA984E to inform the operator to mount a new tape.) IEAVTSDC calls IEAVTSD1 to begin post-dump exit processing. IEAVTSD1 obtains storage and copies the dump header record into this storage. IEAVTSD1 posts IEAVTDSV and passes a copy of the dump header record. IEAVTDSV begins post-dump exit processing. IEAVTSD1 then returns to IEAVTSDC. If the system was set non-dispatchable, IEAVTSDC sets it dispatchable. IEAVTSDC removes any SVC dump serialization. Upon completion of this cleanup process, SVC dump returns to the caller with a return code in register 15 showing the success or failure of the dump operation.</p>	IEAVTSD1	
	IEAVTSDO	SDOUTPUT			FREEFLGS
<p>14 IEAVAD00 calls module IEAVTSDU to determine if any component-related data is to be included in the dump. If the trace table option (TRT) is specified on the SDUMP macro, IEAVTSDU loads and calls the master trace dump exit (IEEMB879). Otherwise, processing continues at step 15. IEEMB879 determines if the master trace is active and, if it is, invokes the SDUMP output service to write the master trace data on the dump data set. If the master trace is not active, IEEMB879 returns to IEAVTSDU.</p>	IEAVTSDU				
	IEEMB879				
<p>15 If GTF is active, and TRT has been requested, IEAVTSDF accesses the trace records gathered by GTF using the GTFREAD service. As each GTF record is read, IEAVTSDF calls IEAVTSDP to write the record on the dump data set.</p>	IEAVTSDF	SDGTF			
<p>16 SVC dump writes an end-of-file mark on the dump data set after the last dump record has been written. If the DCB parameter was not specified, IEAVTSDC issues message IEA911E to the operator showing the unit address and data set name of the data set on which the SVC dump is located. If the indicator in the RTCT is set to issue message IEA994A, IEAVTSDC</p>	IEAVTSDC	SDENDUP			
					ENDDS
					WRITEMSG

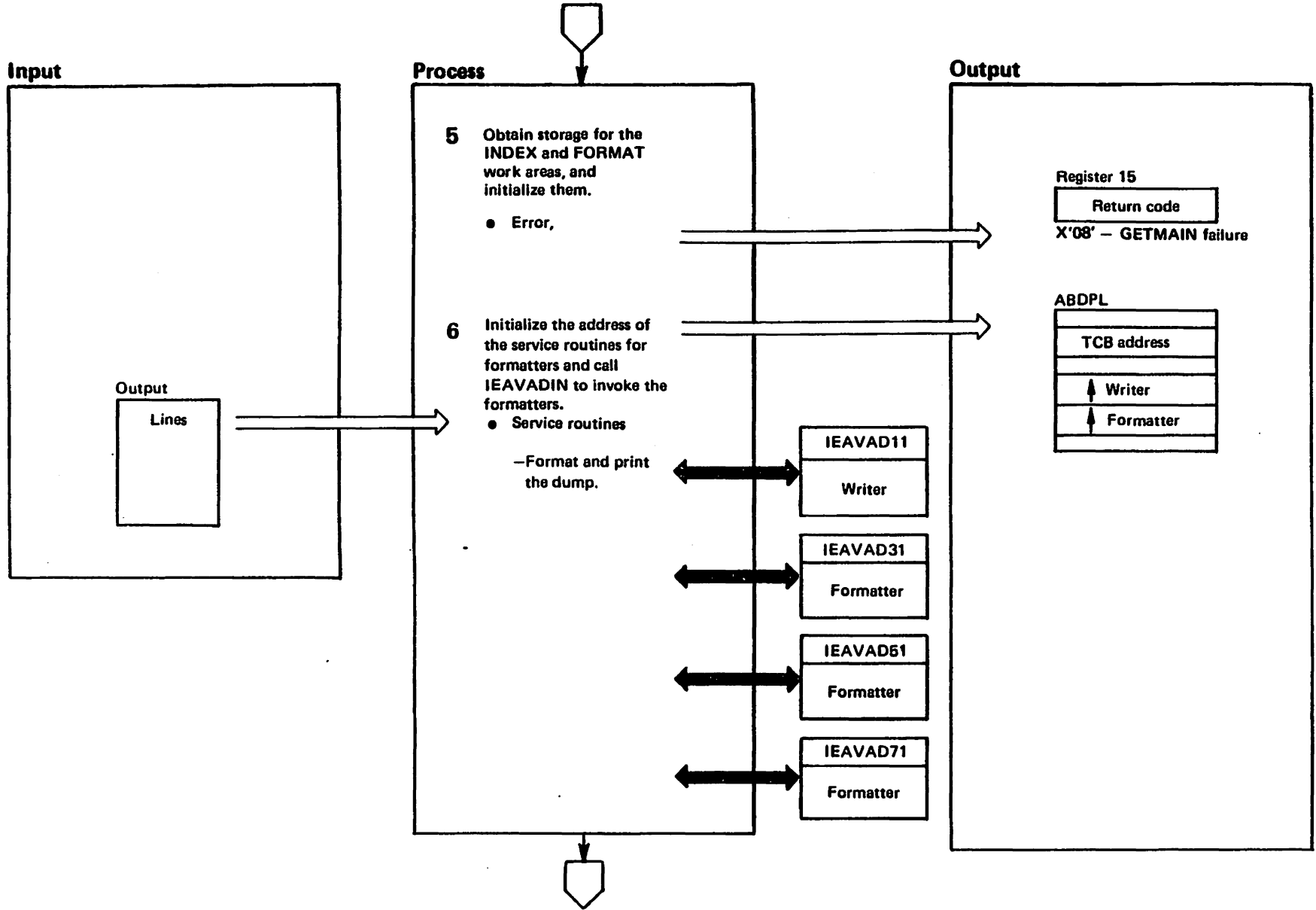
IEAVAD01 - Snap Dump Processing (Part 1 of 14)



IEAVAD01 -- Snap Dump Processing (Part 2 of 14)

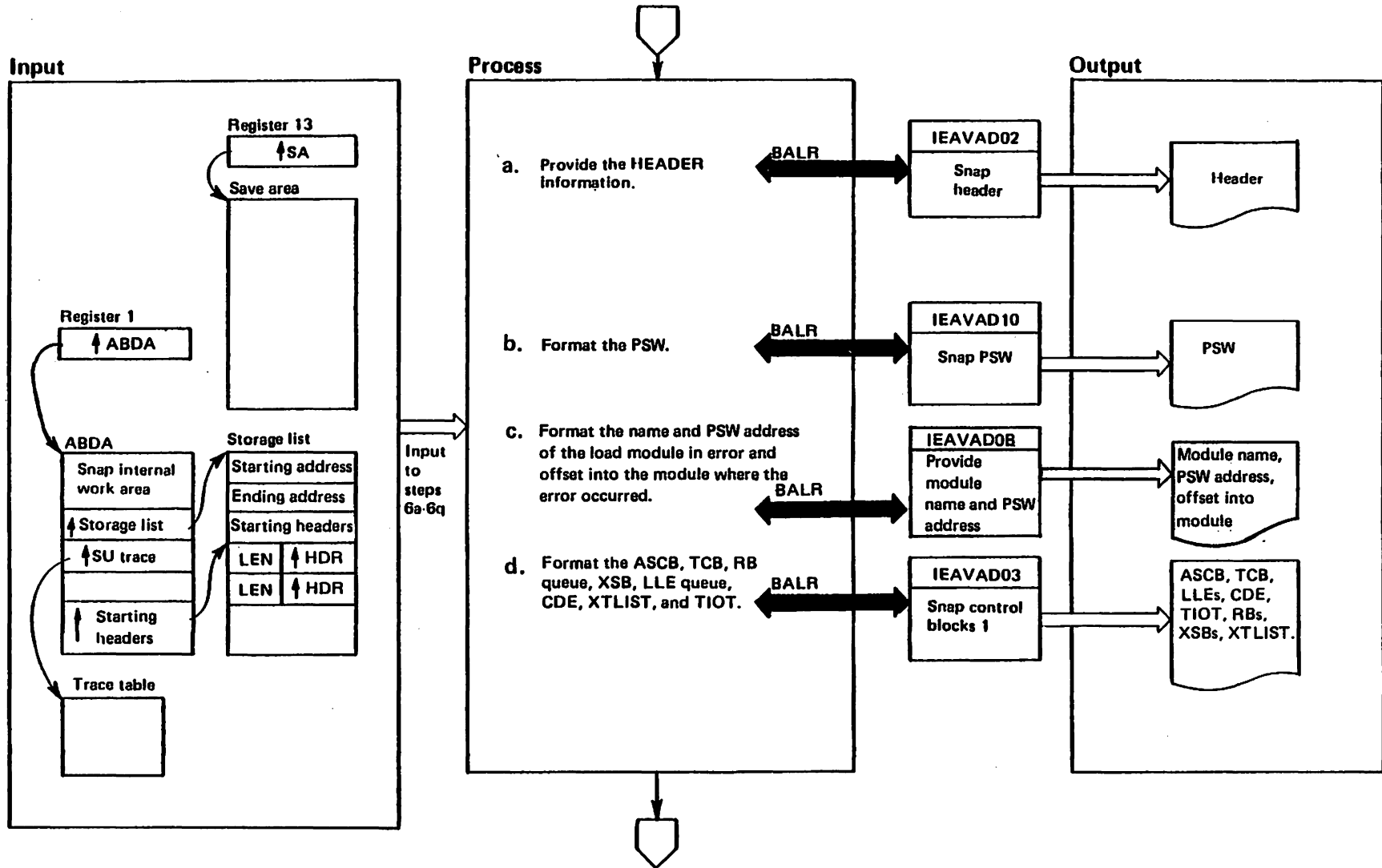
Extended Description	Module	Label
<p>The SNAP dump routines produce a formatted dump of various areas of storage, depending on the parameters. SNAP receives control from IEAVAD00 when the SNAP macro is issued.</p> <p>The main SNAP module, IEAVAD01, does initialization and then routes control to IEAVADIN. IEAVADIN then invokes various formatting routines based on the SNAP options. These formatting routines format the dump.</p>		
<p>1 The SNAP routine obtains storage for an ABDAREA. This area contains save areas, output buffers, and work areas for the format routines. If this area can not be obtained, SNAP returns a return code of X'08' in register 15 and IEAVTABD frees the copied trace table, if it was obtained.</p>	IEAVAD01	SNPINIT
<p>2 SNAP establishes an ESTAE to provide recovery for the dump process. The recovery routine (SNPRCVR) re-receives control when an error occurs. If this routine receives control, SNPRCVR issues an informational message 'ERROR FORMATTING XX'. If an error occurs in SNAP mainline, SNPRCVR performs an SDUMP and RECORD. If the error occurred in a format routine, SNAP routes control to the next format routine. Format routines, which in some cases expect a program check, cover their modules with an internal retry address.</p>	IEAVAD01	SNPINIT

Extended Description	Module	Label
<p>3 SNAP processing does not validity check the parameter list for calls from ABDUMP, nor does SNAP enqueue upon the dump data set for calls from ABDUMP; ABDUMP has already performed the ENQ process. If SNAP encounters a bad TCB, it returns a return code of X'08' in register 15. If the current task is not being dumped, SNAP calls STATUS to set the task non-dispatchable.</p>	IEAVAD01	SNPVALCK
<p>4 SNAP checks the validity of the DCB. If the DCB is not available, SNAP puts a return code of X'04' in register 15 and returns to the caller. If the DCB is not valid or not opened, SNAP returns a return code of X'0C' in register 15.</p>		SNPCHK



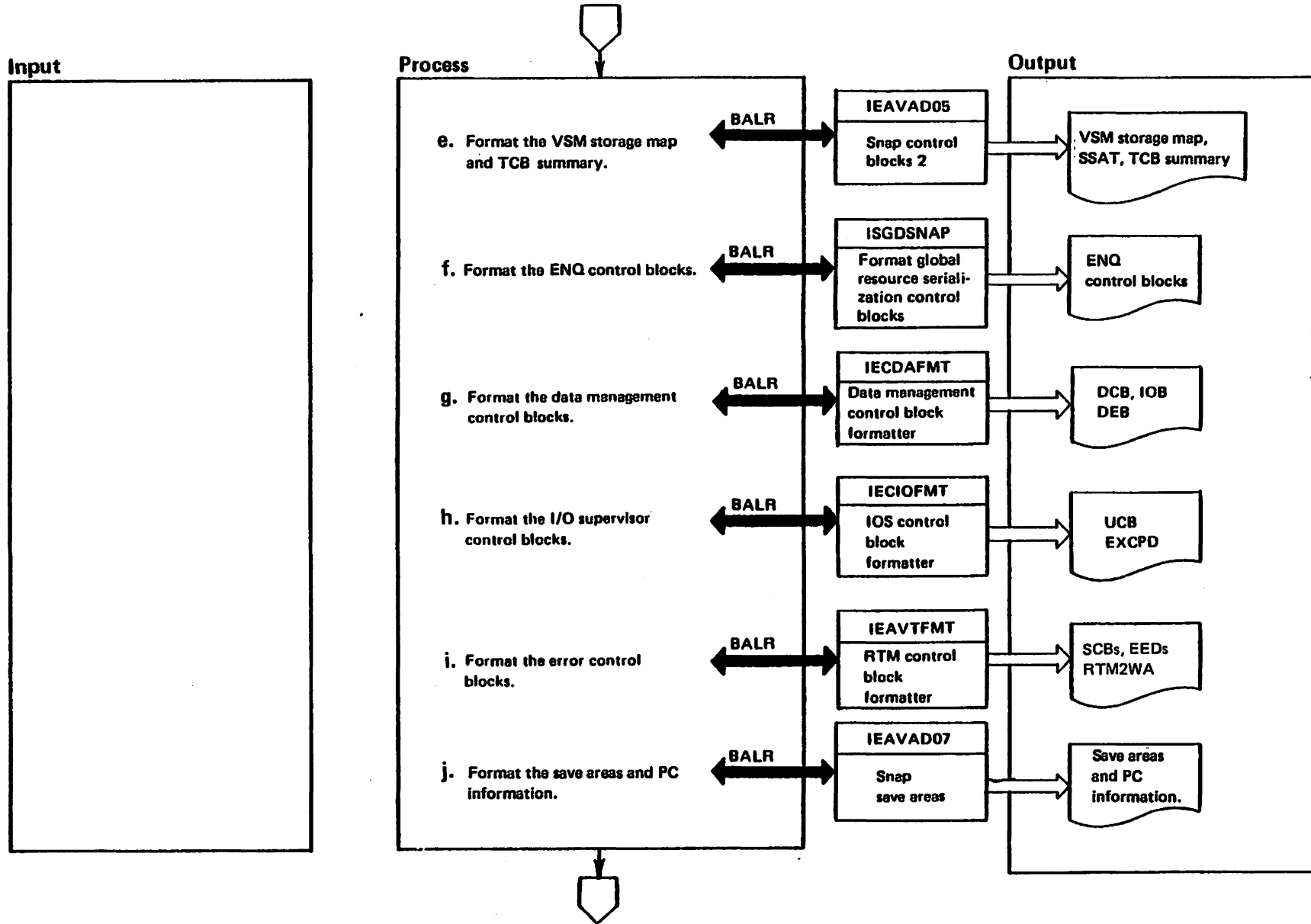
IEAVAD01 - Snap Dump Processing (Part 4 of 14)

Extended Description	Module	Label
5 SNPINDEX obtains storage for the INDEX buffer and a work area to be used by the format routines. SNAP maintains this index during the dump process and prints it at the end of the dump.	IEAVAD01	SNPINDEX
SNPROUT invokes IEAVADIN. IEAVADIN invokes the formatters depending on the dump options specified in the SNAP parmlist (IHASNP). This routine routes control to the formatters with register 1 pointing to the IHAABDA, register 13 pointing to the save area, and register 14 pointing to the return address.	IEAVAD01	SNPROUT
6 All SNAP formatters use the following service routines to convert and print the dump.		
IEAVAD11 - This routine is the output routine and all format routines call it to print the lines of the dump on an output device.	IEAVAD11	
IEAVAD31 - This format routine unpacks and translates data in the print line, providing indentation.	IEAVAD31	
IEAVAD51 - This module translates data in the print line.	IEAVAD51	
IEAVAD71 - This module translates blocks of storage. The snap FORMAT routines use all of the formatting and output routines to produce printable format.	IEAVAD71	



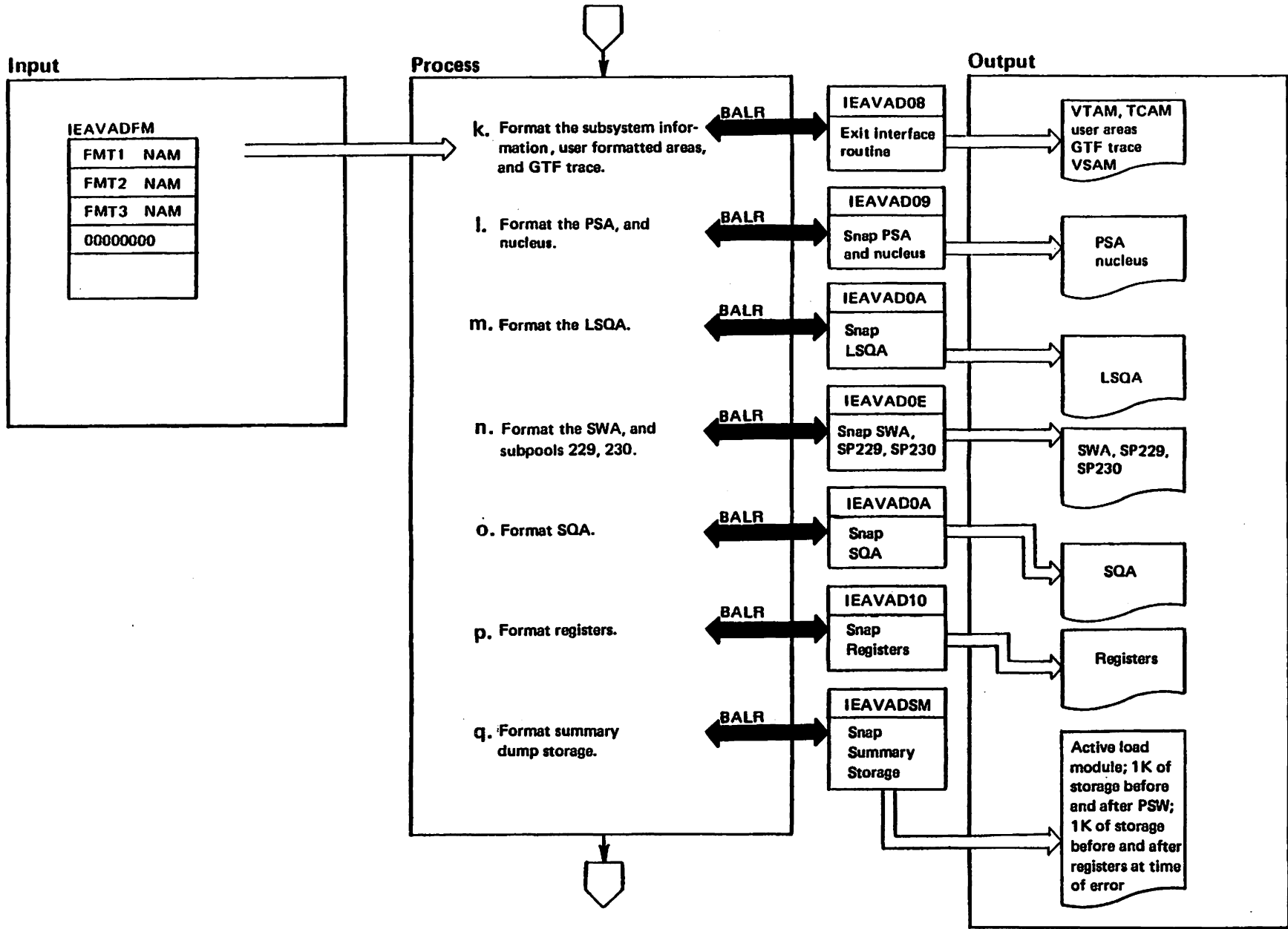
IEAVAD01 - Snap Dump Processing (Part 6 of 14)

Extended Description	Module	Label
a. IEAVAD02 - This format routine displays the jobname, stepname, time, date, DUMPID, completion code, ILC (instruction length count), interruption code, and CPUID (serial number).	IEAVAD02	
b. IEAVAD10 - This format routine formats the PSW either at the time of the abend or at the time the SNAP macro was issued.	IEAVAD10	
c. IEAVAD0B - This format routine provides the name and PSW address of the load module. The offset into the module where the error occurred is provided as the default output.	IEAVAD0B	
d. IEAVAD03 - If the dump options specified included CB or SUM, this format routine displays the ASCB, CDE, TCB, RB queue, LLE queue, XTLIST, and TIOT. IEAVAD03 displays these control blocks by using the service aids exit interface. See "SPL Service Aids" for a description of this interface.	IEAVAD03	



IEAVAD01 - Snap Dump Processing (Part 8 of 14)

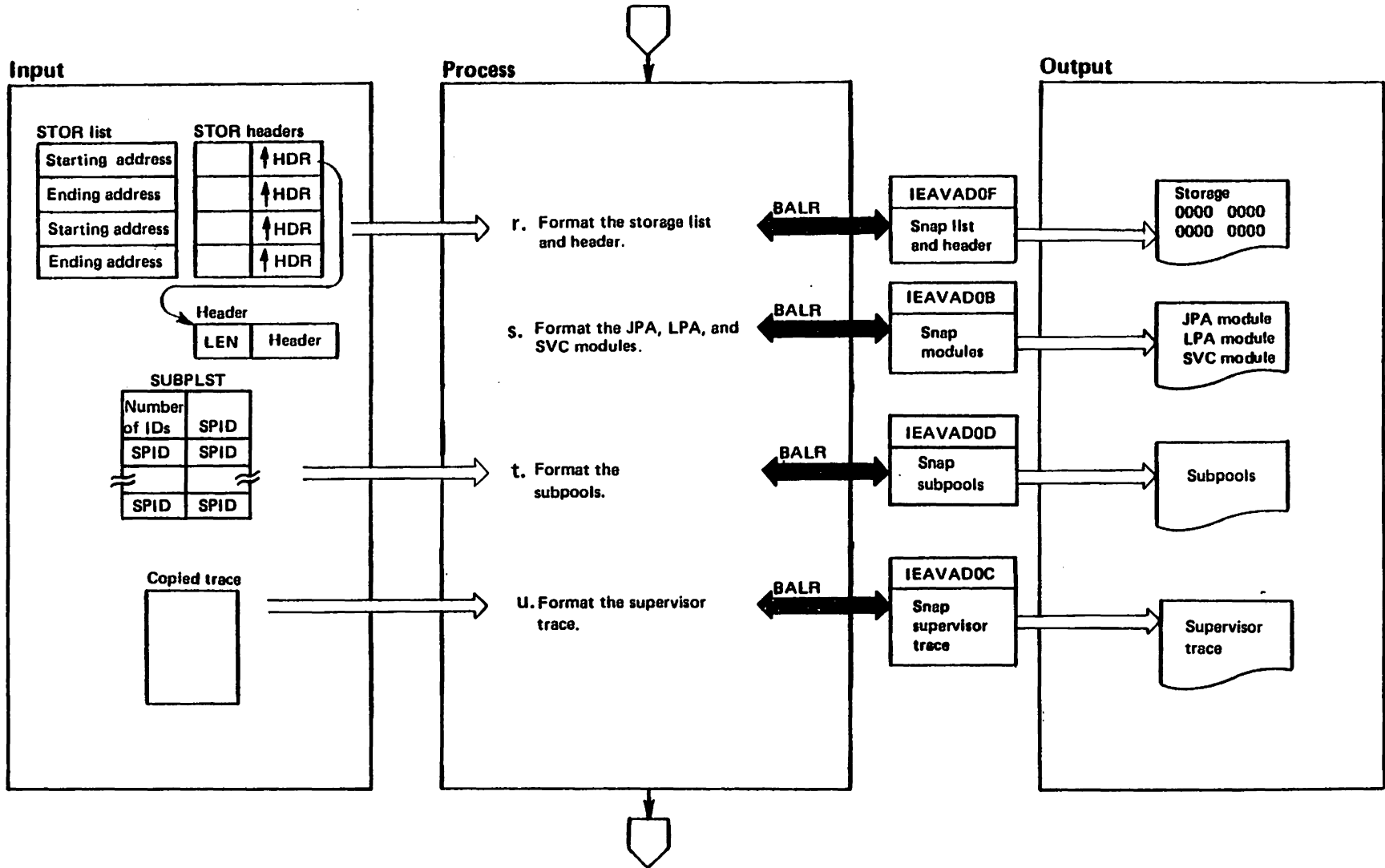
Extended Description	Module	Label
e. IEAVAD05 - If the dump options specified included CB or SUM, this format routine formats the VSM storage map, SSAT and TCB summary. If the dump options specified included PCDATA, this format routine also formats the PC table.	IEAVAD05	
f. ISGDSNAP - If the dump options specified included the Q option, this format routine formats QCBs and QELs.	ISGDSNAP	
g. IECDAFMT - This format routine displays the DCB, IOB and DEB. If the dump options specified included CB, this format routine formats the DEB. If the dump options specified included DM, this format routine formats all of the data management control blocks (DEB, DCB, IOB, LCB's, ICB's).	IECDAFMT	
h. IECIOFMT - If the dump options specified included IO, this format routine displays the UCB and EXCPD related to the task being dumped.	IECIOFMT	
i. IEAVTFMT - If the dump options specified included ERR or SUM, this format routine displays the RTM2WA, the STAE control block and the EEDs related to the failing task.	IEAVTFMT	
j. IEAVAD07 - If the dump options specified included SAH, SA or SUM, this format routine displays the save areas and PCLINK stacks.	IEAVAD07	



IEAVAD01 - Snap Dump Processing (Part 10 of 14)

Extended Description	Module	Label
k. IEAVAD08 - This routine invokes the VTAM (ISTHAFD1), TCAM (IGC0E054), VSAM (IDA0195A), and user (IGC0905A) formatters to format VTAM, TCAM, VSAM, and user control blocks if CB is specified as a dump option. If CB was the specified dump option, this routine tests the user format CSECT (IEAVADFM). If this CSECT contains any entries (MODULE NAMES), this routine loads these routines. SNAP invokes each entry in the list. If the TRT option was specified and GTF is active, SNAP invokes GTF (IGC0F05A) to print the GTF trace table.	IEAVAD08	
l. IEAVAD09 - This routine formats the PSA and nucleus or the PSA and non-page protected nucleus if NUC was a requested dump option. This routine formats the PSA and the complete virtual nucleus if ALLVNUC was a requested dump option.	IEAVAD09	
m. IEAVAD0A - This invocation of the routine formats the local system queue area (LSQA). This routine receives control when LSQA is specified as a dump option.	IEAVAD0A	
n. IEAVAD0E - This format routine formats the scheduler work area (SWA) if requested on the dump options. SNAP prints the SWA addresses from low to high. Subpools 229 and 230 are also formatted when LSQA is a requested dump option.	IEAVAD0E	
o. IEAVAD0A - This invocation of the routine formats the system queue area (SQA). This routine receives control when SQA is specified as a dump option.	IEAVAD0A	
p. IEAVAD10 - This invocation of the routine formats the registers at the time of error. These registers are obtained from the RTM2 work area (RTM2EREG) if SUM was the requested dump option, or from the TCB (TCBGRS) if REGS was the requested dump option.	IEAVAD10	

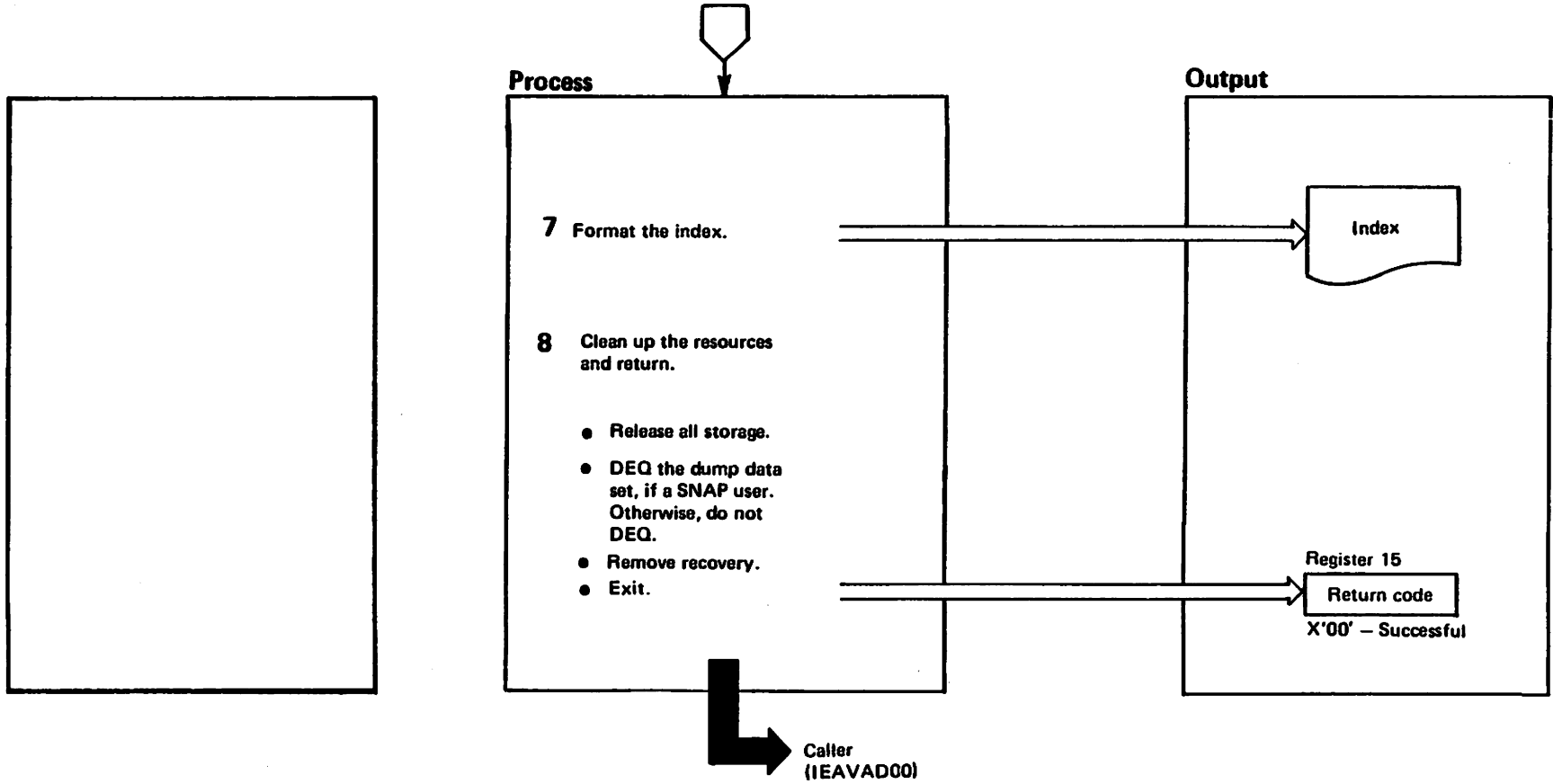
Extended Description	Module	Label
q. IEAVADSM - This routine formats part of the summary dump information for SYSABEND and SYSUDUMP dumps if SUM was a requested dump option. This information includes the active load module, 1K of storage before and after the address contained in the PSW, and 1K of storage before and after the addresses contained in the registers at the time of error. No duplicate data is formatted.	IEAVADSM	



IEAVAD01 - Snap Dump Processing (Part 12 of 14)

Extended Description	Module	Label
r. IEAVAD0F - This routine formats the areas specified by the user storage list. If STORAGE HEADERS were specified, SNAP prints the headers above each storage range.	IEAVAD0F	
s. IEAVAD0B - This invocation of the routine formats the JPA, LPA, and active SVC modules when JPA, LPA, or ALLPA is a specified dump option.	IEAVAD0B	
t. IEAVAD0D - This format routine formats the user subpools (0-127 and 252). This routine is invoked if SPLS is a specified dump option. SNAP sorts all subpool data in ascending order by subpool number and prints the subpool data for the SPLS option. This routine is also called for SUBPLST option to display storage for specific subpools requested by the user. SNAP sorts the subpool data beginning with the lowest address within individual subpools, and prints the subpool data for the SUBPLST option. The SPLS option and the SUBPLST option are mutually exclusive.	IEAVAD0D	
u. IEAVAD0C - This routine builds the exit parameter list (IHAABDPL) and invokes the trace formatting routine (IEAVETFC). IEAVETFC formats and displays the trace table. IEAVAD0C receives control if TRT and/or SUM are requested dump options. When TRT is specified for an authorized user, IEAVAD0C displays the entire trace table. When TRT is specified for an unauthorized user, IEAVAD0C displays only the trace entries for the current address space. If SUM is specified, whether the user is authorized or unauthorized, IEAVAD0C displays only the trace entries for the current address space.	IEAVAD0C	

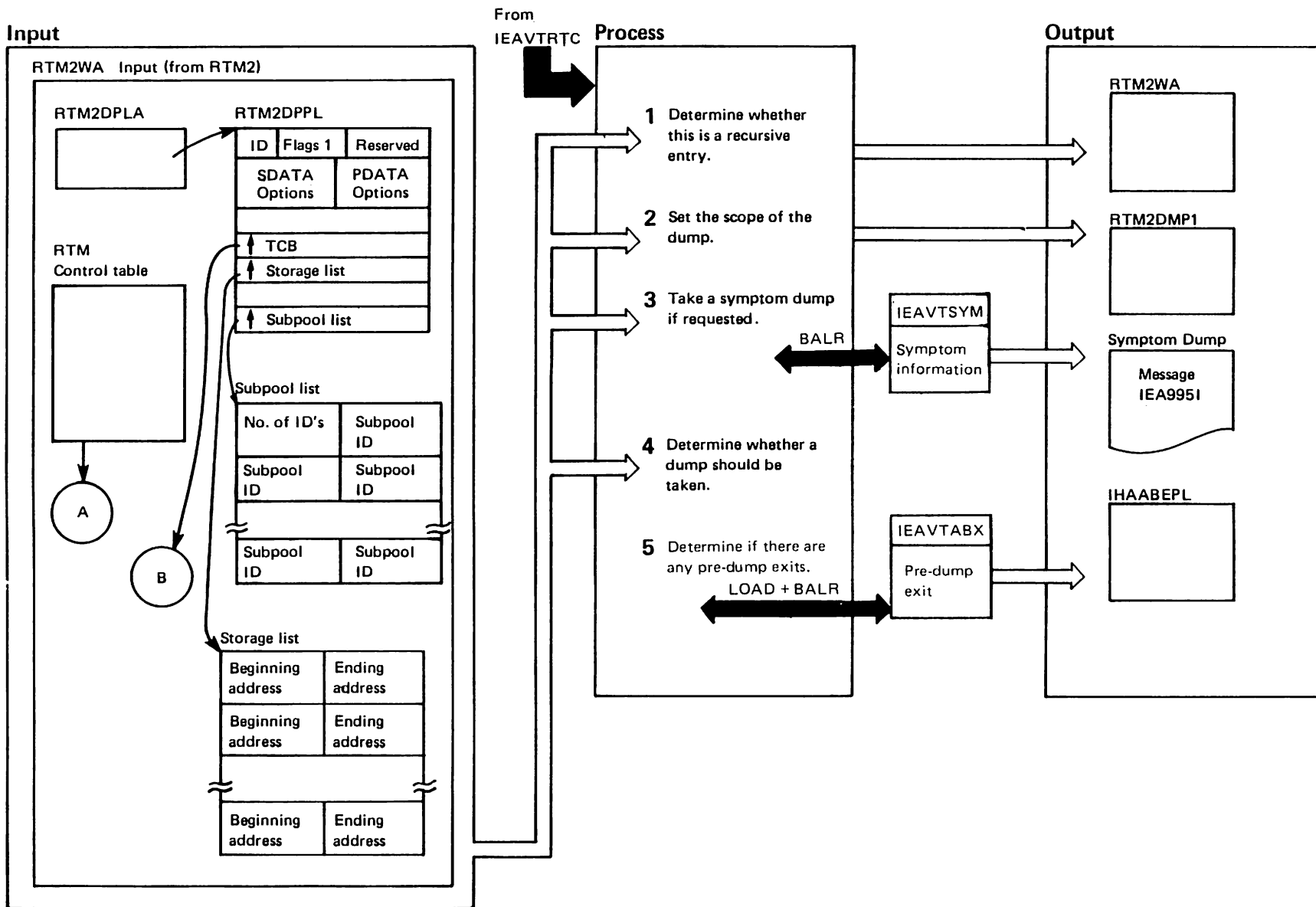
IEAVAD01 - Snap Dump Processing (Part 13 of 14)



IEAVAD01 - Snap Dump Processing (Part 14 of 14)

Extended Description	Module	Label
<p>7 SNAP displays the dump index at the end of all dumps. IEAVAD01 updates and maintains the index as the dump is processed. This index maintains all key areas' page numbers. IEAVAD01 displays the active modules in alphabetical order.</p>	IEAVAD01	SNPINDX
<p>8 SNAP cleans up all the resources it obtained and returns to the caller. SNAP frees the storage it obtained via a FREEMAIN, and dequeues the dump data set if the caller is not ABDUMP. SNAP deletes the ESTAE and returns to the caller.</p>	IEAVAD01	SNPCLNUP

IEAVTABD - ABDUMP Processing (Part 1 of 10)



IEAVTABD - ABDUMP Processing (Part 2 of 10)

Extended Description	Module	Label	Extended Description	Module	Label
<p>Terminating tasks can request a storage display. RTM2 provides the dump via ABDUMP processing. The RTM2WA contains the dump options for the terminating task; ABDUMP processing checks these options, prepares the dump data set, constructs a SNAP parameter list (for the actual dump), and gives control to SNAP processing (see the M.O. diagram SNAP Dump Processing (IEAVAD01) for the description of SNAP's operation).</p> <p>1 IEAVTABD protects itself from recursions by setting indicators to denote external functions in control, and to denote the completion of external functions.</p> <p>IEAVTABD turns the in-control indicators off when control returns from the external function.</p> <p>When an error occurs, IEAVTABD checks the first set of indicators (the in-control set) to determine if an external function had control. If an external function had control, control goes to a clean up subroutine, ADRCLN, to perform the necessary cleanup.</p>			<p>If an external function did not have control, IEAVTABD determines the last completed function, initializes the new RTM2WA with information from the previous work area, and passes control to the appropriate routine in IEAVTABD to continue processing.</p> <p>This subroutine performs the necessary cleanup, requests an SVC dump, and gives control back to the caller.</p>		
	IEAVTABD	ADRECOV	<p>2 The scope of a dump can either be a single task if RETRY with dump has been requested from an ESTAE exit, or if the task has a subtask that is a job-step task; a failing task tree, if no recovery from the ABEND was accomplished, or a jobstep tree if no recovery was accomplished and the ABEND is a step ABEND. If it is a RETRY with dump (RTM2DREQ=1 and RTM2RETR=1) situation or if the task has a subtask that is a jobstep, IEAVTRTC sets the RTM2DMP1 flag to 1 and places the current TCB address in the TCB field of the dump parameter list in the RTM2WA (SNPTCBA). If it is not a step ABEND from a subtask of a step (RTM2STPT=0), IEAVTRTC places the current TCB address in the SNPTCBA, otherwise IEAVTRTC places the address of the jobstep TCB in the field.</p>	IEAVTRTC	RTCADINT

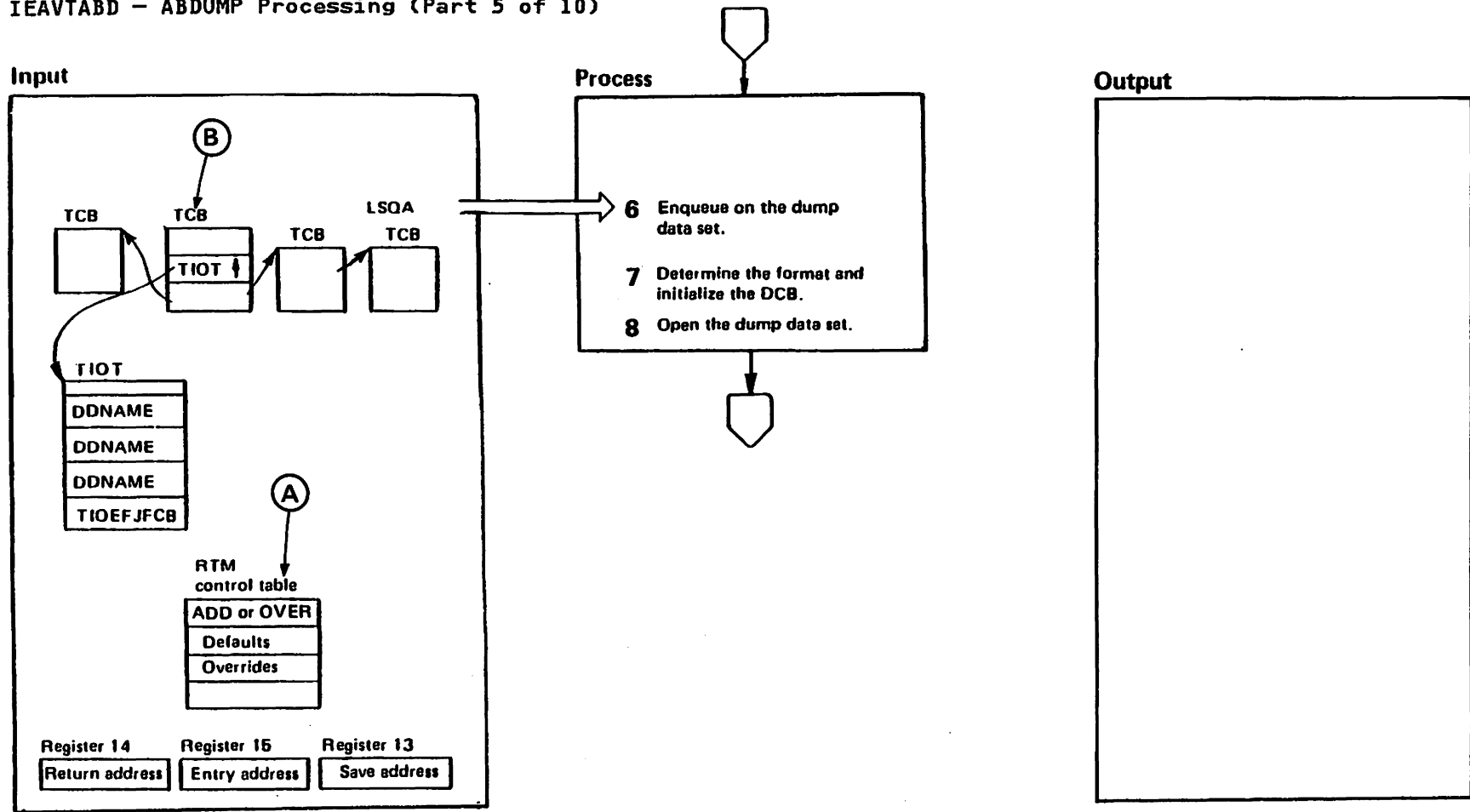
IEAVTABD -- ABDUMP Processing (Part 3 of 10)

No diagram.
Extended description continued on next page.

IEAVTABD - ABDUMP Processing (Part 4 of 10)

Extended Description	Module	Label	Extended Description	Module	Label
<p>3 IEAVTABD checks the RTCT (recovery termination control table) bits (RTCTISAB, RTCTISYU, and RTCTISYM) for a no-dump condition. If all the bits are one, no dumps (SYSABEND, SYSUDUMP, or SYSMDUMP) are wanted. The CHNGDUMP operator command sets these flags and indicates to IEAVTABD that no dump is wanted. If the dumps are not being ignored, ABDUMP issues the RACROUTE macro to determine if the user is authorized to receive the dump. If the return code is 0 or 4, the dump will be taken. Otherwise, the user is not authorized to receive the dump and message IEA8481 results. If the user is authorized to receive the dump, the RTM2WA bits (RTM2NODP, RTM2NOSA, RTM2NOSM, and RTM2NOSU) are checked for no dump. If all of these bits are on ('1'B), no dumps (SYSABEND, SYSUDUMP, SYSMDUMP) are wanted and message IEA8481 results. These bits are set using the SLIP service and indicate to ABDUMP that no dump is wanted. If any or all of these RTM2WA bits are off ('0'B), the TIOT is scanned for a DD entry for SYSABEND, SYSUDUMP, or SYSMDUMP. If one the the DDNAMEs is found and dumps are being taken for that DD type, the NOSYM bit (either RTCTABNS, RTCTSUNS or RTCTSYMN) for that DD type is checked. If the NOSYM bit is on, no SYMPTOM dump is taken. If none of the DDNAMEs is found, the NOSYM bit for the SYSUDUMP DD type (RTCTSUNS) is checked. If the NOSYM bit is on, control returns immediately to RTM2 with a return code of 0. If a SYMPTOM dump is taken, message IEA995I is issued using MLWTO, routing code 11. This message displays the PSW, registers, ERRORID, abend code, and other information about the error.</p>	IEAVTABD		<p>5 If a dump is to be taken, module IEAVTABX is loaded to determine if there are any pre-dump exits to be given control. IEAVTABX is checked for the presence of any exit load module names. If a name is found, the pre-dump exit parameter list (IHAABEPL) is built, the exit routine is loaded, and control is given to the exit routine passing the IHAABEPL in register 1. Upon completion of all exits, the exits are deleted and the return code is checked to determine if the exit modified the dump options (RC=4), suppressed the dump (SLIP did not indicate no suppression, RC=8) or kept the original options (RC=0). IEAVTABX is deleted. If the dump options were changed (RC=4), the new SNAP parameter list is placed in the RTM2WA. If the dump was suppressed (RC=8), and SLIP ACTION=NOSUPPRESSION was not specified, message IEA8481 is issued and control is returned to RTM2 with a 0 return code in the RTM2WA.</p>		
<p>4 If one of the DDNAMEs is found on the TIOT, the no-dump bit in the RTCT for the type of dump requested is checked. If the no-dump flag is on for the DD found, a further scan will be made of the TIOT to determine if any other dump DD is found. If no dump is to be taken, control returns to RTM2 with a 0 return code in the RTM2WA.</p>	IEAVTABD				

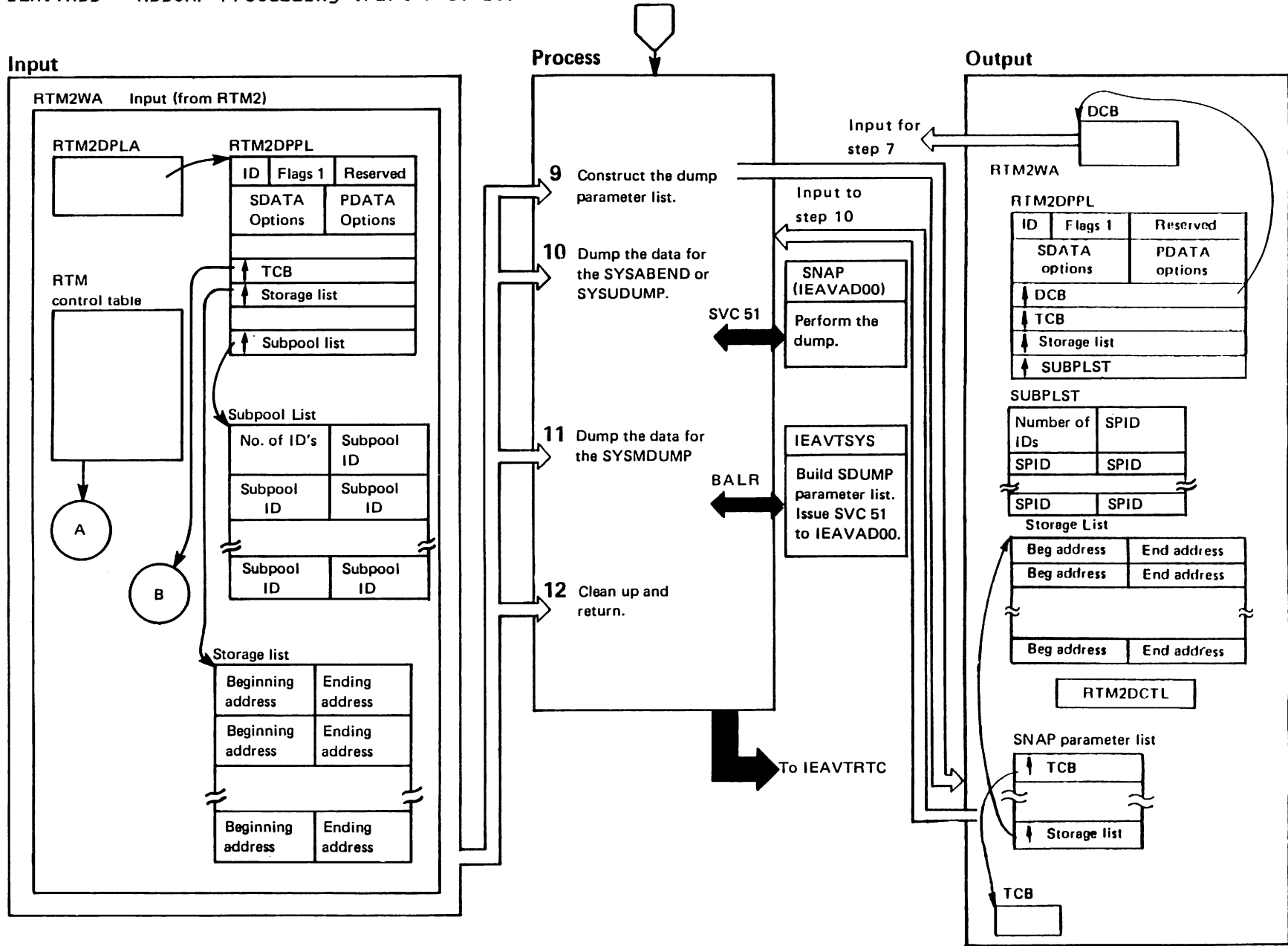
IEAVTABD - ABDUMP Processing (Part 5 of 10)



IEAVTABD - ABDUMP Processing (Part 6 of 10)

Extended Description	Module	Label
<p>6 IEAVTABD enqueues on the dump data set using the option RET=HAVE, a major name of SYSIEA01 and a minor name of IEA. On a nonzero return code from ENQ, IEAVTABD issues an SVC dump and returns control to RTM2.</p>		
<p>7 IEAVTABD obtains storage for the DCB and parameter list from subpool 230. If DDNAME=SYSMDUMP, DSN=SYS1.SYSMDPxx (xx can be 00-FF), and DISP=SHR are specified, IEAVTABD initializes the DCB with MACRF=(R,W), DSORG=PS, LRECL=4104, BLKSIZE=4104, RECFM=FB, and EODAD=DSRDEXIT. Otherwise, IEAVTABD initializes the DCB with DSORG=PS, MACRF=W, RECFM=VBA LRECL=125, and DDNAME as defined in the TIOT.</p>		
<p>8 IEAVTABD opens the DCB in TCB key (via MODESET). If unsuccessful (DCBOFOPN=0), IEAVTABD routes message IEA030I 'OPEN FAILED FOR DUMP DATA SET FOR JS)' to the programmer. IEAVTABD returns control to RTM2 with a return code of 4.</p>		

IEAVTABD - ABDUMP Processing (Part 7 of 10)



IEAVTABD - ABDUMP Processing (Part 8 of 10)

Extended Description	Module	Label	Extended Description	Module	Label
<p>9 IEAVTABD determines the dump options for SNAP from the options passed by RTM2, the installation default options (specified in PARMLIB members IEAABD00-SYSABEND, IEADMP00-SYSUDUMP, and IEADMR00-SYSMDUMP) or the installation override options with any installation additions (specified via the CHNGDUMP operator command). When more than one of these groups is available, the order of selection is as follows:</p> <ol style="list-style-type: none"> When OVER is requested, the options specified via the CHNGDUMP operator command override the options specified via the PARMLIB members or passed by RTM2. If the caller uses the CHNGDUMP operator command with the ADD function of CHNGDUMP, IEAVTABD merges these options with options specified in the PARMLIB members (if available) and merges these with RTM2WA passed options, if available. Lacking CHNGDUMP options, IEAVTABD merges the options passed from RTM2WA with options specified in the PARMLIB members (if available). If no options were passed by RTM2, the PARMLIB options define the dump contents. However, if PARMLIB options are not present and no options were passed by RTM2, no dump is provided. 			<p>10 Based on the type of dump requested, two different processes take place:</p> <p>SYSABEND/SYSUDUMP PROCESS</p> <p>After IEAVTABD has dumped the input TCB, it checks to determine whether an ABEND is in progress (RTM2DMP=0). If so, IEAVTABD calls STATUS to prevent the subtasks from terminating during dump processing. IEAVTABD dumps the subtasks of the abending task and the mother task. IEAVTABD sets TCBFS to one to indicate that it has dumped an abending task. On a nonzero return code from SNAP, IEAVTABD issues message IEA912I 'RECOVERY/TERMINATION DUMP FAILED' and returns control to RTM2 with a return code from SNAP stored in RTM2SNCC. Return codes from SNAP are:</p> <ul style="list-style-type: none"> 0 - Successful completion 4 - DCB not opened, undefined page reference on DCB 8 - TCB not valid, undefined page reference on TCB, insufficient storage, invalid parameter list, a subtask is a jobstep TCB 12 - DCB type incorrect 	IEAVSETS	

If an abend 'x22' abend is in progress, or if SUBTASKS was requested, IEAVTABD selects the options for TCBs (other than the input TCB) as follows, to prevent the display of redundant data:

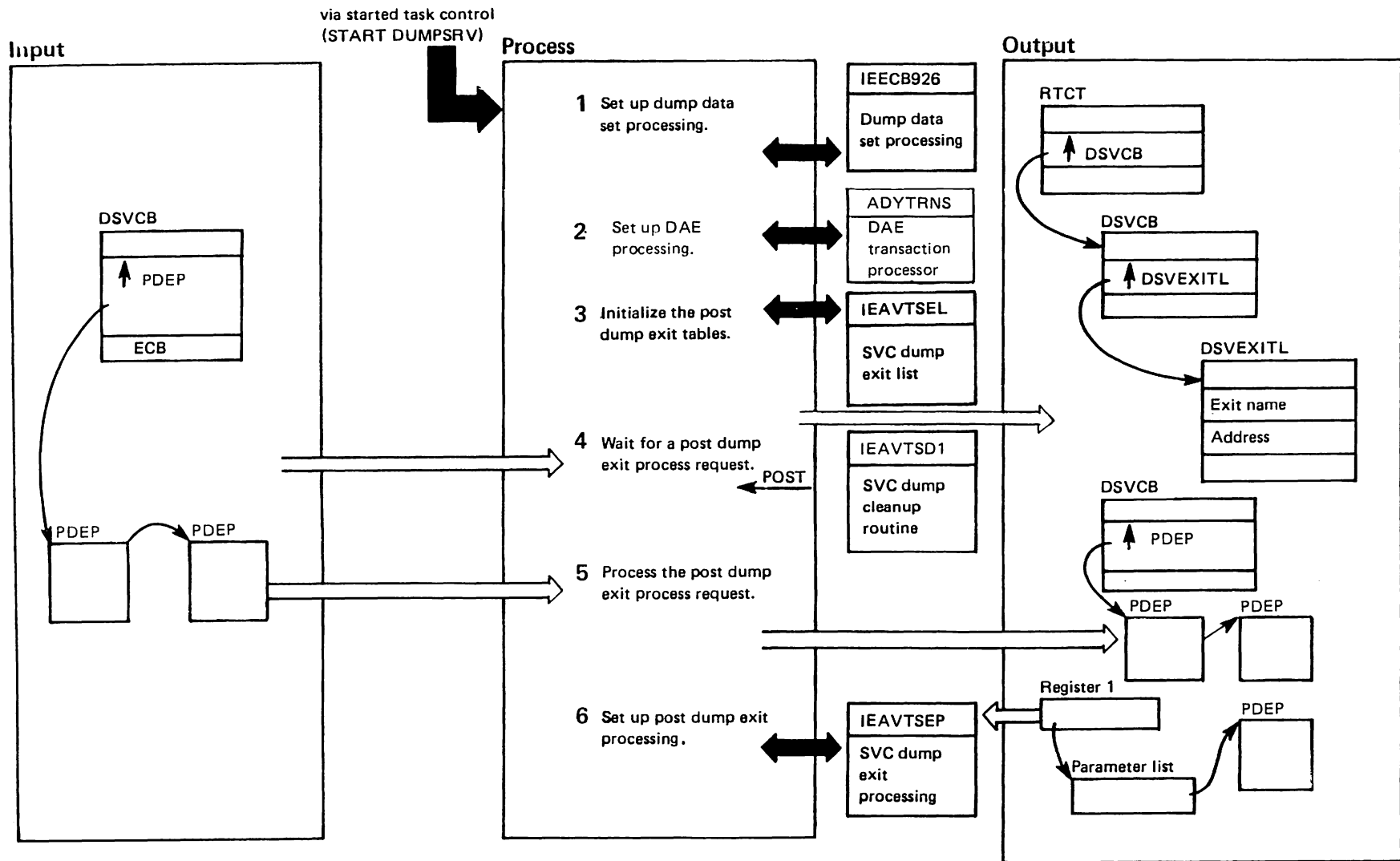
- The SDATA options NUC, SQA, LSQA, SWA, TRACE, QCB, PC, SUM, ALLVN are turned off for each subtask.
- The SDATA options CB, DM, IO, ERR are left on if requested for the first task.

IEAVTABD - ABDUMP Processing (Part 9 of 10)

No diagram.
Extended description continued on next page.

IEAVTABD - ABDUMP Processing (Part 10 of 10)

Extended Description	Module	Label	Extended Description	Module	Label
<p>11 If SYSDUMP DD entry is found, module IEAVTSYS is called to process the SYSDUMP request. IEAVTSYS builds the SDUMP parameter list based on the dump options requested. The dump options are obtained from the CHNGDUMP operator command options, RTM2 passed options and SYS1.PARMLIB member IEADMR00. IEAVTSYS obtains storage for an SDUMP work area and SDUMP buffers. IEAVTSYS obtains the SDUMP buffers from 2 different storage areas. The first half is obtained from LSQA, and the second half is obtained from CSA user key. The address of these areas are passed to SDUMP via the SDUMP parmlist. A DECB is also built and the address is passed in the SDUMP buffer.</p> <p>When an abend is in progress (RTM2DMP1=0), IEAVTABD only takes one dump for each call from RTM2. On a zero return code from SDUMP, IEAVTSYS issues message IEA993I 'SYSDUMP taken to XXXX' and returns control to RTM2. On a return code of four, IEAVTSYS issues message IEA998I</p>	IEAVTSYS		<p>'PARTIAL SYSDUMP taken to XXXX' and returns control to ABDUMP. On a return code of eight from SDUMP, IEAVTSYS issues message IEA912I 'RECOVERY/TERMINATION DUMP FAILED' and returns control to RTM2 with the return code from SDUMP stored in RTM2SNCC. If SDUMP passes a return code of twelve, IEAVTSYS issues message IEA838I, 'SYSDUMP SUPPRESSED AS A DUPLICATE OF: original dump information', and returns control to ABDUMP. Return codes from SDUMP are:</p> <p>0 - Complete dump was taken. 4 - Partial dump was taken. 8 - System was unable to take a dump. 12 - SYSDUMP was suppressed.</p> <p>12 IEAVTABD closes the dump data set, frees the DCB storage, sets the dump flag indicator (TCBFS=0) to zero if an abend was in progress, and dequeues from the dump data set. IEAVTABD ensures that the storage acquired by RTM2 for the copied trace table and the RBBUFFER used by SNAP is freed. If an abend was in progress, IEAVTABD sets the subtask dispatchable.</p>		



IEAVTDSV - DUMPSRV Address Space Jobstep Task (Part 2 of 2)

Extended Description	Module	Label	Extended Description	Module	Label
<p>IEAVTDSV is attached as the job step task in the DUMPSRV address space. It sets up the DUMPSRV address space task structure and provides the interface to SVCDUMP and SYSDUMP for post SVCDUMP exit processing. It also sets up the address space for dump data set processing and for DAE processing. IEAVTDSV is invoked when the procedure DUMPSRV is read from SYS1.PROCLIB during system initialization; or IEAVTDSV is invoked by the SVCDUMP memory termination resource manager (IEAVTSDR) when the previous DUMPSRV fails. There is a program properties table entry for IEAVTDSV which contains the properties of the DUMPSRV address space.</p> <p>1 The dump data set processing task (IEECB926) is attached as a daughter of IEAVTDSV. It processes the DUMPDS command. An EXTR routine (IEAVTDSE) is provided to handle IEECB926 task termination. The IEECB926 task is a never-ending task unless a non-recoverable error occurs.</p> <p>2 IEAVTDSV attaches ADYTRNS, the dump analysis and elimination (DAE) transaction processor, as a daughter of IEAVTDSV. ADYTRNS processes transactions for all DAE functions. IEAVTDSV provides an end-of-task exit routine (IEAVTDSE) for ADYTRNS task terminations. ADYTRNS is a never-ending task and only stops when a non-recoverable error occurs.</p> <p>3 IEAVTDSV tries to locate the post dump exit list (IEAVTSEL) by using LOAD. If it is found, IEAVTDSV obtains storage for a table (DSVEXITL) from subpool 0, and places its address in the DSVCB (DSVEXITS). For each exit in IEAVTSEL, IEAVTDSV issues a LOAD. If the LOAD is successful, IEAVTDSV stores the module entry point in the DSVEXITL table. If a LOAD fails for any of the exits, IEAVTDSV issues message IEAV8331. If no valid exits are found, IEAVTDSV frees the DSVEXITL table (using module IEAVGM00) and clears the pointer to it (DSVEXITS). If at least one valid exit is found, IEAVTDSV obtains storage so that SVCDUMP can provide the post dump exits with a copy of the dump header. IEAVTDSV places the address of this 4K storage area in the RTCT (RTCTCPYH). IEAVTDSV then issues a DELETE for IEAVTSEL.</p>			<p>4 IEAVTDSV issues a WAIT, waiting for the SVCDUMP post-dump exit processor (IEAVTSD1) to post the ECB (DSVEECBE) to initiate the post exit processing.</p> <p>5 IEAVTDSV issues an ENQ on the post dump exit resource (SYSIEA01 SDPOSTEX) in order to serialize the post dump exit processing request queue. The queue is anchored in the DSVCB (DSVEANCH). IEAVTDSV removes a post-dump exit parameter list (PDEP) element from the queue for processing. The PDEP contains a copy of the dump header record and a DAE post-dump parameter list (DSPD). IEAVTDSV then issues DEQ to release the post dump exit resource. If the queue was empty, IEAVTDSV returns to step 4 to wait for the next post dump exit request.</p> <p>6 If the queue was not empty, IEAVTDSV attaches the SVCDUMP post exit processor (IEAVTSEP) as a daughter of IEAVTDSV. IEAVTDSV passes the PDEP as input to the post dump exit processor. IEAVTDSV returns to step 5 to check whether or not there are any more requests to process.</p> <p>If the ATTACH of IEAVTSEP is unsuccessful, IEAVTDSV frees the PDEP. IEAVTDSV issues a WAIT so that the DUMPSRV address space is placed in a never-ending wait to keep the IEECB926 task active.</p> <p>Recovery processing</p> <p>Recovery is provided by an ESTAE (ESTAEDSV). If IEAVTDSV fails to attach IEECB926, the ESTAE issues a retry to IEAVTDSV to attempt to process SVCDUMP post exits. If an error occurs during the post exit processing the ESTAE issues a retry to IEAVTDSV, putting post exit processing in a permanent wait, but allowing the IEECB926 task to remain active. The ESTAE records all errors and issues an SVCDUMP. If the task fails to recover, the DUMPSRV address space terminates and the SVCDUMP memory termination resource manager (IEAVTSDR) gets control to clean up any critical resources.</p>		

IEAVTSDB - MODULE DESCRIPTION

DESCRIPTIVE NAME: SVC Dump Disabled Interrupt Exit (DIE) Processor

FUNCTION:

IEAVTSDB protects critical SVC Dump resources through the use of disabled interrupt exits (DIEs). IEAVTSDB protects the summary dump real storage buffer (RSB) and system non-dispatchability. IEAVTSDB also protects the suspended unit of work during suspend summary dump processing.

ENTRY POINT: IEAVTSDB

PURPOSE: See function.

LINKAGE: CALL

CALLERS:

IEAVAD00,
IEAVTSDT,
IEAVTSDX,
IEAVTSSD

INPUT:

SDDIE contains a function code to indicate which function IEAVTSDB is to perform. SDDIE also contains pointers to all the working storage necessary for IEAVTSDB to run. There are two SDDIE control blocks in the system: one for the real storage buffer timer DIE and the other for the non-dispatchability timer DIE.

OUTPUT:

- . Adds a TQE to the timer queue which establishes the non-dispatchability DIE.
- . Schedules an SRB which adds a TQE to the real timer queue which establishes the RSB DIE.

EXIT NORMAL: Returns to the caller.

EXIT ERROR: No exit error conditions.

ENTRY POINT: RSBSRB

PURPOSE:

RSBSRB establishes the RSB timer DIE by invoking the SETDIE timer service.

LINKAGE: CALL

CALLERS: Dispatcher

INPUT:

SDDIE contains:

- . The first base register
- . Storage for the TQE
- . The time interval indicating how often the DIE is to receive control.

OUTPUT:

- . The TQE for the RSB DIE is placed on the timer queue.
- . Register 7 is set up to point to SDDIE at the time of the SETDIE. This causes register 7 to be pointing to SDDIE when the DIE is entered.

EXIT NORMAL: Returns to the dispatcher.

EXIT ERROR: No exit error conditions.

IEAVTSDB - MODULE DESCRIPTION (Continued)

ENTRY POINT: SDBDIE

PURPOSE:

SDBDIE is a general purpose DIE which handles the DIE interface for the RSB DIE and the non-dispatchability DIE.

LINKAGE: CALL

CALLERS: External interrupt TIMER SLIH.

INPUT:

SDDIE contains:

- . Work areas
- . Storage for the TQE
- . The time interval indicating how often the DIE is to receive control.

OUTPUT:

- . Resources are cleaned up by the RSB DIE and NDISPDIE subroutines.
- . If any resources remain to be protected, the TQE for the DIE is placed back on the timer queue.

EXIT NORMAL: Returns to the external interrupt TIMER SLIH.

EXIT ERROR: No exit error conditions.

ENTRY POINT: NDISPSRB

PURPOSE: NDISPSRB sets the system or tasks dispatchable.

LINKAGE: CALL

CALLERS: Dispatcher

INPUT:

SDDIE contains:

- . The first base register
- . Storage for use by the SRB.

OUTPUT: Various parts of the system are set dispatchable.

EXIT NORMAL: Returns to the dispatcher.

EXIT ERROR: No exit error conditions.

ENTRY POINT: SDBFRR

PURPOSE:

SDBFRR is a functional recovery routine (FRR). It recovers from errors in IEAVTSDB.

LINKAGE: Call

CALLERS: RTM

INPUT:

FRR parameter list (mapped by FRRPARAM) which contains the following:

- . Base register
- . Data register
- . Address of SDDIE
- . Flags indicating which resources need to be recovered.

OUTPUT:

- . The system diagnostic work area (SDWA) and the variable recording area (VRA) are filled in with diagnostic data.
- . Recording to SYS1.LOGREC is requested.

IEAVTSDB - MODULE DESCRIPTION (Continued)

EXIT NORMAL: Return to RTM.

EXIT ERROR: No exit error conditions.

EXTERNAL REFERENCES:

ROUTINES:

- IARXPLCK - Obtains RSM serialization.
- IARXPRSB - Frees RSM serialization for the real storage buffer.
- IEAVESRT - Resets the SVC Dump caller for STOP/RESET functions.

SETDIE - Establishes a TQE on the timer queue.

CONTROL BLOCKS:

Common name	Macro Name	Usage
ASCB	IHAASCB	read
ASVT	IHAASVT	read
ASXB	IHAASXB	read
CVT	CVT	read
FRRS	IHAFRRS	read, write
OUCB	IRAUCB	read
PSA	IHAPSA	read
PVT	IHAPVT	read
RB	IHARB	read
RB	IKJRB	read
RTCT	IHARTCT	read, write
RTSD	IHARTSD	read, write
SCVT	IHASCVT	read
SDDAT	IHASDDAT	read
SDDIE	IHASDDIE	read, write
SDRSN	IHASDRSN	write
SDUMP	IHASDUMP	read
SDWA	IHASDWA	read, write
SDWOK	IHASDWRK	read, write
SMDLR	IHASMDLR	read
SMWK	IHASMWK	read, write
SMWKRSCB	IHASDRSB	read, write
SRB	IHASRB	read, write
SVT	IHASVT	read
TCB	IKJTCB	read
TPC	IEAVVTPC	read
TQE	IHATQE	read, write
VRAMAP	IHAVRA	read, write

TABLES: No tables used.

SERIALIZATION:

- IEAVTSDB uses:
- . The dispatcher lock to serialize the TQE element
 - . RSM locks to serialize the RSB
 - . The local lock to serialize calls to STATUS
 - . SDDIESER to serialize the SDDIE storage
 - . RTCTSDPL and CVTSDBUF(high-order bit) are set to serialize all of the SVC dump process.

IEAVTSDB - MODULE OPERATION

IEAVTSDB receives control to protect the summary dump real storage buffer (RSB) and system non-dispatchability. IEAVTSDB establishes disabled interrupt exits (DIEs) to protect these resources. A DIE is a routine that receives control (disabled) at specified intervals for I/O and external interrupts. IEAVTSDB uses a timer queue element (TQE) to identify when the DIE receives control and to provide the address of the DIE.

IEAVTSDB has five entry points.

Entry point IEAVTSDB performs the following:

- . Schedules an SRB which establishes a DIE to protect the summary dump real storage buffer (RSB). The RSB contains the summary dump data. IEAVTSDB needs an SRB because the SETDIE service requires the dispatcher lock and higher locks may be held on branch entry SVC dump invocations of this module.
- . Establishes a DIE to protect the system non-dispatchability. This DIE also protects the system from resource deadlocks during suspend summary dump processing.

Entry point RSBSRB establishes a DIE to protect the summary dump RSB. RSBSRB uses the SETDIE service to establish the DIE.

Entry point SDBDIE passes control to the DIE routines which protect the system resources.

- . Subroutine RSDIE protects the RSB. RSDIE frees RSB frames when SVC Dump appears to hang while holding real frames in the RSB. This condition occurs when no RSB frames are freed within a DIE time interval. The RSDIE frees the RSB frames to prevent SVC Dump from indefinitely holding on to real storage.
- . Subroutine NDISPDIE protects system non-dispatchability and the suspend summary dump process. NDISPDIE determines if SVC Dump is hung with the caller's unit of work stopped for suspend summary dump processing. SVC Dump determines this by examining a count of the number of virtual storage buffer blocks filled. If this count remains the same across the DIE time interval, SVC Dump is hung. The DIE resets the suspended unit of work to try to relieve possible system resource contention involving that suspended unit of work.

Subroutine NDISPDIE then determines if SVC DUMP is hung while the system or tasks are set non-dispatchable. SVC Dump determines whether it is hung by examining a count of the number of records written to the dump data set. If the count remains the same across the DIE time interval, SVC Dump is hung. The DIE schedules an SRB to reset the system or tasks dispatchable. The SRB is required because the DIE cannot obtain the local lock which is needed for branch entries to STATUS.

Entry point NDISPSRB invokes STATUS to set the system or tasks dispatchable. Subroutines NDISPDIE or NDISPSRB schedule this SRB.

RECOVERY OPERATION:

Entry point SDBFRR is the functional recovery routine (FRR) that provides recovery for all entry points in IEAVTSDB. It does the following:

- . Fills in the SDWA with diagnostic data.
- . Fills in the VRA with component data.
- . If possible, retries the request.
- . Frees locks which were obtained by IEAVTSDB.
- . Releases SDDIE serialization.
- . Returns to RTM.

IEAVTSDB - DIAGNOSTIC AIDS

ENTRY POINT NAMES: IEAVTSDB
RSBSRB
SDBDIE
NDISPSRB
SDBFRR

MESSAGES: None

ABEND CODES: None

WAIT STATE CODES: None

RETURN CODES:

EXIT NORMAL:

Decimal values in field SDWRCDE:

- 0 - Percolate the error. Any FRRs established prior to this one receive control
- 4 - Retry the request.

REGISTER CONTENTS ON ENTRY:

ENTRY POINT IEAVTSDB: Irrelevant

ENTRY POINT RSBSRB: Irrelevant

ENTRY POINT SDBDIE:

Register 0 - Irrelevant
Register 1 - Address of the TQE
Registers 2-6 - Same as when the SETDIE was issued
Register 7 - Address of the SDDIE
Registers 8-10 - Same as when the SETDIE was issued
Register 11 - First base register
Register 12 - Second base register
Register 13 - Irrelevant
Register 14 - Return address
Register 15 - Entry point address

NOTE = The SETDIE service saves registers 2-12 at the time the SETDIE is issued. These registers are restored on entry to the DIE.

ENTRY POINT NDISPSRB: Irrelevant

ENTRY POINT SDBFRR:

Register 0 - Address of 200-byte FRR work area
Register 1 - Address of the SDWA
Registers 2-13 - Irrelevant
Register 14 - Return address of RTM
Register 15 - Entry point address to the FRR

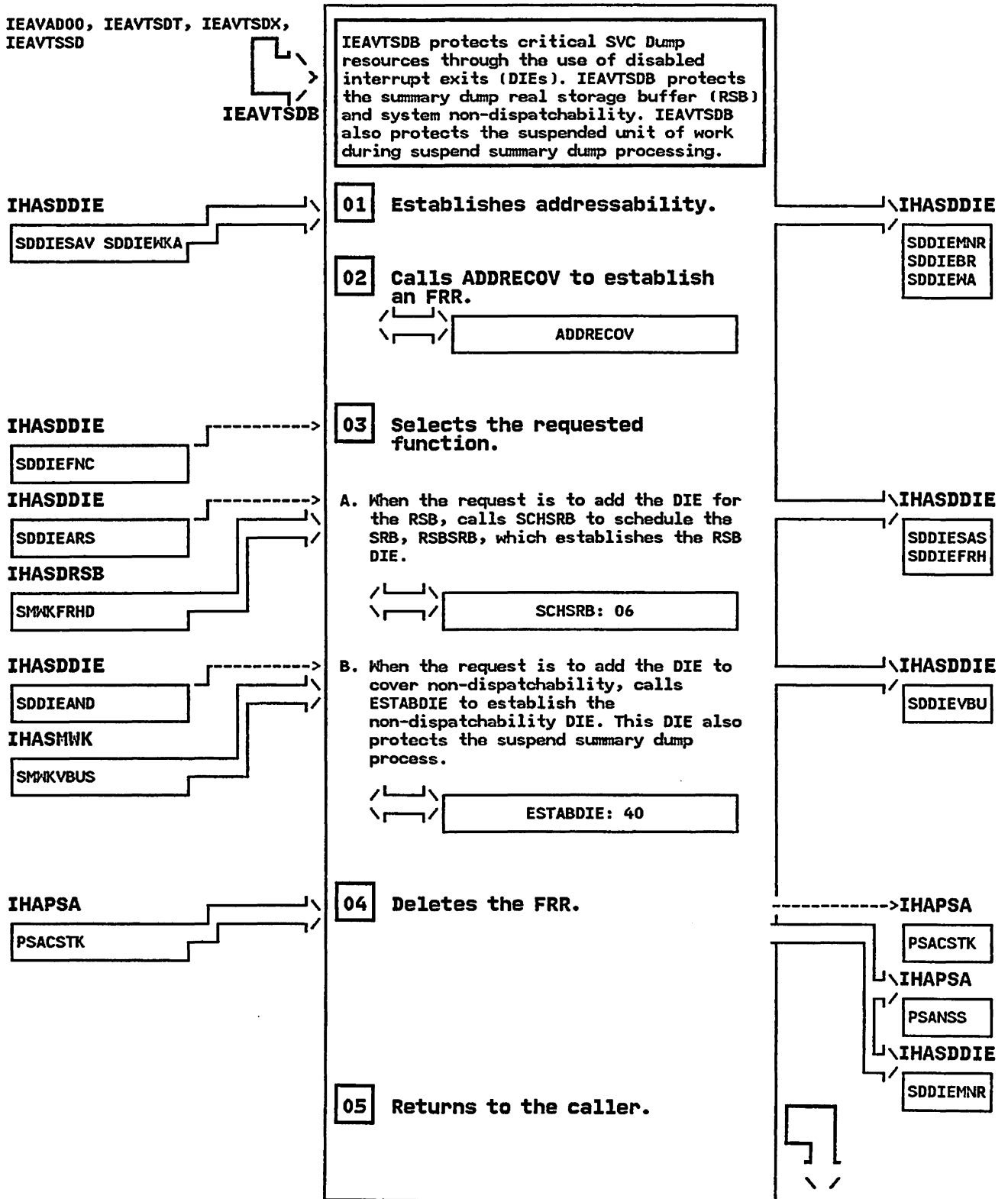
REGISTER CONTENTS ON EXIT:

EXIT NORMAL:

Registers 0-13 - Unpredictable
Register 14 - Unchanged
Register 15 - 0

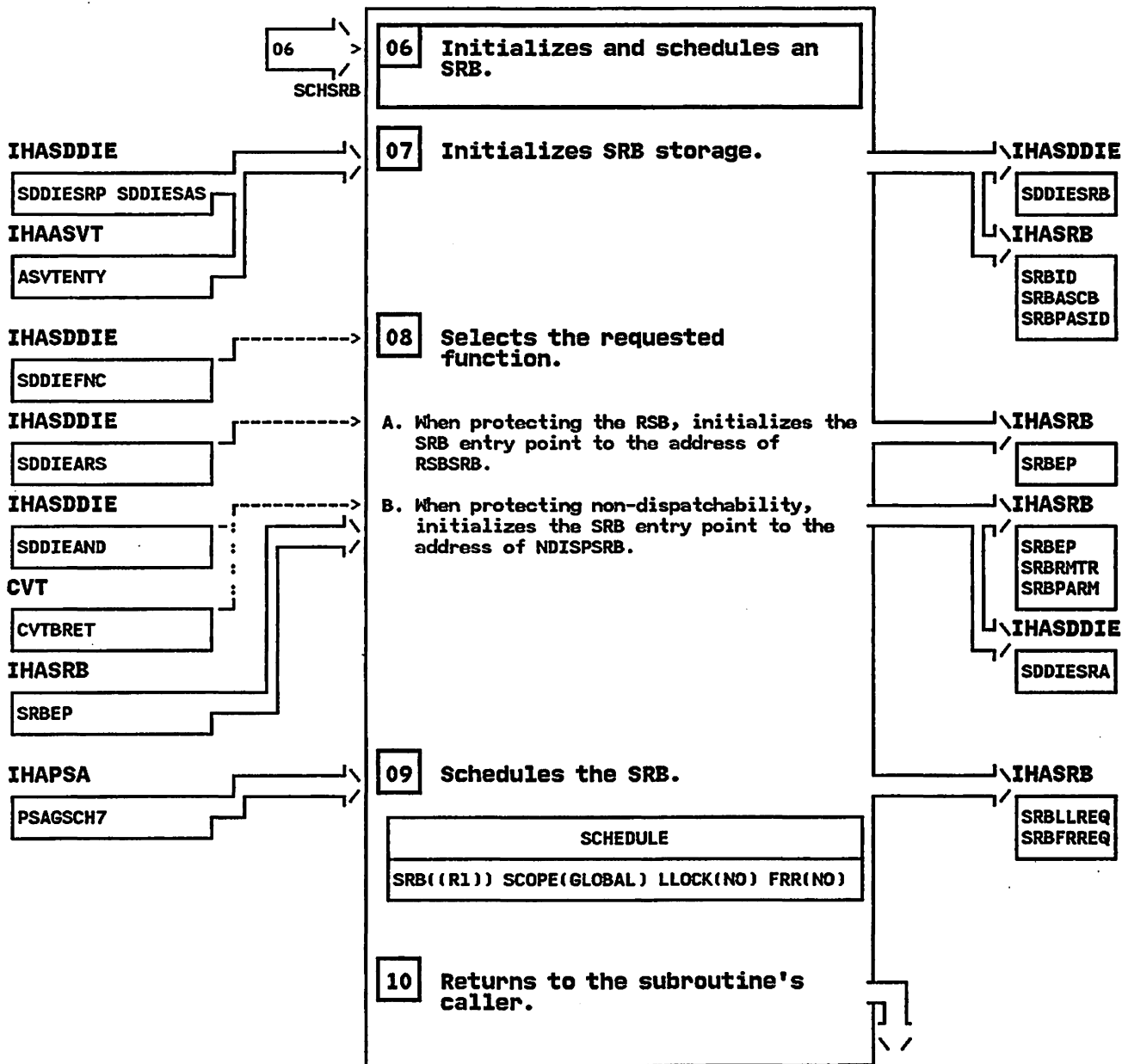
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 01



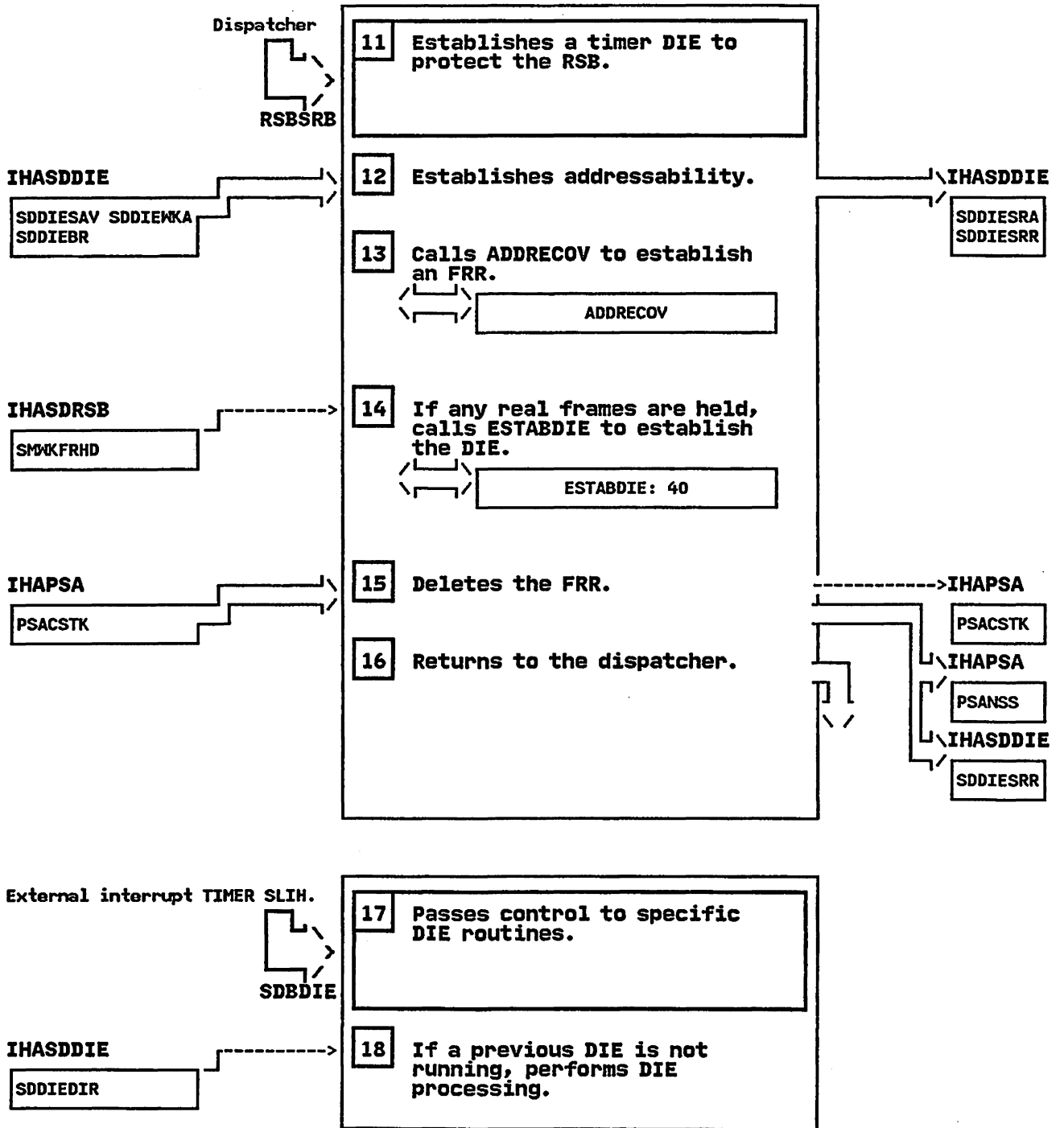
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 06



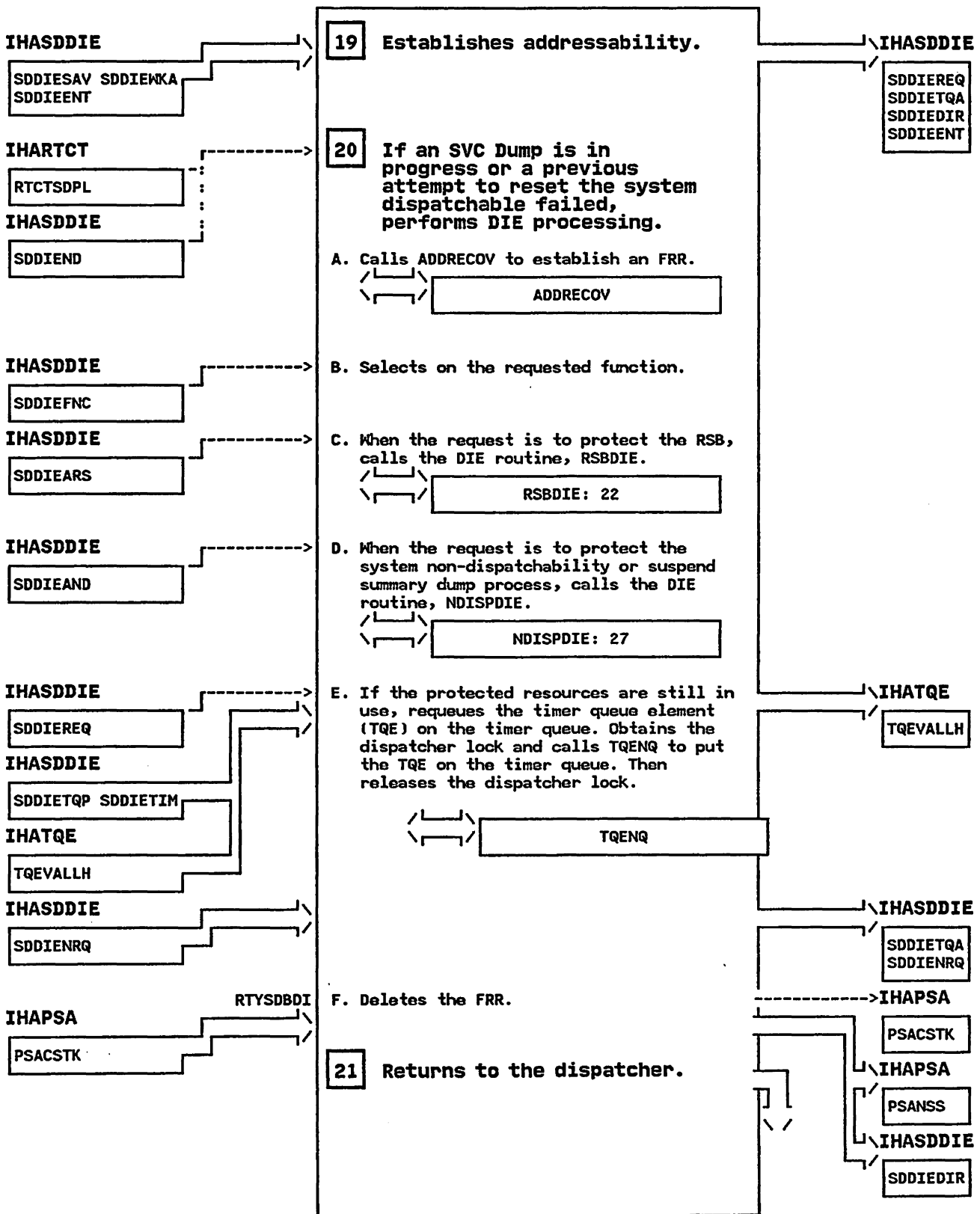
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 11

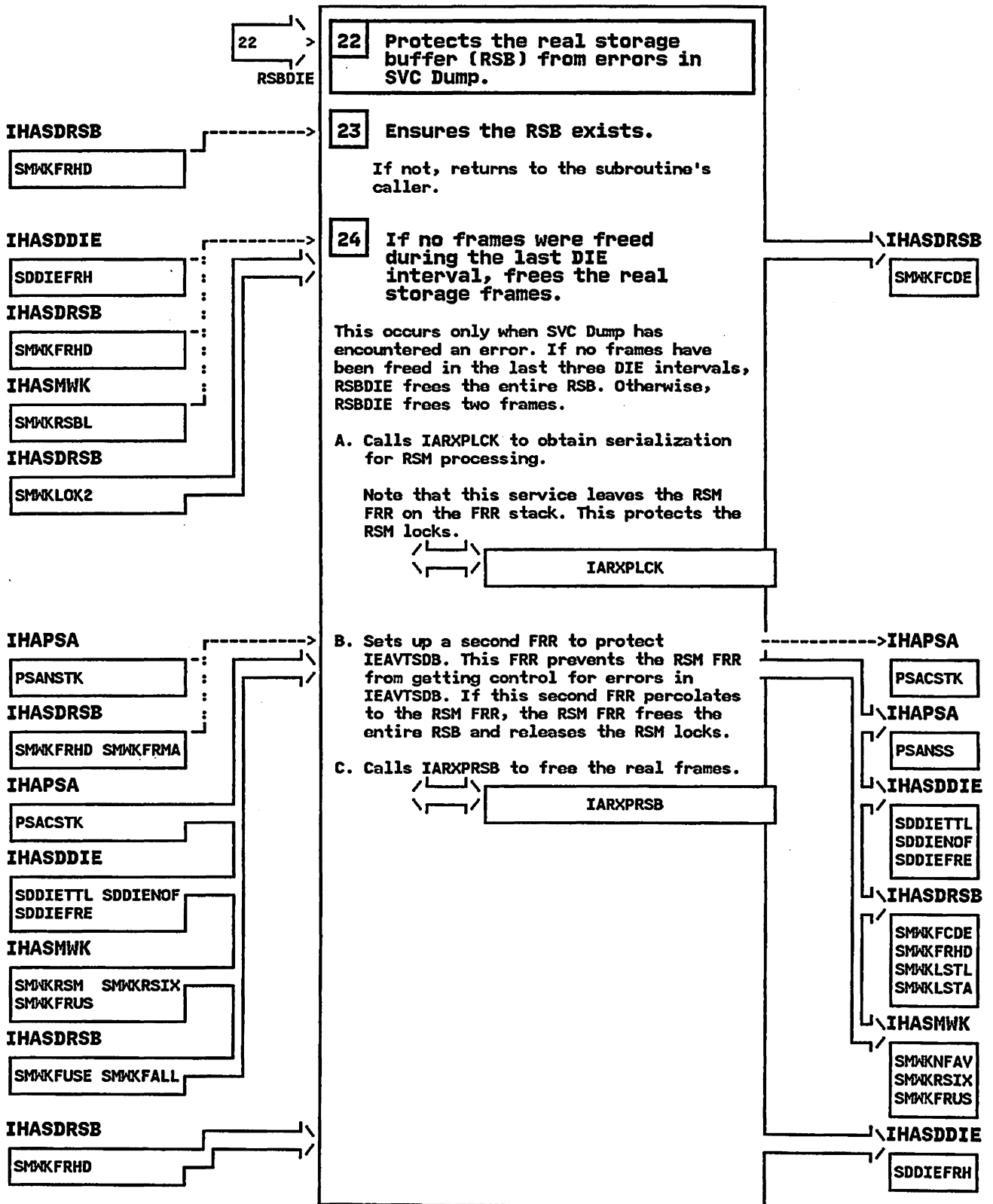


IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 19

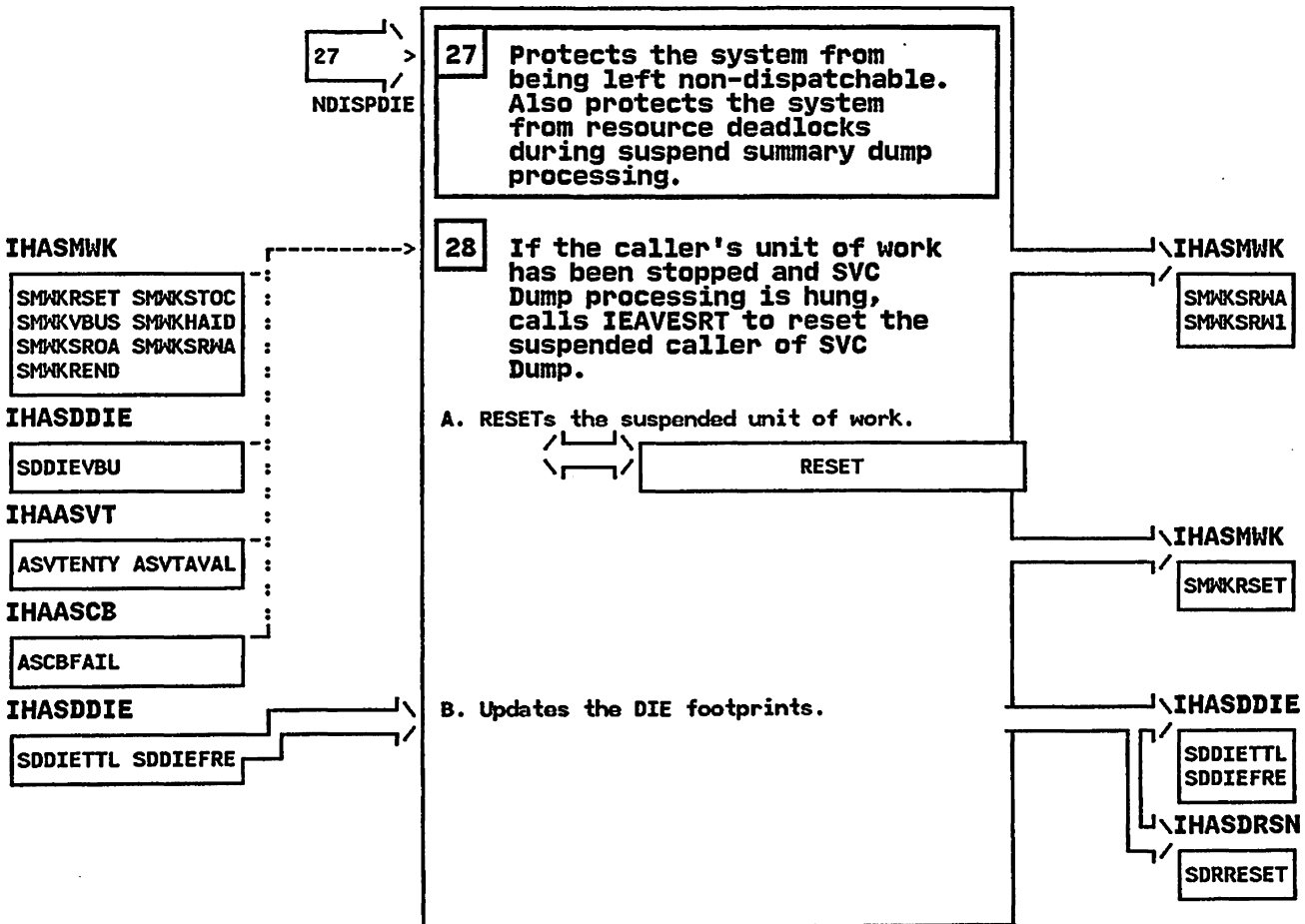
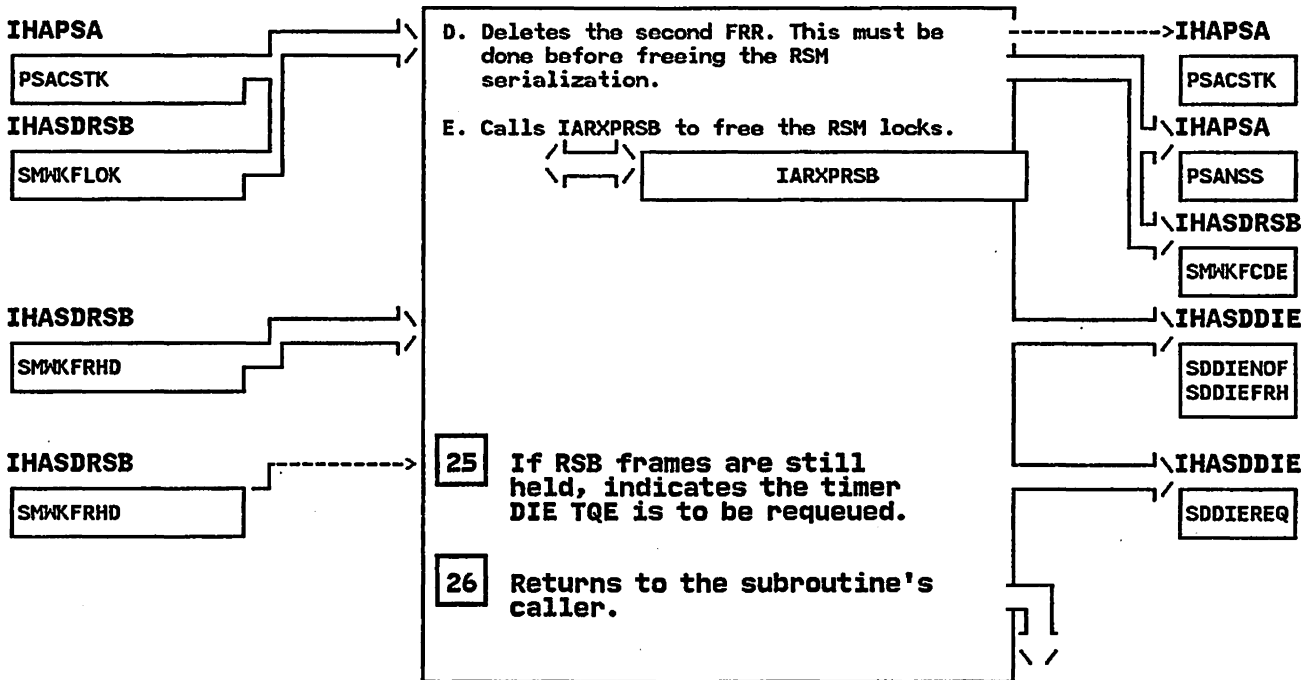


IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor



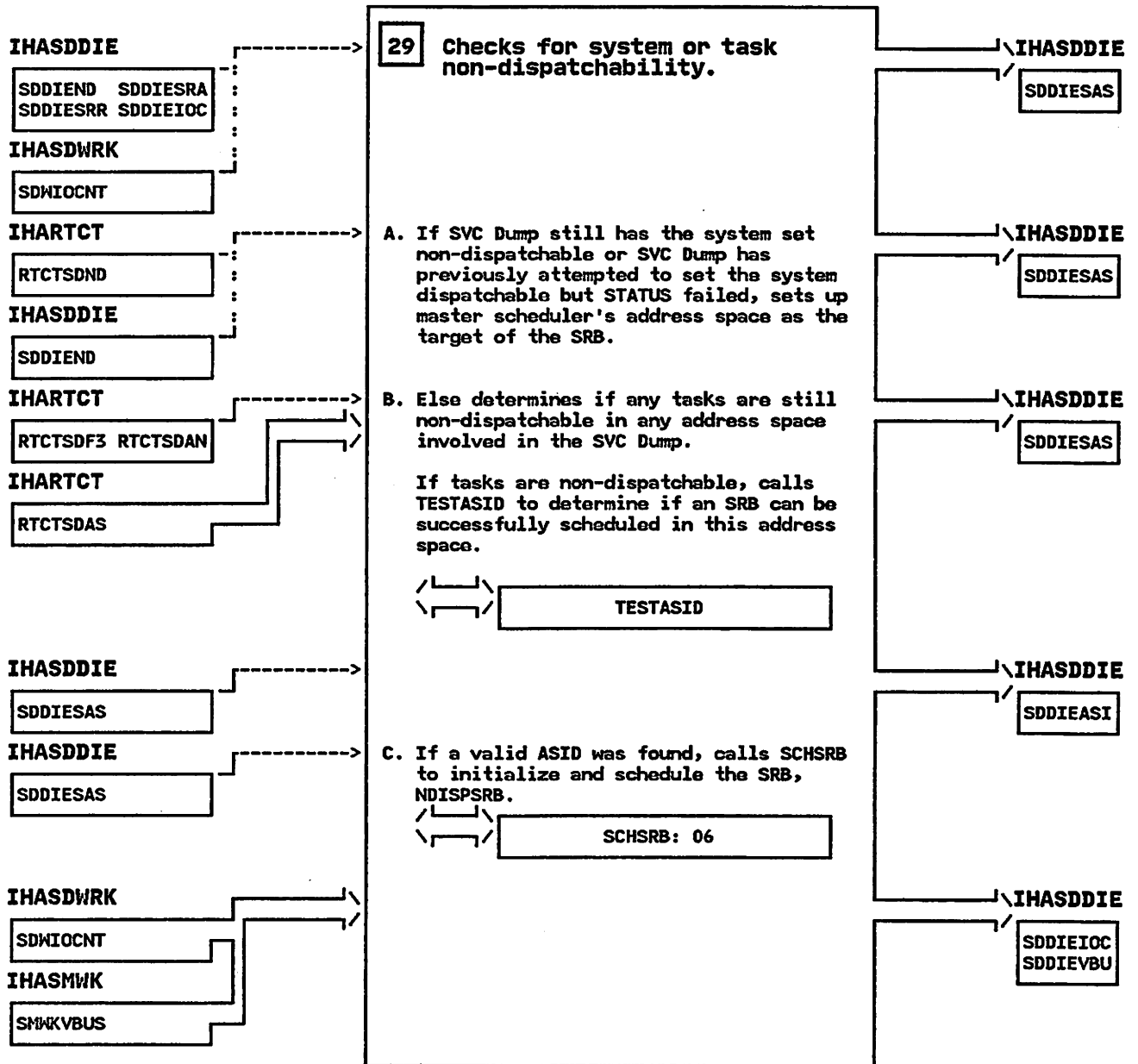
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 24D



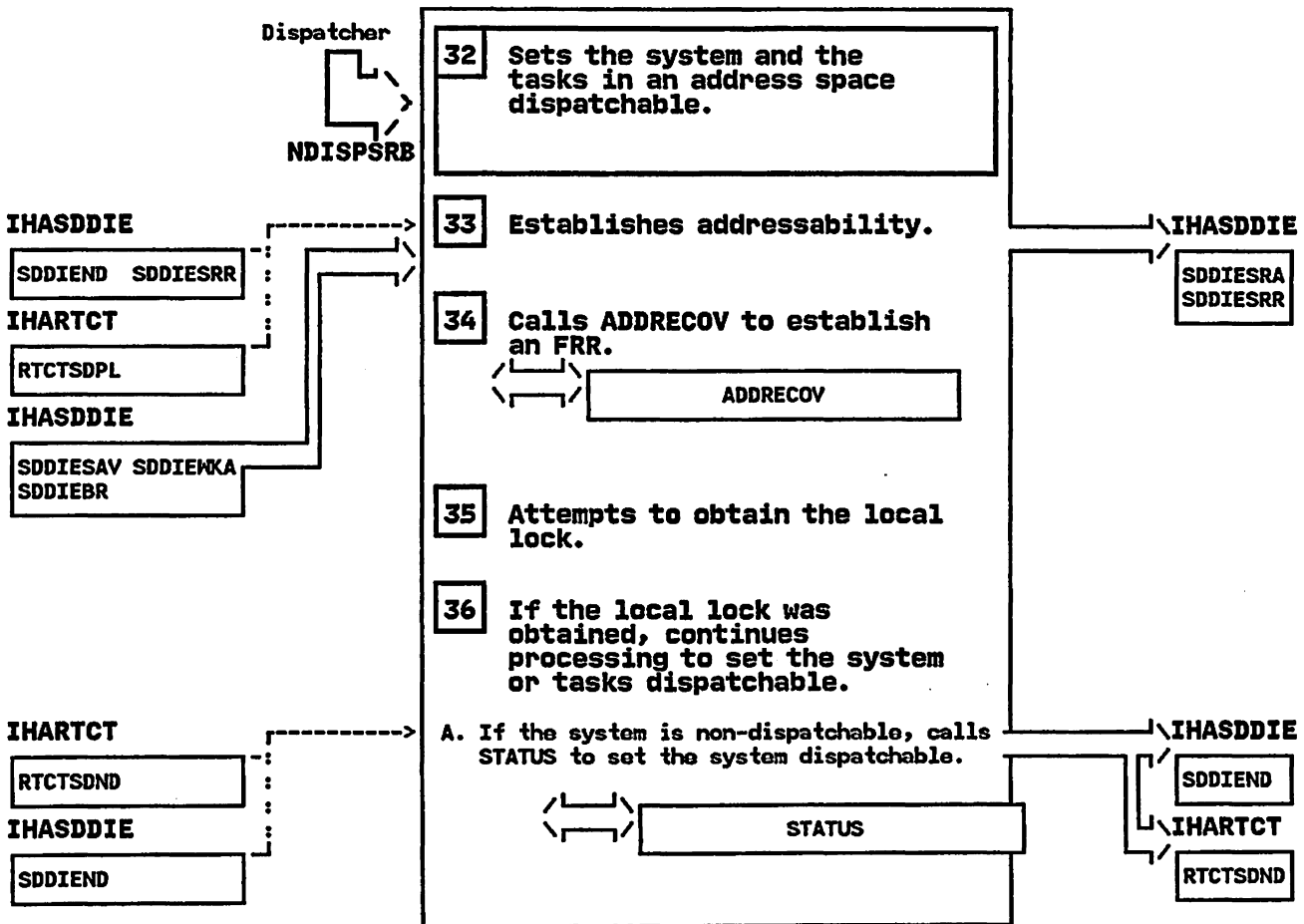
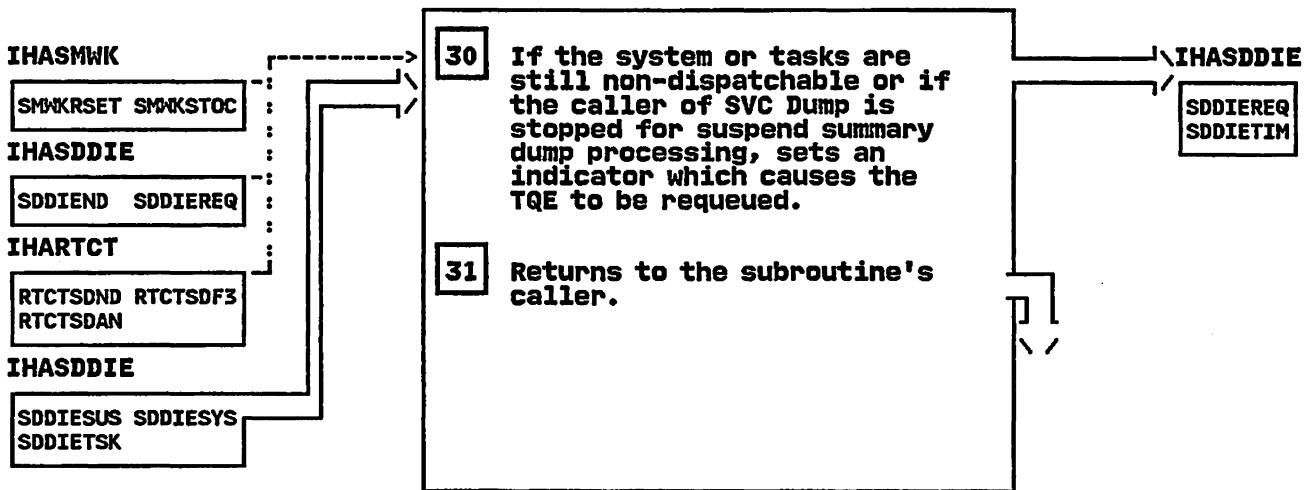
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 29



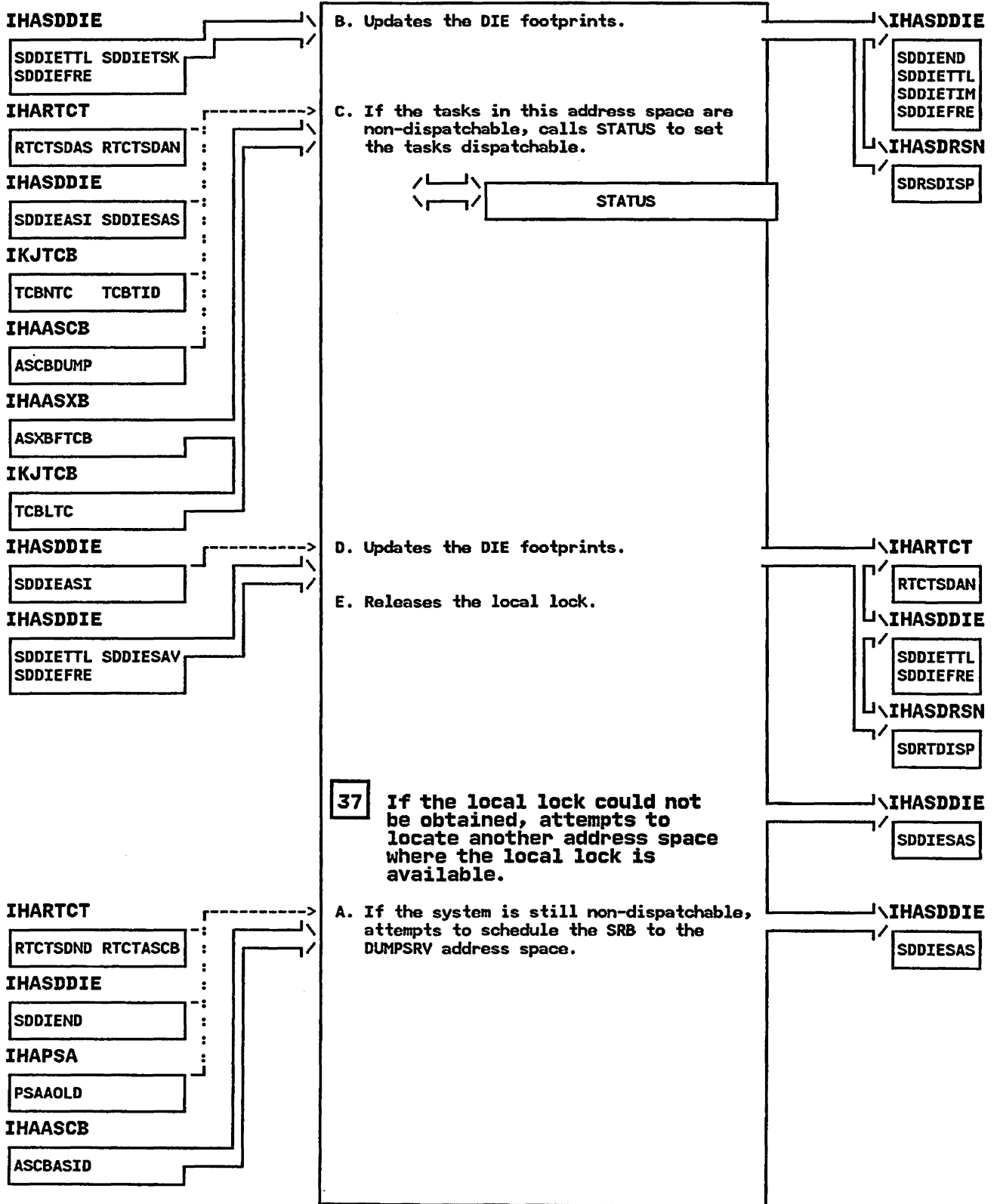
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 30



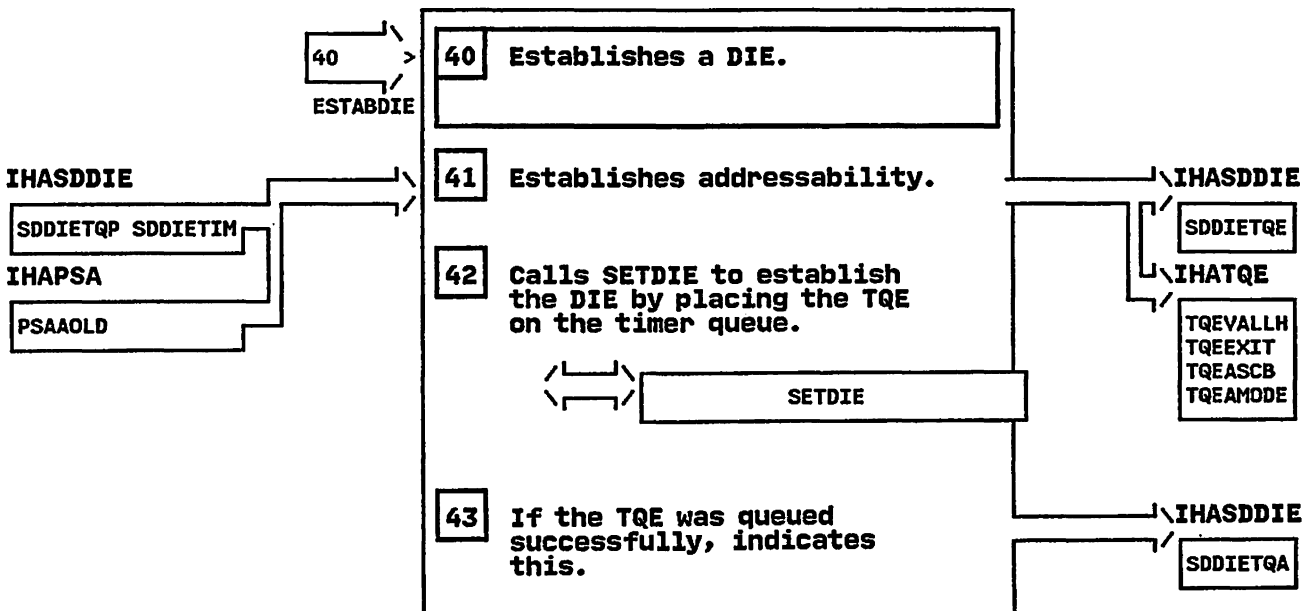
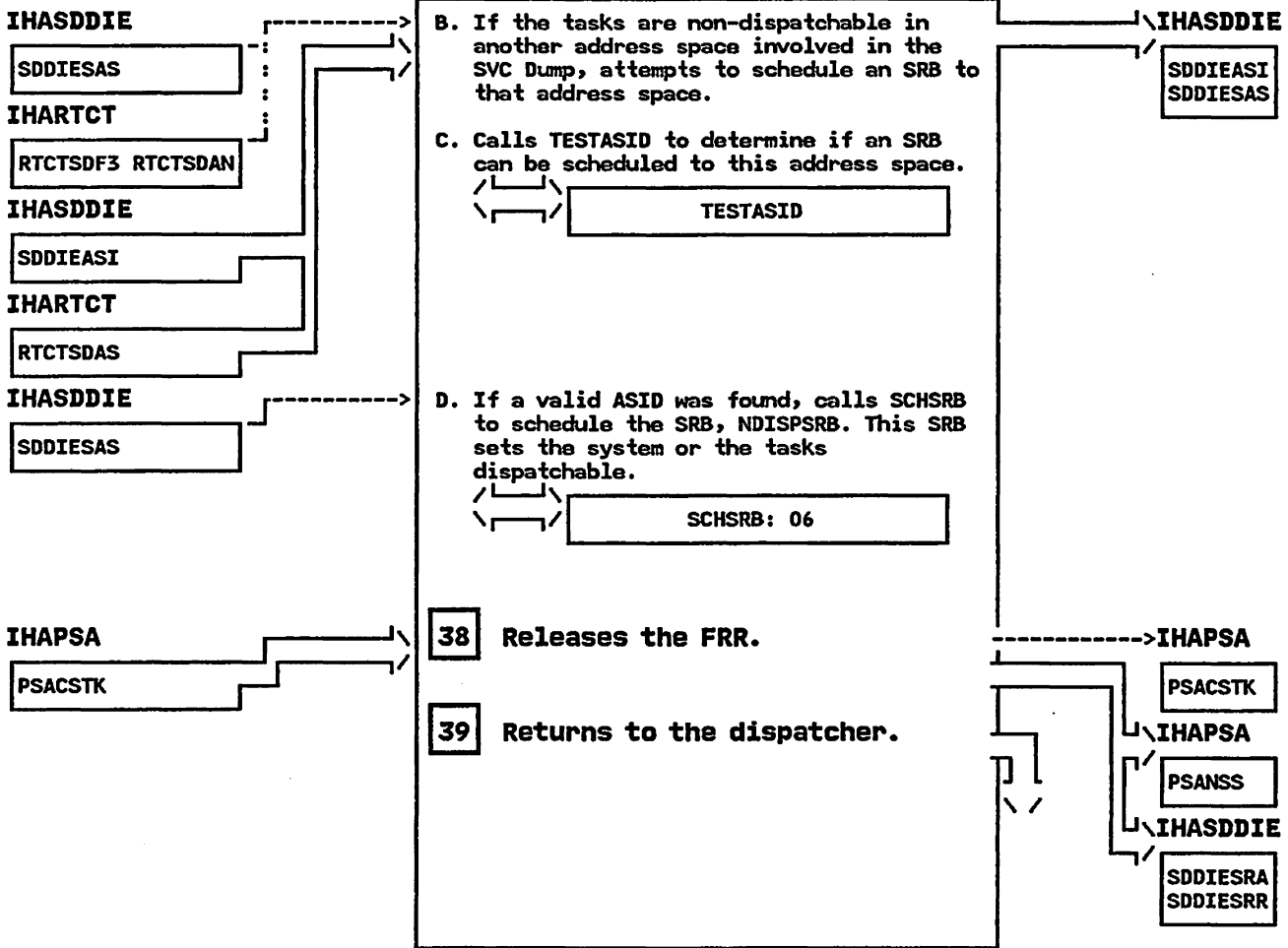
IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

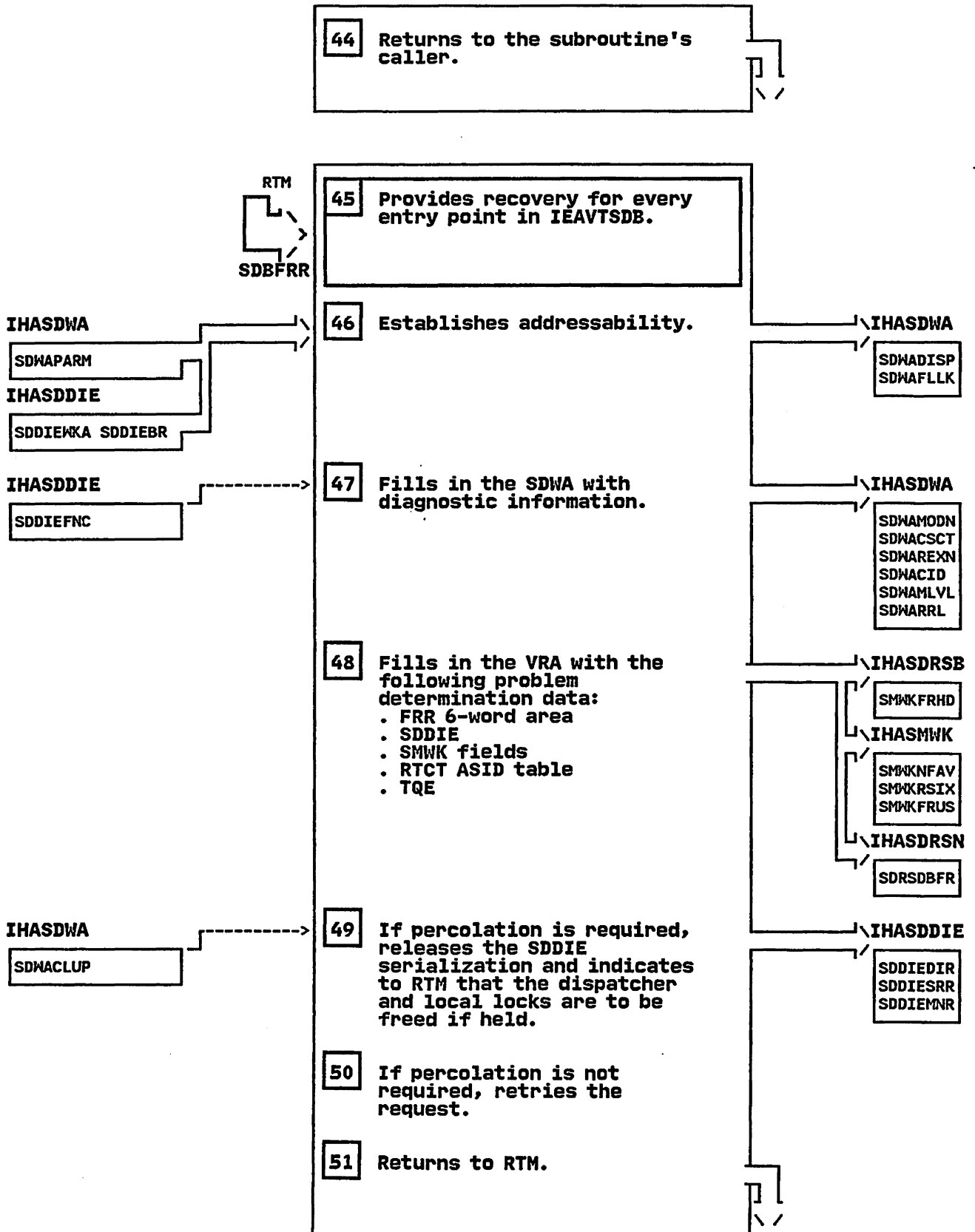
STEP 36B



IEAVTSDB - SVC Dump Disabled Interrupt Exit (DIE) Processor

STEP 37B



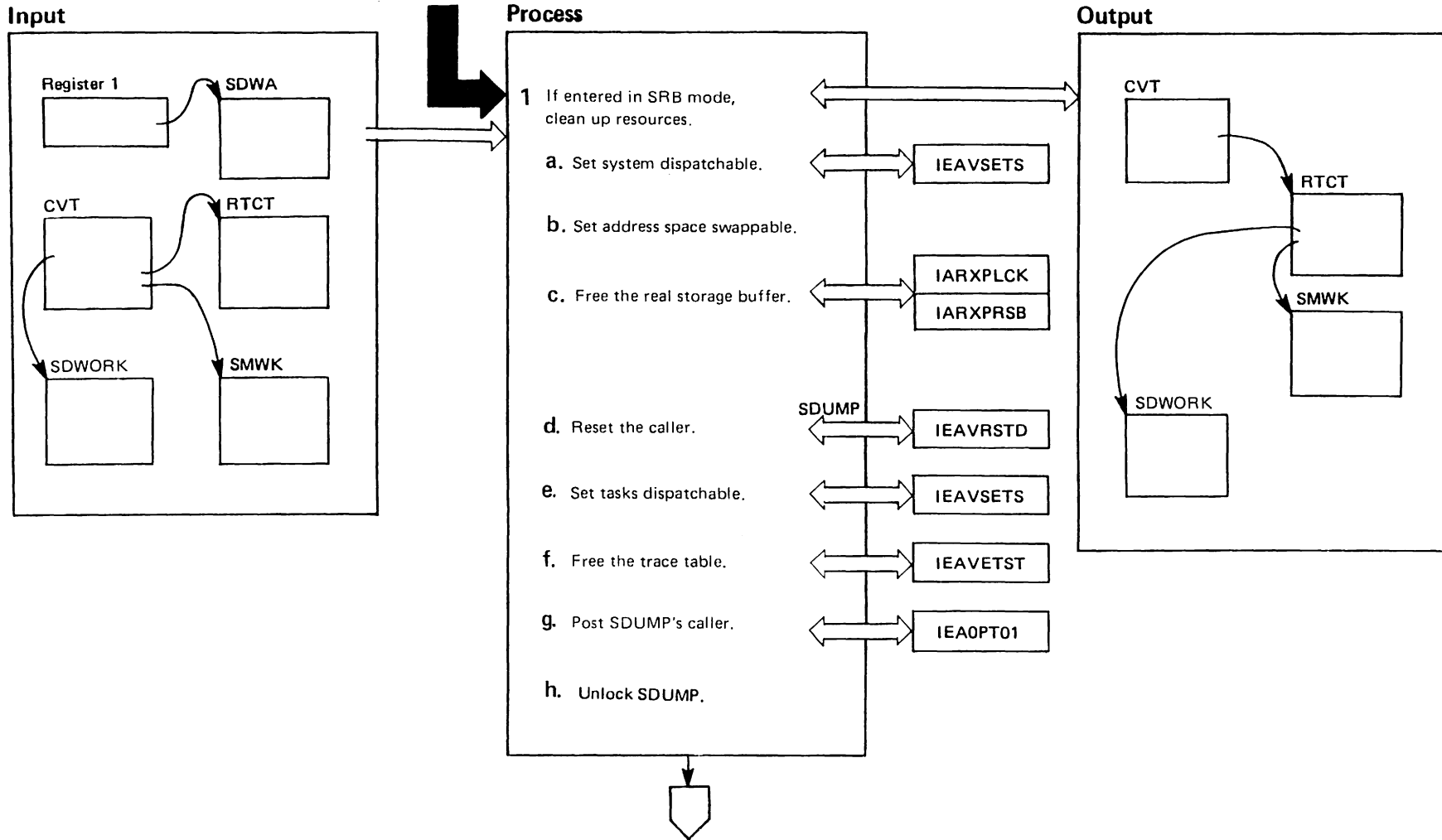


IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 1 of 6)

DMP-110 MVS/XA SLL: Dumping Ser LY28-1670-1 (c) Copyright IBM Corp. 1987, 1989

From IEAVTMMT
for memory termination
From IEAVTSKT
for task termination

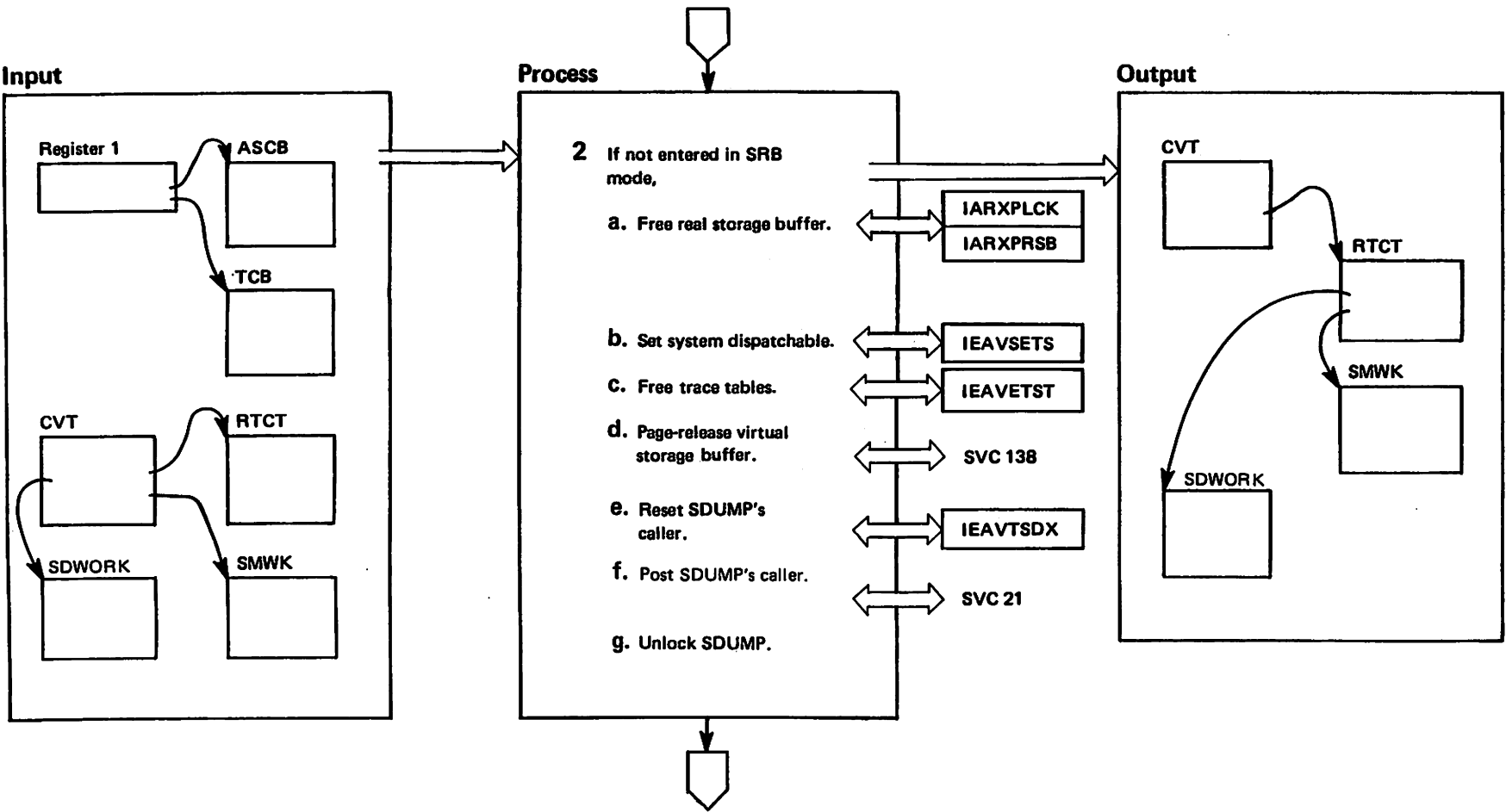
From RTM
for SRB's FRR
process



IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 2 of 6)

Extended Description	Module	Label	Extended Description	Module	Label
<p>IEAVTSDR has three major functions:</p> <ul style="list-style-type: none"> ● Contains the FRR for the SDUMP SRB (label SCHSRB in IEAVTSDX). This FRR cleans up any critical system resources that SDUMP held. ● Is the task and memory termination resource manager for address spaces and dump tasks (IEAVTSDT) involved in an SDUMP. ● Is the memory termination resource manager for the DUMPSRV address space. <p>1 If IEAVTSDR is entered in SRB mode, it is running as the FRR for the SRB (label SCHSRB in IEAVTSDX). IEAVTSDR does the following to clean up resources if necessary:</p> <ul style="list-style-type: none"> a. If this is the first SRB to run, IEAVTSDR sets the system dispatchable. b. If the SRB issues a SYSEVENT DONTSWAP, IEAVTSDR sets the address space in which the SRB is running, swappable. c. IEAVTSDR frees the real storage buffer by taking the following steps: <ul style="list-style-type: none"> ● Obtains the CPU lock to provide disablement ● Calls IARXPLCK to obtain RSM serialization ● Calls IARXPRSB to free the real storage buffer ● Calls IARXPRSB to free RSM serialization ● Frees the CPU lock 			<ul style="list-style-type: none"> d. If a suspend summary dump was being processed and the SRB in the DUMPSRV address space failed, IEAVTSDR resets SDUMP's caller to prevent terminating the caller. IEAVTSSD performs the STOP. e. IEAVTSDR sets the tasks in the address space dispatchable if the SRB set them non-dispatchable. f. If SDUMP obtained the trace table, IEAVTSDR issues a SNAPTRC FREE to free it. g. If SDUMP's caller specified the ECB option on the SDUMP SDUMP macro, IEAVTSDR posts the caller's ECB using cross memory post. IEAVTSDR does not post the caller if the caller's memory is terminating. h. If the SRB that failed is the first, last or only SDUMP SRB, and the dump cannot continue, IEAVTSDR unlocks SDUMP. 		

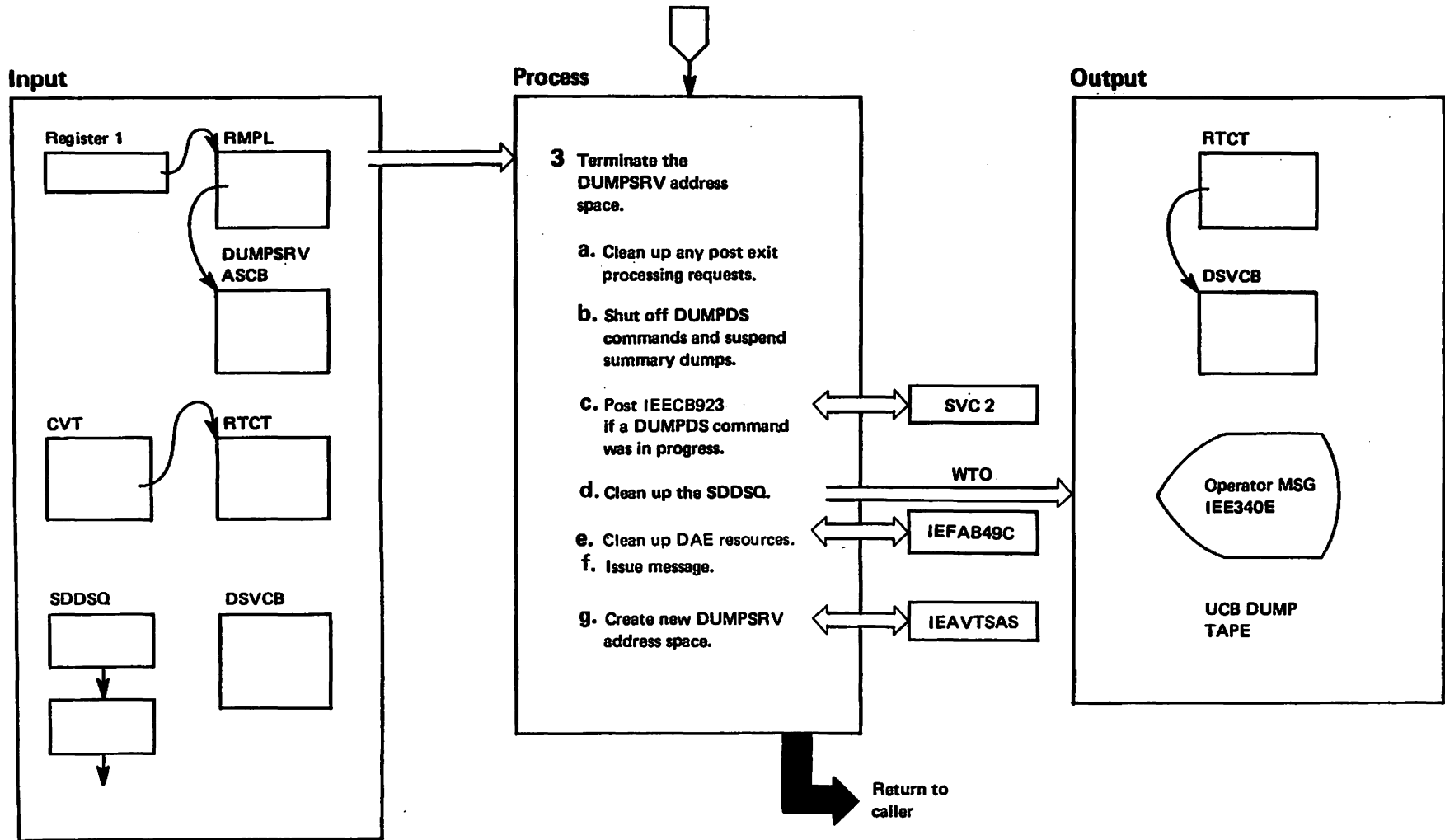
IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 3 of 6)



IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 4 of 6)

Extended Description	Module	Label
<p>2 If IEAVTSDR is not entered in SRB mode, it is running as the task or memory termination resource manager. The resource manager parameter list (RMPL) contains bits that indicate normal/abnormal and task/address space termination as well as the TCB and ASCB address of the terminating function. IEAVTSDR performs the following cleanup only if SDUMP is active and the terminating task or memory is involved in the SDUMP.</p> <ul style="list-style-type: none">a. Frees the real storage buffer as in step 1c.b. Sets the system dispatchable.c. Frees the trace table as in step 1f.d. If IEAVTSDR is in control to handle the dump task (IEAVTSDT) failing in the DUMPSRV address space, and it is running a suspend summary dump, IEAVTSDR page-releases the virtual storage buffer.e. Resets SDUMP's caller as in step 1d.f. Posts SDUMP's caller as in step 1g.g. Unlocks SDUMP as in step 1h.		

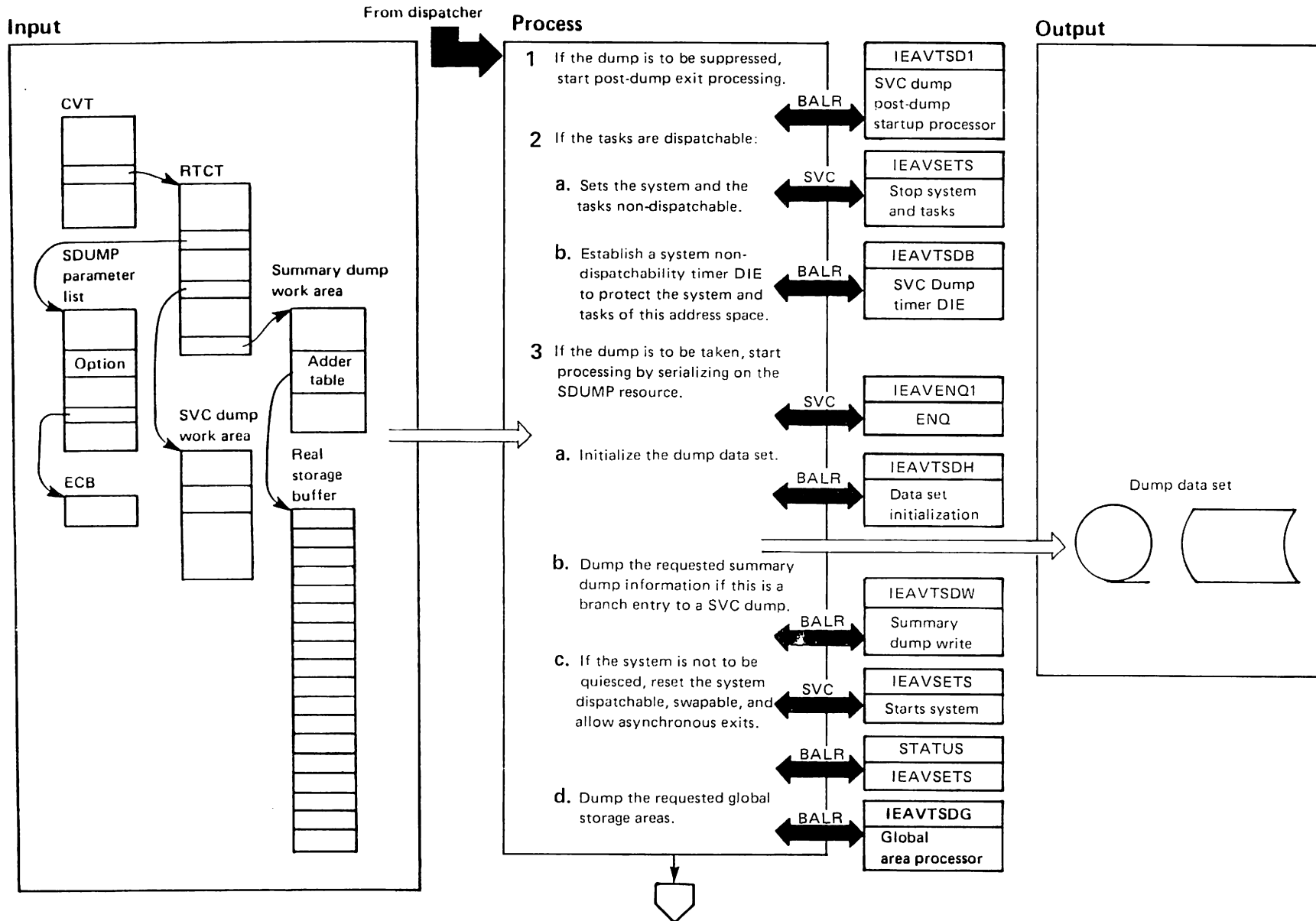
IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 5 of 6)



IEAVTSDR - SVC Dump/DUMPSRV Resource Manager (Part 6 of 6)

Extended Description	Module	Label	Extended Description	Module	Label
<p>3 If IEAVTSDR is entered for a normal or abnormal termination of the DUMPSRV address space,</p> <p>a. IEAVTSDR issues an ENQ on the post dump exit resource name to serialize cleanup with a possible SDUMP. If the ENQ is successful, the common storage on the post dump exit processing queue is freed. IEAVTSDR issues a DEQ on the post dump exit resource name.</p> <p>b. IEAVTSDR clears flags and fields in the RTCT and DSVCB to prevent any future DUMPDS commands, suspend summary dumps, or post exit processing.</p> <p>c. If a DUMPDS command was being processed by IEECB926, IEAVTSDR posts IEECB923 to allow the command to complete.</p> <p>d. IEAVTSDR marks all entries on the SDUMP data set queue (SDDSQ) that were added with the DUMPDS command as unusable. IEAVTSDR issues message IEE340E for each data set that is marked unusable. If a SYS1.DUMP tape was added with DUMPDS, IEAVTSDR calls IEFAB49C to rewind and unload the tape. IEAVTSDR clears the attention index in the UCB to indicate the tape is no longer controlled by SDUMP.</p> <p>e. IEAVTSDR cleans up DAE resources. IEAVTSDR does the following:</p> <ul style="list-style-type: none"> ● Resets the current DAE default options to stop DAE. ● Deletes the DAE low-priority transaction queue. ● Deletes the DAE high-priority transaction queue. ● Frees the storage for each entry in the high-priority queue. ● Deletes any outstanding DAE operator messages via the DOM macro instruction. ● Deletes the DAE in-storage symptom queue via the CPOOL DELETE macro instruction. 			<p>f. IEAVTSDR issues message IEA607E/I to indicate whether the DUMPSRV address space is going to be restarted.</p> <p>g. If the address space is to be restarted, IEAVTSDR links to module IEAVTSAS to perform the call to the system address space initialization function. If the address space create fails, IEAVTSDR issues message IEA768I to inform the operator.</p>		

IEAVTSDT - Resident Dump Task Processing (IEAVTSDT) (Part 1 of 6)



IEAVTSDT - Resident Dump Task Processing (IEAVTSDT) (Part 2 of 6)

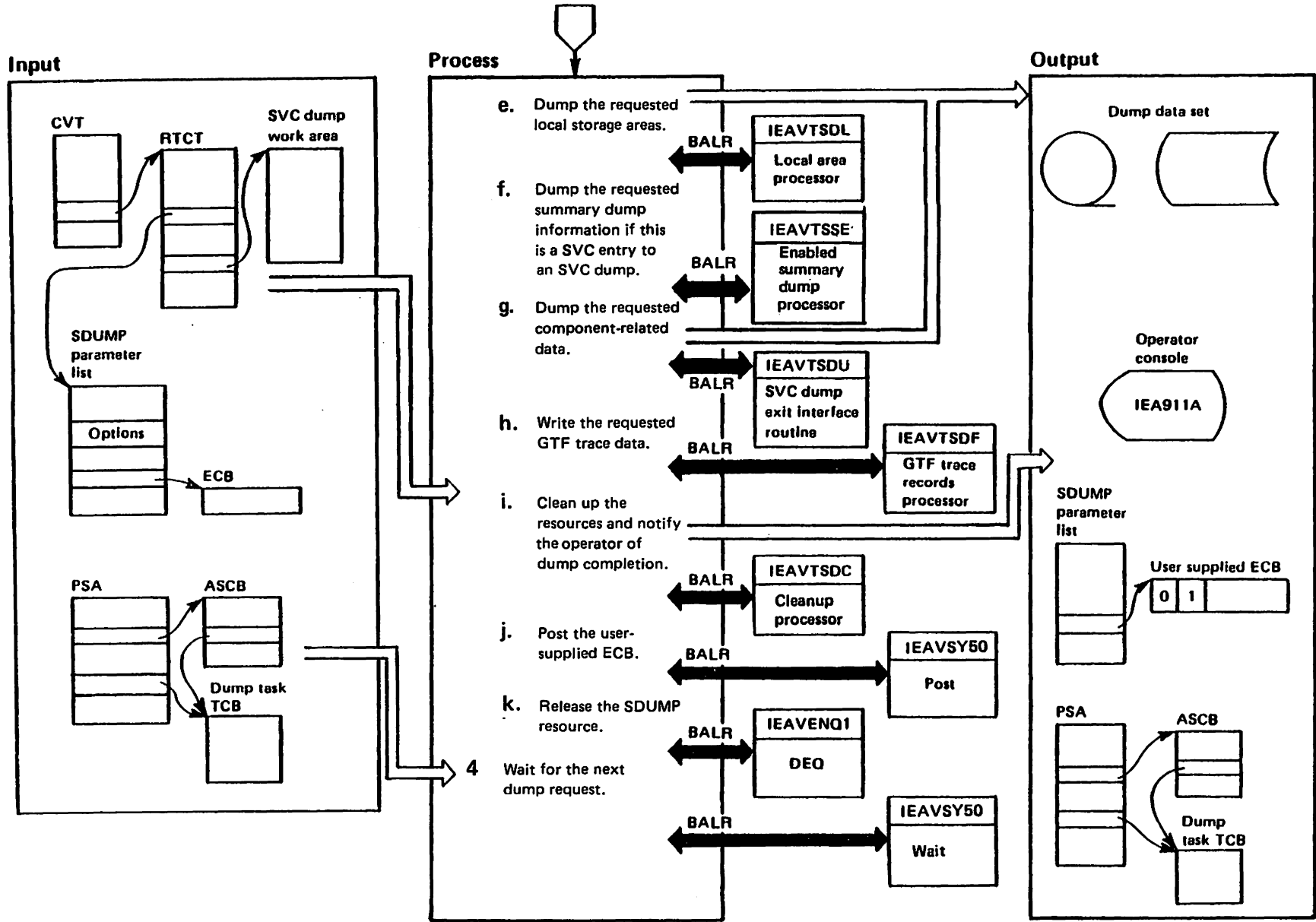
Extended Description	Module	Label	Extended Description	Module	Label
The dispatcher gives IEAVTSDT control as the result of a POST request from an SDUMP SRB.	IEAVTSDX	SCHSRB	a. If a dump data set has not been selected and initialized for this SVC dump request, IEAVTSDT calls module IEAVTSDH to select and initialize the dump data set for this dump. IEAVTSDH writes out the SDUMP 4K buffer if BUFFER=YES was specified on the SDUMP macro. If IEAVTSDH is running on a processor with an active MSSF and the processor is not a part of a complex with a VM machine, then IEAVTSDH uses SVC 122 to obtain the console loop trace data. If the console loop trace data has not previously been read, IEAVTSDH calls IEAVTSDP to dump the console loop trace data on the dump data set.	IEAVTSDH	
1 If the dump task has been posted to start post-dump processing, this means that DAE indicated that the dump should be suppressed. IEAVTSDT calls module IEAVTSD1 to obtain storage and to copy the dump header record into this storage. IEAVTSD1 posts IEAVTDSV and passes a copy of the dump header record. IEAVTDSV begins post-dump exit processing. IEAVTSD1 then returns to IEAVTSDT.	IEAVTSD1		b. If a suspend or disabled summary dump was taken for a branch entry to SVC dump, IEAVTSDT calls module IEAVTSDW to write out the summary dump information on the SYS1.DUMPxx dump data set.	IEAVTSDW	
2 If the tasks of the requested address space are dispatchable:			c. If the system is not to be quiesced, IEAVTSDT calls STATUS to reset the system dispatchable. IEAVTSDT also tells system resource management (SRM) that the address space can be swapped by issuing a SYSEVENT=OKSWAP. IEAVTSDT also allows this task to accept asynchronous exits.		
a. IEAVTSDT calls STATUS to set the system and the tasks of the address space to be non-dispatchable.			d. If global storage areas (such as NUC, SQA, CSA, LISTA storage ranges, etc.) were requested and were not dumped by the resident dump task previously, IEAVTSDT calls module IEAVTSDG. IEAVTSDG writes the requested global storage areas on the dump data set. If SVC dump set the system non-dispatchable, IEAVTSDT sets the system dispatchable.	IEAVTSDG	
b. IEAVTSDT calls IEAVTSDB to establish a system non-dispatchability timer DIE to protect the system and tasks of this address space from being left non-dispatchable due to an unrecoverable error in SVC Dump.					
3 This ENQ on the SDUMP resource keeps more than one address space of a multiple address space SVC dump from dumping to the same dump data set at the same time. The QNAME used for this ENQ is SYSIEA01. The RNAME used for this ENQ is SDUMPENQ. The resident dump task in each address space performs only the part of the SVC dump processing that relates to that address space.	IEAVTSDT	SDTREQ			
An ESTAE recovery exit is established in the dump task for the address space when the address space is created. This recovery routine is located in the IEAVTSDT (label DTESTAE1) object module and is used as backup for the main ESTAE.		DTESTAE1			
IEAVTSDT establishes an ESTAE recovery exit before the SVC dump processing is started in the address space. This recovery routine is located in the IEAVAD00 (label SDESTAEX) CSECT and is used for all of SVC dump processing.					

IEAVTSDT - Resident Dump Task Processing (IEAVTSDT) (Part 3 of 6)

DMP-118 MVS/XA SLL: Dumping Ser

LY28-1670-1

(c) Copyright IBM Corp. 1987, 1989



IEAVTSDT – Resident Dump Task Processing (IEAVTSDT) (Part 4 of 6)

Extended Description	Module	Label	Extended Description	Module	Label
<p>e. IEAVTSDT calls IEAVTSDL for every address space requested. If they are available, IEAVTSDL dumps the default areas (authorization table, address space second table, linkage table, and entry tables) for all address spaces. If local areas were requested (for example SWA, LSQA, RGN), IEAVTSDL dumps these areas. For LIST and LISTA, IEAVTSDL obtains the address of the storage areas from the RTSD and writes the local storage areas on the dump data set. For SUBPLST, IEAVTSDL obtains the local subpools, requested for each address space, from the RTSD. For the master address space, IEAVTSDL only dumps the local storage areas if the master address space was a requested address space.</p>	IEAVTSDL		<p>h. If the TRT option is specified and GTF is active, IEAVTSDT calls IEAVTSDF to write the GTF trace records on the dump data set. These trace records are the last things written on the dump data set prior to the end-of-file mark. For multiple address space dumps, the resident dump task in the last address space to be dumped is the only one that calls module IEAVTSDF.</p>	IEAVTSDF	
<p>f. If a summary dump was requested for an SVC entry to SVC dump, IEAVTSDT calls module IEAVTSSE to move the following areas to the dump data set:</p> <ul style="list-style-type: none"> ● ASID, jobname, and stepname. ● Storage addresses specified with the SUMLIST parameter on the SDUMP macro. ● Any RTM2 work area (RTM2WA) associated with the failing task, 2K of storage before and after every valid and unique address in the program status word, and the registers for the failure that were saved in each RTM2WA. 	IEAVTSSE	IEAVTSDP			
					RTM2P
<p>g. IEAVTSDT calls module IEAVTSDU to determine if any component-related data is to be included in the dump. If the trace table option (TRT) is specified on the SDUMP macro, IEAVTSDU loads and calls the master trace dump exit (IEEMB879). Otherwise, processing continues at step i. IEEMB879 determines if the master trace is active and, if it is, invokes the SDUMP output service to write the master trace data on the dump data set. (See the diagram and extended description of IEEMB879.) If the master trace is not active, IEEMB879 returns to IEAVTSDU. When multiple address spaces are being dumped, the resident dump task in the last address space to be dumped is the only one that calls module IEAVTSDU.</p>	IEAVTSDU				
				IEEMB879	

IEAVTSDT -- Resident Dump Task Processing (IEAVTSDT) (Part 5 of 6)

No diagram.

Extended Description continued on next page.

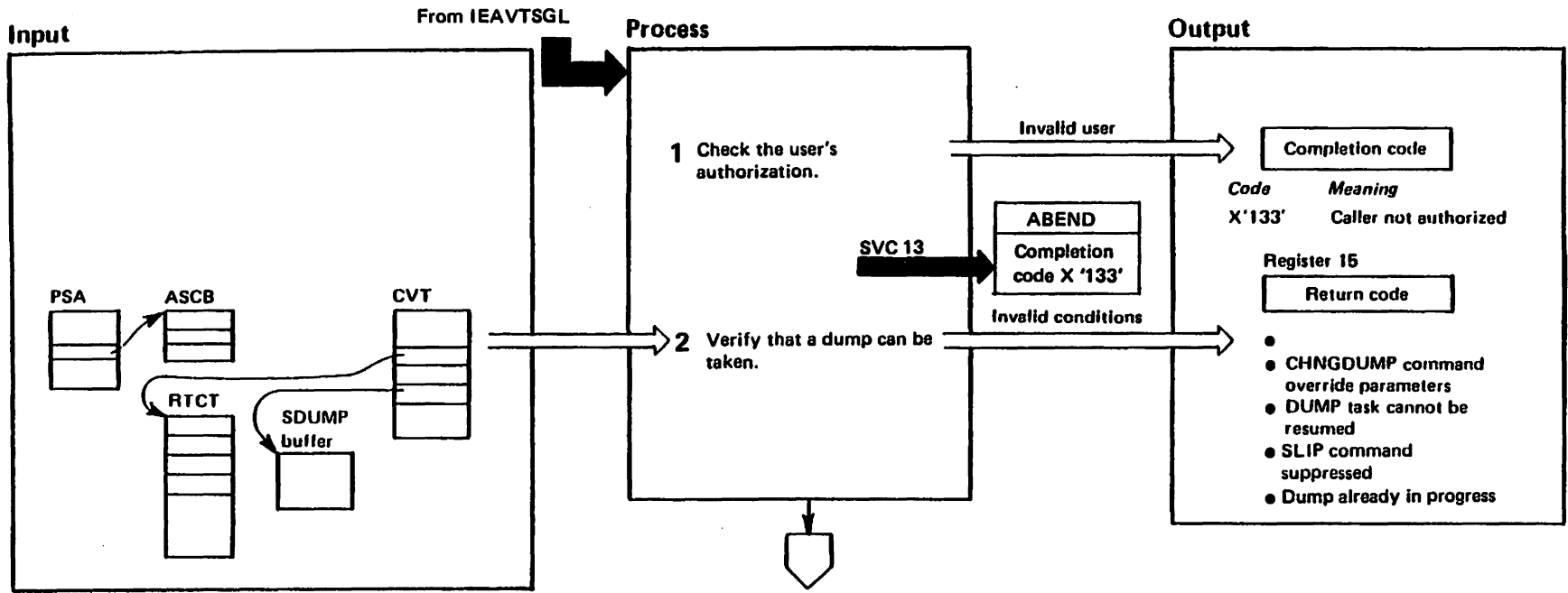
DMP-120 MVS/XA SLL: Dumping Ser

LY28-1670-1 (c) Copyright IBM Corp. 1987, 1989

IEAVTSDT - Resident Dump Task Processing (IEAVTSDT) (Part 6 of 6)

Extended Description	Module	Label	Extended Description	Module	Label
<p>i. IEAVTSDT calls IEAVTSDC to clean up SVC dump resources when necessary. All of the tasks in the address space that were set non-dispatchable are reset dispatchable and the system is set dispatchable. If all other address space dumps are complete, IEAVTSDC issues message IEA911E to the operator. If this SVC dump filled the last available dump data set, it issues message IEA994E or IEA994A to the operator. If data sets are available, it deletes operator message IEA994E. If tape dump data sets are being used, it issues message IEA984E to inform the operator to mount a new tape. An end-of-file is written on the dump data set. IEAVTSDC page-frees any modules or control blocks which were page-fixed by SDUMP and frees the system trace tables if obtained by SDUMP. It initializes the acquired areas for post exit processing and enqueues them on the post exit processing queue. IEAVTSDC also frees resources acquired by SVC dump. If other address space dumps are active, IEAVTSDC does not perform resource cleanup. IEAVTSDC calls IEAVTSD1 to initiate post exit processing with the following steps:</p> <ul style="list-style-type: none"> ● Obtains storage for the post-dump exit processing (PDEP) queue element. ● Copies the dump header record saved by IEAVTSDH and the DAE pre-dump/post-dump parameter list (DSPD) into the PDEP. ● Enqueues on the post dump exit resource (major name: IEASYS01, minor name: SDPOSTEX). ● Adds the PDEP to the post-dump exit queue. ● Posts IEAVTDSV to cause IEAVTDSV to perform post exit processing. See the module description for IEAVTDSV and IEAVTSEP for further details on post dump exit processing. The post exit processing continues after the dump has completed. ● Dequeues from the post exit resource. ● Returns to IEAVTSDC. 	IEAVTSDC		<p>j. If no other address space dumps are active and the caller supplied an ECB, IEAVTSDT posts the ECB with the appropriate return code.</p> <p>k. IEAVTSDT releases the SDUMP resource using a DEQ macro.</p> <p>4 IEAVTSDT uses the branch-entry WAIT service to cause the dump task to logically wait for the next dump request. The SRB scheduled by module IEAVTSDX uses the branch-entry POST service to reactivate the dump task to process the next dump request.</p>	IEAVTSDT	

IEAVTSDX - Schedule Dump Processing (Part 1 of 8)



IEAVTSDX - Schedule Dump Processing (Part 2 of 8)

Extended Description	Module	Label
----------------------	--------	-------

Schedule dump processing (IEAVTSDX) receives control from an interface module (IEAVTSGI). IEAVTSGI saves the caller's registers and addressing mode before passing control to IEAVTSDX in 31-bit addressing mode. This interface module is used to maintain downward compatibility with branch entry SDUMP users running in 24-bit addressing mode.

Schedule dump processing receives control from SVC 51 (IEAVAD00) when the ASID, ASIDLST, TYPE=XMEM/XMEME or LISTA keywords and the BRANCH=NO keyword have been specified on the SDUMP macro or when BRANCH=NO and SUBPLST have been specified and at least one of the ASIDs in the subpool list is different from the current ASID in control.

Schedule dump processing also receives control as a branch entry when the BRANCH=YES keyword has been specified on the SDUMP macro.

1 IEAVTSDX verifies the authorization of the caller.

Callers of schedule dump processing must be in supervisor state, key 0 and have at least one of the following attributes:

- Be running disabled with a PSASUPER bit set
- Hold any lock (including the CML lock)
- Be in SRB mode
- Be in cross memory mode
- Be in EUT FRR mode

IEAVTSDX gives callers who are not in supervisor state a system X'0C2' abend. IEAVTSDX gives callers who are not key 0 a system X'0C4' abend. IEAVTSDX gives callers who are not running in the proper mode a X'133' abend completion code.

SDXAUTH

Extended Description	Module	Label
----------------------	--------	-------

2 IEAVTSDX establishes an FRR and initializes the FRR work area. IEAVTSDX then:

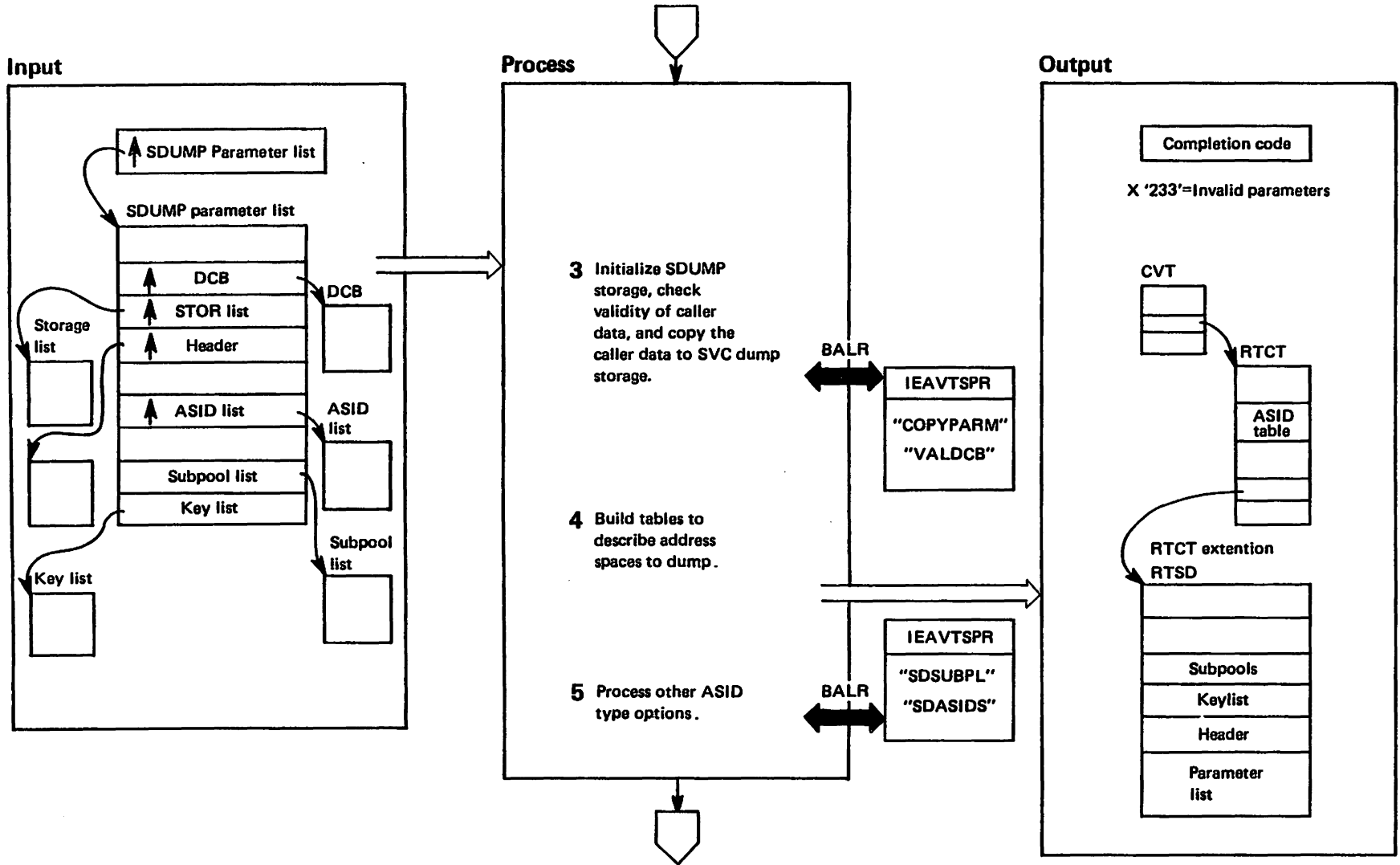
- Checks the RTCTSDPL field in the RTCT and the high order bit of the CVTSDBF field to see if an SVC DUMP is already in progress.
- Checks the RTCTISVC field to see if a CHNGDUMP SET, SDUMP, NODUMP command has been issued, causing no SVC DUMPs to be taken.
- Checks the RTCT and the SDUMP parameter list to see that a dump data set exists.
- Checks any current RTM control blocks (RTM1WA, RTM2WA) to see if a caller has issued a SLIP NODUMP command.

If any of the above validity checks fail, IEAVTSDX deletes the FRR and returns control to the caller with a return code of 8 in register 15. The reason for a return code of 8 is placed in the caller's SDWA in SDWASDRD.

SDXAUTH

IEAVTSDX - Schedule Dump Processing (Part 3 of 8)

DMP-124 MVS/XA SLL: Dumping Ser LY28-1670-1 (c) Copyright IBM Corp. 1987, 1989



"Restricted Materials of IBM"
Licensed Materials - Property of IBM

IEAVTSDX - Schedule Dump Processing (Part 4 of 8)

Extended Description	Module	Label
<p>3 Information describing the caller's cross memory state is saved in the RTSD. IEAVTSDX calls the common SDUMP parameter list routine (IEAVTSPR) for COPYPARM function, which does the following:</p>	IEAVTSPR	POSSIBLE

- Clears the RTSD area used to hold the caller's parameter list.
- Copies the user parameter list into the RTSD
- Copies the dump header to the RTSD
- Copies the storage list to the RTSD
- Determines the dump options to be used from the combination of the CHNGDUMP options and the SDUMP parameter list.

If the caller specified the DCB option, IEAVTSDX calls IEAVTSPR for the VALDCB function, which validates the DCB. The DCB must be OPEN and point to a device supported by SDUMP.

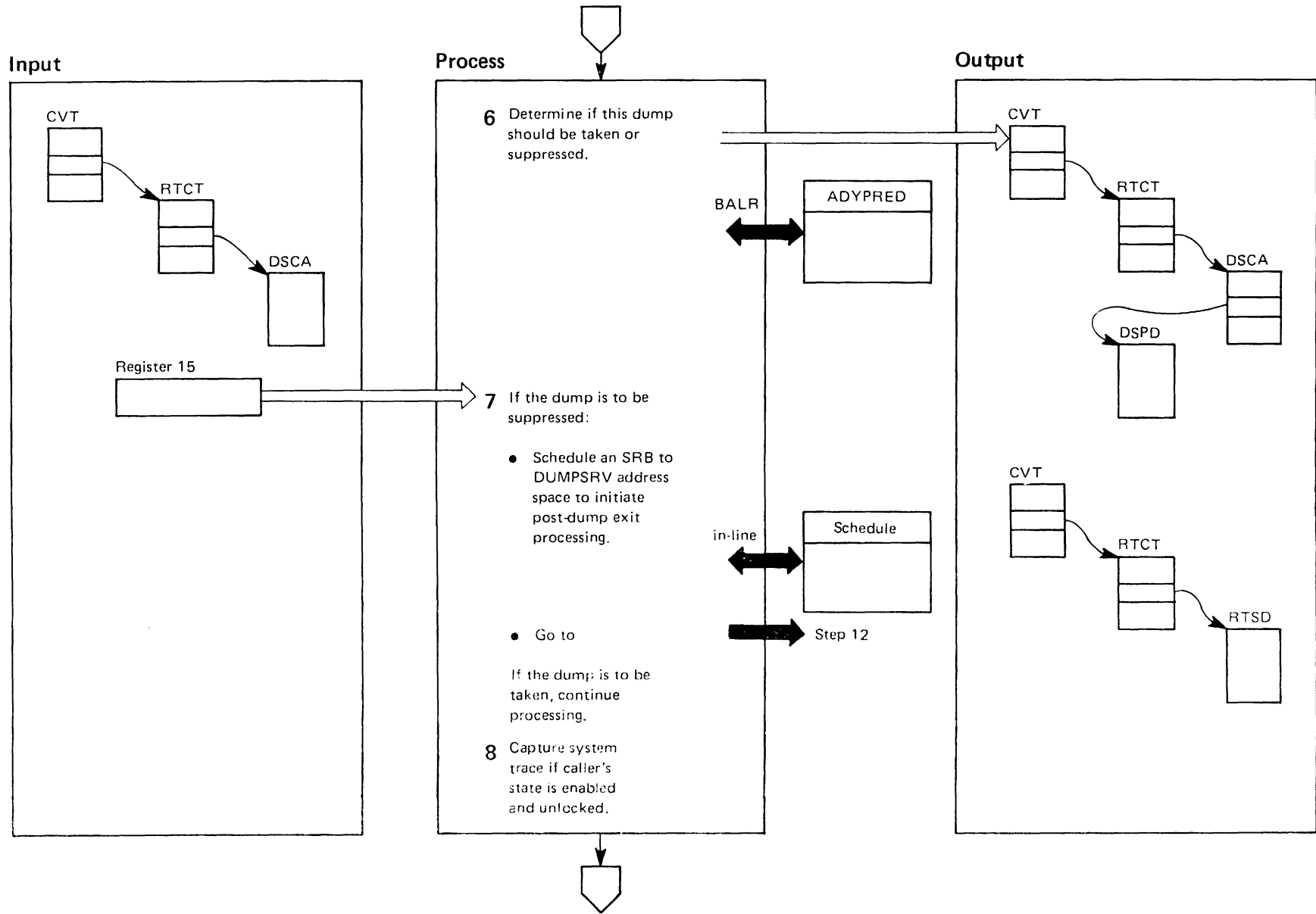
If any of the above user data is invalid, the caller is abended with an X '233' and an appropriate reason code.

4 If the TYPE=XMEME keyword is specified, IEAVTSDX locates the RTM control blocks related to the error (RTM1 or RTM2 workarea). The PASID, SASID, and HASID at the time of error are stored in the RTCT ASID table. If the TYPE=XMEM keyword is specified or if the TYPE=XMEME processing fails because no current valid RTM control blocks could be found, the current PASID, SASID, HASID, and CML (if a CML lock is held) ASIDs are moved to the RTCT ASID table. The RTCT ASID table has a limit of 16 unique ASIDs. Duplicate ASID entries are not made.

Extended Description	Module	Label
<p>5 If the SUBPLST keyword was specified, IEAVTSDX calls IEAVTSPR IEAVTSPR for the SDSUBPL function. The SDSUBPL function divides the subpool list into a global subpool list and a local subpool list for each specified ASID. The SDSUBPL function adds new ASIDs to the RTCT ASID table, and copies the KEYLIST (if specified) into the RTSD.</p>	IEAVTSPR	

IEAVTSPR is always called to process the SDASIDS function. This function analyzes the LISTA and ASIDLST or ASID keywords. New ASIDs are added to the RTCT ASID table, and the LISTA storage ranges are copied to the RTSD.

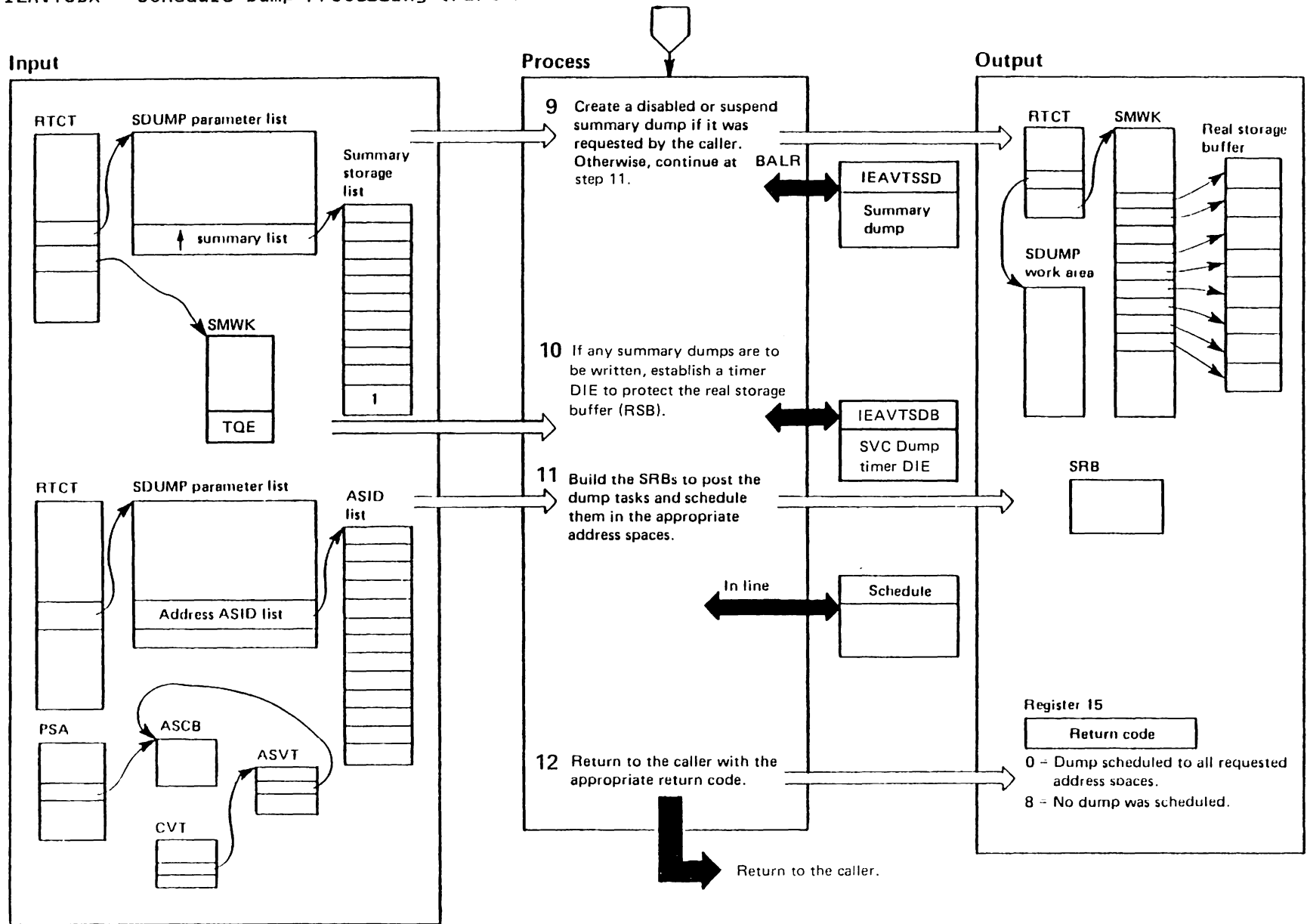
IEAVTSDX - Schedule Dump Processing (Part 5 of 8)



IEAVTSDX - Schedule Dump Processing (Part 6 of 8)

Extended Description	Module	Label
<p>6 IEAVTSDX initializes the pre-dump parameter list (DSPD). IEAVTSDX passes the DSPD to ADYPRED, the DAE pre-dump processor. ADYPRED determines if this dump should be suppressed and builds the DAE portion of the dump header record. When ADYPRED returns control, IEAVTSDX checks the return code in register 15 to determine if the dump should be suppressed.</p>	ADYPRED	
<p>7 If the dump should be suppressed, IEAVTSDX saves the dump suppression status flags and the no dump reason code (X'0B' in SDWASDRC) in the caller's SDWA. IEAVTSDX also saves the no-dump reason code in RTSDSDRC. (IEAVTSDT later checks this field.) IEAVTSDX then schedules an SRB to the DUMPSRV address space to post the dump task (IEAVTSDT) which initiates post-dump exit processing. (See the idagram for IEAVTSDT for more information about post-dump exit processing.) Once the SRB has been scheduled, IEAVTSDX sets the no-dump return code (X'08') in register 15 and returns to the caller.</p>		
<p>8 If a system trace was requested, a test is made to see if the caller is in an unlocked, enabled state and did not specify TYPE=NOLOCAL on the SDUMP macro. If so, IEAVTSDX attempts to capture the trace before scheduling an SRB to initiate the dump. If the caller is in task mode and a trace table is pointed to by the RTM2 work area, this trace table is used. Otherwise, IEAVTSDX issues the SNAPTRC macro to capture the trace table.</p>	IEAVTSDX SDXTRACE	
		IEAVTSDX SDXSNAPT

IEAVTSDX - Schedule Dump Processing (Part 7 of 8)



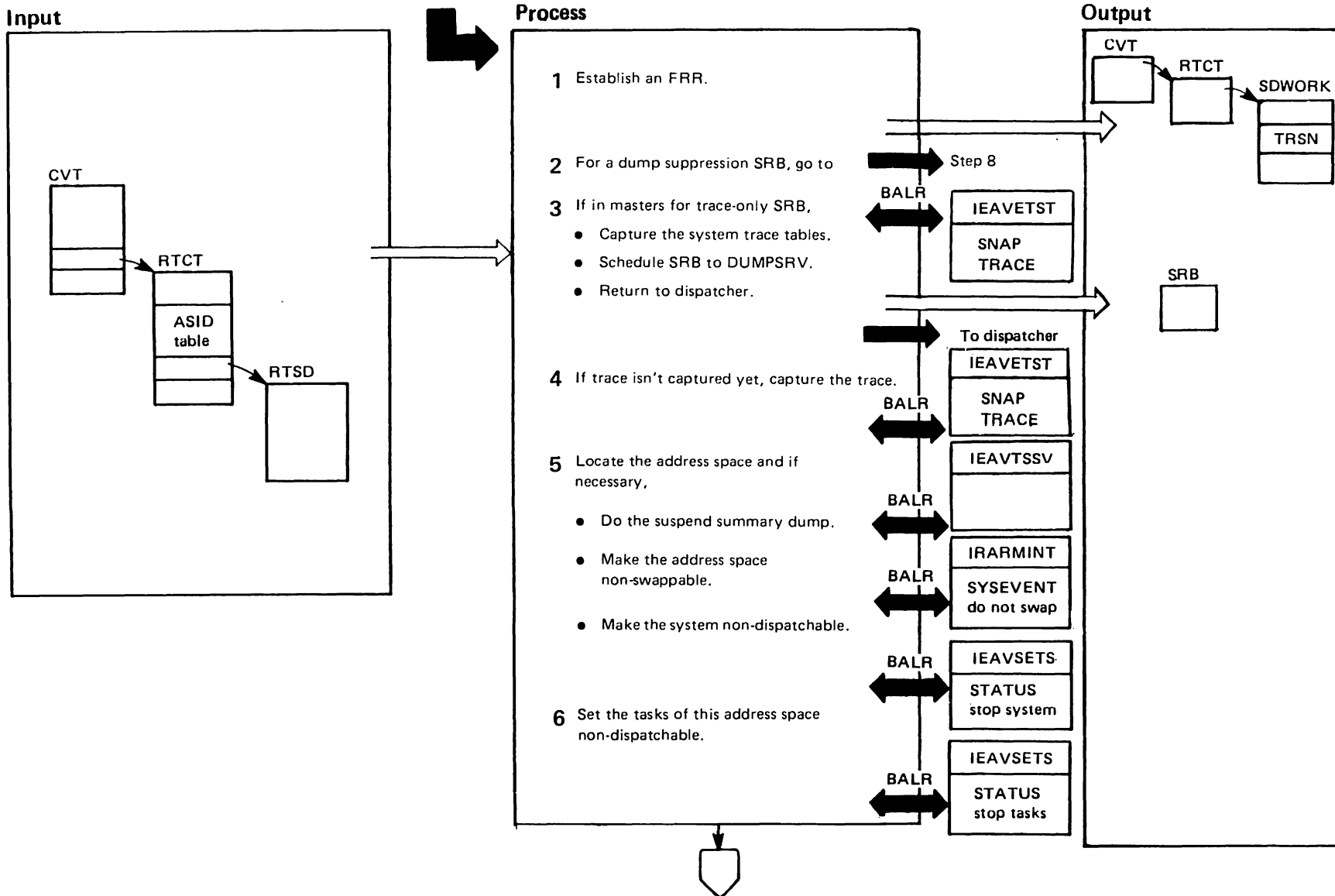
IEAVTSDX - Schedule Dump Processing (Part 8 of 8)

Extended Description	Module	Label
<p>9 If a caller has requested a disabled or suspend summary dump (SDATA=SUMDUMP/SUM with BRANCH=YES), IEAVTSDX calls module IEAVTSSD. See the M.O. diagram Summary Dump Processing (IEAVTSSD) for the details of IEAVTSSD's processing.</p>	IEAVTSSD	
<p>10 For disabled summary dump processing, IEAVTSDX calls IEAVTSDB to establish a timer DIE to protect the real storage buffer (RSB) if IEAVTSDX holds at least one summary dump real storage buffer.</p>	IEAVTSDX	
<p>11 If the caller requested a suspend summary dump (SDATA=SUM, SUSPEND=YES specified), IEAVTSDX performs no SRB scheduling at this time. The summary dump process has scheduled all needed SRBs. See the M.O. diagram Summary Dump Processing (IEAVTSSD) for the details of this SRB scheduling process.</p>		SCHBDSRB

Extended Description	Module	Label
<p>If the caller did not request a suspend summary dump (SUSPEND=NO specified or defaulted to on the SDUMP macro), IEAVTSDX builds and schedules a global SRB.</p> <ul style="list-style-type: none"> If the RTCT ASID table contains more than one ASID from the keywords TYPE=XMEM/XMEME, LISTA, ASID, SUBPLST, or ASIDLST, and the caller requested at least one global storage area (NUC, SQA, CSA, PSA, ALLPSA, TRT, SUMDUMP), IEAVTSDX, builds and schedules the global SRB to the master address space (ASID 1). If the RTCT ASID table contains only one ASID, IEAVTSDX builds and schedules an SRB to the address space for which IEAVTSDX saved an ASID in the RTCT ASID table. The maximum number of SRBs scheduled is 15. 		
<p>12 After IEAVTSDX has scheduled all of the necessary SRB(s), it returns control to the caller with register 15 containing a return code.</p>		

IEAVTSDX - Schedule Dump SRB Processing (Part 1 of 4)

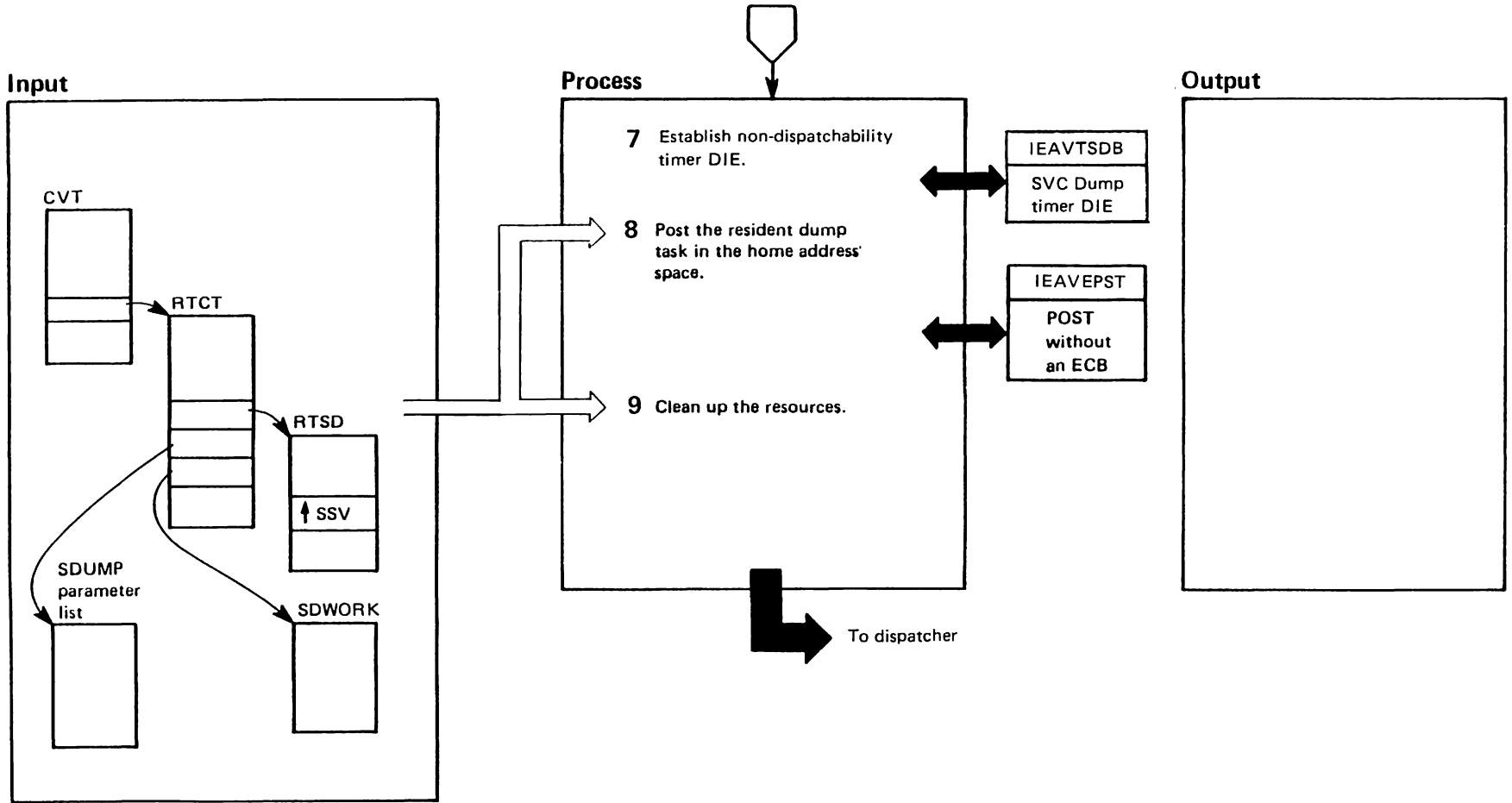
From the dispatcher
as a global process SRB



IEAVTSDX - Schedule Dump SRB Processing (Part 2 of 4)

Extended Description	Module	Label	Extended Description	Module	Label
<p>All SVC dump requests that specify BRANCH=YES, LISTA, SUBPLST, ASID, ASIDLST, TYPE=XMEM, or TYPE=XMEME cause schedule dump SRB processing to receive control. This routine prepares the system and the requested address space for an SVC dump and starts the scheduled SVC dump process by posting the resident dump task (IEAVTSDT). In some cases, IEAVTSDX invokes suspend summary dump processing (IEAVTSSV). This routine can run in any address space in the system. IEAVTSDX also receives control for dump suppression processing in either the DUMPSRV or the master scheduler address space. If the request is for a dump that is to be suppressed, IEAVTSDX posts the resident dump task (IEAVTSDT) which starts post-dump exit processing.</p>			<ul style="list-style-type: none"> The caller requested that the CSA and SOA be dumped and did not specify QUIESCE=NO. <p>In these cases IEAVTSDX calls IRARMINT to make the current address space non-swappable. IEAVTSDX calls STATUS to set the system non-dispatchable. When control returns from STATUS, only the exempt address spaces in the system will be dispatchable.</p> <p>6 If this address space is one of the requested address spaces to be dumped (it was specified on the ASID, SUBPLST, ASIDLST, LISTA, TYPE=XMEM, or TYPE=XMEME keywords), IEAVTSDX calls STATUS to set all of the tasks in this address space with a lower priority than the dump task non-dispatchable. This is necessary before the system can issue a dump of this address space. Note: the region control task, communications task and the dump task are never set non-dispatchable.</p>	IRARMINT	
				IEAVSETS	
	IEAVTSDT			IEAVTSDX	SCHSRB
				IEAVSETS	
<p>1 IEAVTSDX establishes an FRR. This FRR resides in module IEAVTSDR.</p> <p>2 If this SRB is for the suppression of a dump, processing is continued with step 8.</p> <p>3 If a suspend summary dump was requested and the trace table has not yet been captured, the first SRB to run is in the master address space. The purpose of sending the SRB to masters is to capture the system trace as soon as possible. The trace is captured using the SNAPTRC macro which invokes module IEAVETST. After capturing the trace, an SRB is scheduled to DUMPSRV and the SRB exits to the dispatcher.</p> <p>4 If the trace has not been captured yet, IEAVTSDX issues the SNAPTRC macro to capture the system trace. The SNAPTRC macro calls module IEAVETST.</p> <p>5 IEAVTSDX locates the requested address space in the RTCT ASID table and, if it is in the DUMPSRV address space and SUSPEND=YES was a specified option on the SDUMP macro, calls IEAVTSSV to perform the suspend summary dump. Under some conditions, IEAVTSDX needs to call IRARMINT and IEAVSETS. These conditions are:</p> <ul style="list-style-type: none"> The caller specified that a summary dump must be written and the SRB is running in the master address space, the DUMPSRV address space, or the only requested address space. 					

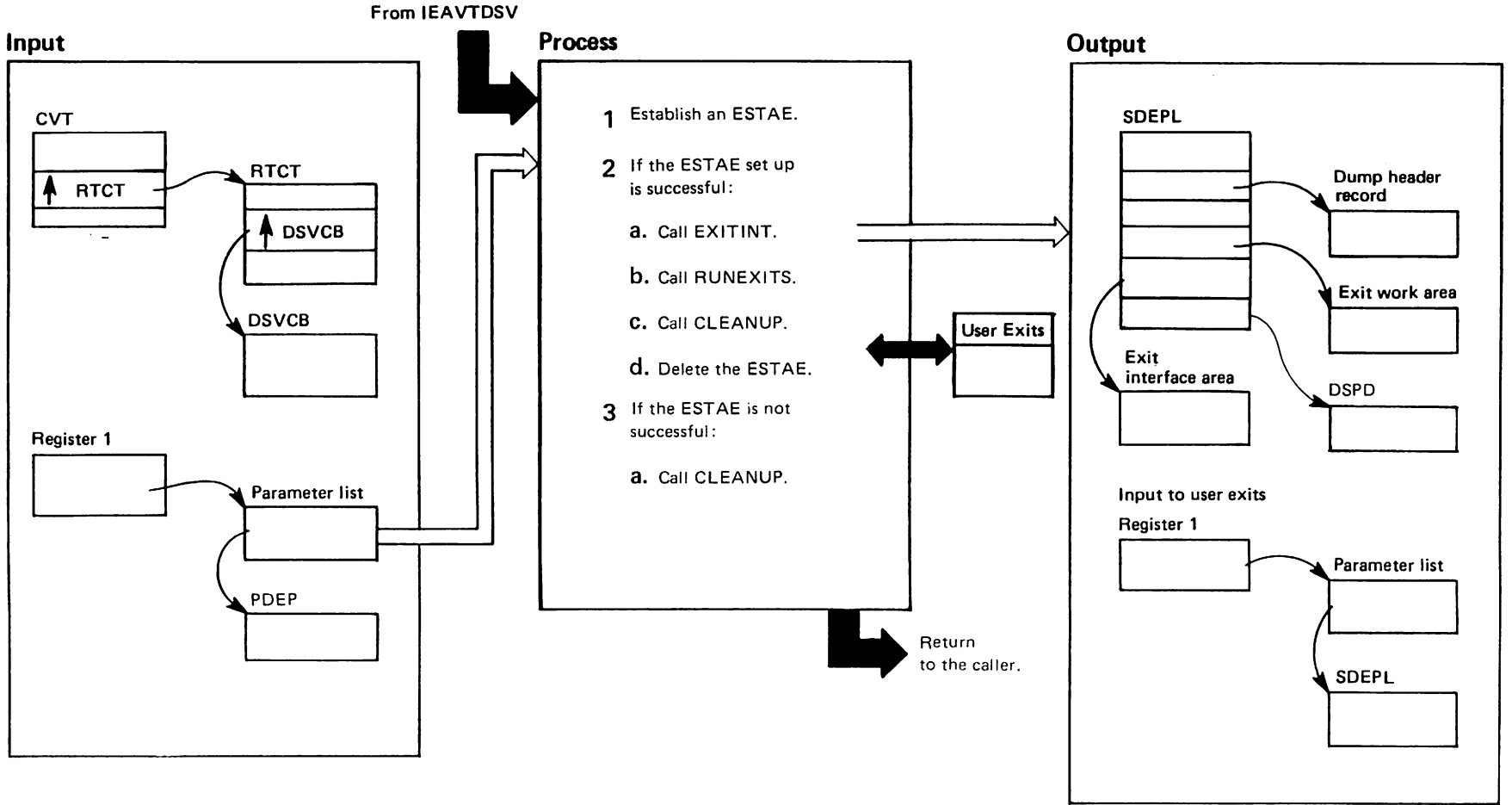
IEAVTSDX - Schedule Dump SRB Processing (Part 3 of 4)



IEAVTSDX - Schedule Dump SRB Processing (Part 4 of 4)

Extended Description	Module	Label
<p>7 IEAVTSDX calls IEAVTSDB to establish the SVC Dump non-dispatchability timer DIE to protect the system and tasks of this address space from being left non-dispatchable due to an unrecoverable error in SVC Dump.</p>		
<p>8 IEAVTSDX uses the POST service (without an ECB) to post the resident dump task in the current address space. IEAVTSDX posts IEAVTSDT to start the dump process or to start the post-dump exit process if the dump is being suppressed. IEAVTSDX sets indicators in the RTCT ASID table entry for this address space to show that the schedule dump SRB routine received control and posted the resident dump task.</p>	IEAVEPST	
<p>9 IEAVTSDX uses SETLOCK to release the local lock and SETFRR to delete the FRR from the current FRR stack. IEAVTSDX returns control to the dispatcher using the return address passed as input in register 14.</p>		

IEAVTSEP - SVC Post Exit Processor (Part 1 of 2)



IEAVTSEP - SVC Post Exit Processor (Part 2 of 2)

Extended Description	Module	Label
----------------------	--------	-------

IEAVTSEP is attached as a subtask of the Job Step Task (IEAVTDSV) in the DUMPSRV address space. Its function is to process the post-dump exits.	IEAVTSEP	
---	----------	--

1 IEAVTSEP sets up an ESTAE recovery routine to handle errors in IEAVTSEP and the exits. Errors in IEAVTSEP cause an SDUMP to be taken. The ESTAE routine records errors in the exits with the name of the exit placed in the SDWA. The ESTAE routine does not take a dump for exit errors; the exits must provide their own recovery routine for an SDUMP.

2 IEAVTSEP checks the value in register 15 for a successful ESTAE setup. If it is equal to zero, the setup was successful.

a. EXITINIT obtains the data areas required by the post-dump exits and builds the parameter list (SDEPL) with pointers to the data areas obtained.

b. The RUNEXITS routine calls the user exits from the SDUMP exit list (IEAVTSEL). Failures in an exit routine are recorded and control passes to the next exit in the list. The DAE post-dump exit routine, ADYPSTD, is the last exit to be called.

c. The CLEANUP routine executes only once. It cleans up all of the obtained storage and terminates. It frees storage for the exit work area, exit interface area, and the copy of the dump header record.

d. IEAVTSEP deletes the ESTAE.

3 If the ESTAE could not be established, the input storage is freed.

IEAVTSRB - MODULE DESCRIPTION

DESCRIPTIVE NAME: SVC Dump SRB Routine

FUNCTION:

IEAVTSRB is the SRB scheduled by IEAVTSDX and by IEAVTSSD (for Suspend summary dump requests). It has three mutually exclusive purposes:

The first purpose is to initiate the dumping process. The SRB may call IEAVTSSV to perform Suspend summary dump processing and then issue SNAPTRC if the system trace tables have not been captured yet for this dump. The SRB will set the system and address space environment so that a dump can be taken by setting the system and/or tasks of the address space non-dispatchable. The SRB will then activate the resident dump task to dump storage in this address space.

The second purpose is to post the dump task to start off post exit processing for a requested dump which has been suppressed by Dump Analysis and Elimination (DAE).

The third purpose is to perform as a special trace SRB in masters address space. This occurs when the caller of SVC DUMP is requesting a Suspend summary dump but is not in an enabled unlocked state and therefore the system trace tables could not be captured earlier. The SRB issues the SNAPTRC macro to get a snapshot of the system trace tables and then schedules an SRB to the DUMPSRV address space to initiate the dumping process.

ENTRY POINT: IEAVTSRB

PURPOSE:

IEAVTSRB is the SRB scheduled by IEAVTSDX and by IEAVTSSD (for Suspend summary dump requests). It has three mutually exclusive purposes:

The first purpose is to initiate the dumping process. The SRB may call IEAVTSSV to perform Suspend summary dump processing and then issue SNAPTRC if the system trace tables have not been captured yet for this dump. The SRB will set the system and address space environment so that a dump can be taken by setting the system and/or tasks of the address space non-dispatchable. The SRB will then activate the resident dump task.

The second purpose is to post the dump task to start off post exit processing for a requested dump which has been suppressed by the dump suppression component.

The third purpose is to perform as a special trace SRB. The SRB issues the SNAPTRC macro to get a snapshot of the system trace tables and then schedules an SRB to DUMPSRV.

LINKAGE: LPSW

CALLERS: None

INPUT: SVC Dump control blocks

OUTPUT:

The system and/or address space has been set non-dispatchable and the dump task has been posted.

ENTRY POINT: SRBFRR2

PURPOSE: Recover from a failure in the SVC Dump SRB Routine.

LINKAGE: LPSW

CALLERS: None

IEAVTSRB - MODULE DESCRIPTION (Continued)

INPUT:

SVC Dump control blocks
 FRR 6 word work area

OUTPUT:

Request RTM to record.
 Request RTM to release the local lock.
 Percolate to SRBFRR1.

EXIT NORMAL: Return to IEAVTRTS.

ENTRY POINT: SRBFRR1

PURPOSE: Recover from a failure in the SVC Dump SRB Routine.

LINKAGE: LPSW

CALLERS: None

INPUT:

SVC Dump control blocks
 FRR 6 word work area

OUTPUT:

The system and/or address space has been set
 dispatchable and the user has been posted.
 Retry, if allowed, requested.

EXIT NORMAL: Return to IEAVTRTS.

EXTERNAL REFERENCES:

ROUTINES:

- IEAVPT00 - Post routine, awaken dump task and
awaken caller.
- IEAVESRT - Reset the stopped SDUMP caller
- IEAVTSDB - Establish timer dies to protect critical
system resources (i.e. the real storage
buffer and system dispatchability).
- IEAVTSSV - Move Suspend summary dump data to
the SVC Dump virtual storage buffer
- IARXPLCK - Obtain RSM serialization
- IARXPRSB - Free real storage frames and free
RSM serialization.

CONTROL BLOCKS:

Common name	COMMON Macro ID	LIBRARY Usage	Function
ASCB	IHAASCB	read	Checks if the address space has failed and whether the dump task is active.
ASVT	IHAASVT	read	Checks if SVC Dump caller is still active before posting the callers ECB.
ASXB	IHAASXB	read	Obtains address of top TCB in order to set all the tasks non-dispatchable/ dispatchable.
CVT	CVT	read and write	Unlocks the 4K SQA buffer, and uses the CVTBRET.
DSPD	ADYDSPD	read	(see ADYDSTAT).
DSTAT	ADYDSTAT	read	Puts DAE suppression flags into VRA in the FRR.
FRRS	IHAFRRS	read and write	Adds and deletes two FRRs.
OUCH	IRAUCB	read	Checks if DUMPSRV is going through swapout processing.

IEAVTSRB - MODULE DESCRIPTION (Continued)

PSA	IHAPSA	read	Obtains the current address space and adds/deletes FRRs.
PVT	IHAPVT	read	Obtains addresses of RSM routines.
RTCT	IHARTCT	read and write	Obtains address of DUMPSRVs ASCB, checks and updates the SVC Dump ASID table, checks and updates SVC Dumps Information/Flags section, checks the SDUMP parm list via RTCTSDPL and unlocks SVC Dump by clearing it in the FRR.
RTSD	IHARTSD	read and write	Sets and checks internal flags and uses save areas internally as well as for calling services
SDDAT	IHASDDAT	read and write	Obtains addresses of SVC Dump control blocks and clears SDDSDPL in the FRR.
SDDIE	IHASDDIE	read and write	Fills in the DIE control block prior to calling IEAVTSDB to establish the DIE.
SDDSQ	IHASDDSQ	write	Sets the serialization flag off to release the current SDDSQ entry in the FRR.
SDMOD	IHASDMOD	read	Obtains starting and ending addresses of SVC Dump modules.
SDWA	IHASDWA	read and write	Obtains the work area address and checks if retry is allowed and fills in diagnostic information to RECORD.
SDWORK	IHASDWRK	read and write	Checks if this SRB was entered as a dump suppression SRB and sets the SRB in error flag (in SDRSN) in the FRR.
SMMK	IHASMMK	read and write	Cleans-up RSB indicators and counters in the FRR, resets the Suspended SVC Dump caller in the FRR, checks if Suspend summary dump is active or in error, checks and sets off the flag indicating this is a special trace SRB, obtains the ASCB address of the local lock being held by the stopped unit of work, and obtains the address of the initialized SRB to DUMPSRV for Suspend summary dump.
SRB	IHASRB	read	(required by IHASDDIE).
SVT	IHASVT	read	(required by IHASMMK).
TCB	IKJTCB	read	
TQE	IHATQE	read	(required by IHASDDIE).
TRSN	IHATRSN	read and write	Fills in to request a snap trace, fills in to request SNAPTRC to free the system trace tables in the FRR, and cleans-up the trace indicators.

SERIALIZATION: Entered with the SVC Dump serialization.

IEAVTSRB - MODULE OPERATION

First, this routine will establish recovery.

If this is a special trace SRB to masters (see notes), it issues the SNAPTRC macro to get a snapshot of the system trace tables. If an SRB can be scheduled to DUMPSRV, IEAVTSRB schedules an SRB to DUMPSRV to initiate the dumping process.

If this SRB was scheduled to initiate the dumping process or was scheduled to do special trace processing but could not schedule the SRB to DUMPSRV, the following processing is done. If a Suspend summary dump is requested and this SRB is the first global SRB and is executing in the DUMPSRV address space, the SRB obtains the local lock and calls IEAVTSSV to complete the Suspend summary dump. If the system trace tables have not been captured yet, the SRB releases the local lock if held and issues the SNAPTRC macro which will capture the system trace tables. IEAVTSRB then obtains the local lock if not already held. If the dump task in this address space is dispatchable and the system should be set non-dispatchable, the SRB issues SYSEVENT to make this address space non-swappable, sets TCBFX to stop IRB scheduling, and calls STATUS to set the system non-dispatchable. The SRB calls IEAVTSDB to establish the Timer DIE which protects the system/tasks from being left non-dispatchable. The SRB calls STATUS to set the tasks (except Commtask and RCT) non-dispatchable. If the dump task in masters is non-dispatchable, the SRB calls STATUS to set it dispatchable. The SRB then calls IEAVPT00 to post the resident dump task and releases the local lock.

If this SRB was scheduled to initiate post exit processing for a requested dump which has been suppressed by DAE, the SRB obtains the local lock, posts the resident (masters or DUMPSRVs) dump task, and then releases the local lock.

IEAVTSRB deletes its recovery and returns to the dispatcher.

IEAVTSRB - DIAGNOSTIC AIDS

ENTRY POINT NAMES: IEAVTSRB
SRBFRR2
SRBFRR1

MESSAGES: None

ABEND CODES: None

WAIT STATE CODES: None

RETURN CODES:

ENTRY POINT IEAVTSRB: None

ENTRY POINT SRBFRR2:

EXIT NORMAL:

SDWARCDE = 0 (percolate)

ENTRY POINT SRBFRR1:

EXIT NORMAL:

SDWARCDE = 0 (percolate) if SDWACLUP is on
or no retry address (FRR6WRT=0)
4 (retry) otherwise

REGISTER CONTENTS ON ENTRY:

ENTRY POINT IEAVTSRB:

Reg 0 - address of SRB
Reg 1 - contents of SRBPARM field
Reg 2-13 - irrelevant
Reg 14 - return address
Reg 15 - entry point

ENTRY POINT SRBFRR2:

Reg 0 - address of 200 byte FRR work area
Reg 1 - address of SDWA
Reg 2-13 - irrelevant
Reg 14 - return address
Reg 15 - entry point

ENTRY POINT SRBFRR1:

Reg 0 - address of 200 byte FRR work area
Reg 1 - address of SDWA
Reg 2-13 - irrelevant
Reg 14 - return address
Reg 15 - entry point

REGISTER CONTENTS ON EXIT:

ENTRY POINT IEAVTSRB:

Reg 0-13 - irrelevant
Reg 14 - return address

IEAVTSRB - DIAGNOSTIC AIDS (Continued)

Reg 15 - irrelevant

ENTRY POINT SRBFRR2:

EXIT NORMAL:

Reg 0-13 - irrelevant
Reg 14 - return address
Reg 15 - irrelevant

ENTRY POINT SRBFRR1:

EXIT NORMAL:

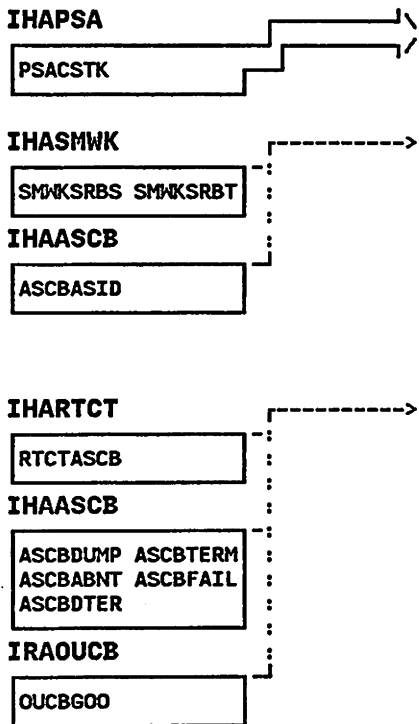
Reg 0-13 - irrelevant
Reg 14 - return address
Reg 15 - irrelevant

IEAVTSRB - SVC Dump SRB Routine

STEP 01



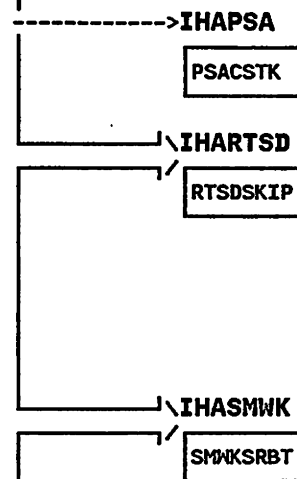
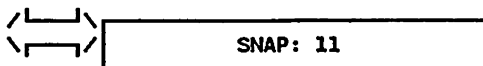
IEAVTSRB is the SRB scheduled by IEAVTSDX and by IEAVTSSD (for Suspend summary dump requests). It has three mutually exclusive purposes: The first purpose is to initiate the dumping process. The SRB may call IEAVTSSV to perform Suspend summary dump processing and then issue SNAPTRC if the system trace tables have not been captured yet for this dump. The SRB will set the system and address space environment so that a dump can be taken by setting the system and/or tasks of the address space non-dispatchable. The SRB will then activate the resident dump task to dump storage in this address space. The second purpose is to post the dump task to start off post exit processing for a requested dump which has been suppressed by Dump Analysis and Elimination (DAE). The third purpose is to perform as a special trace SRB in masters address space. This occurs when the caller of SVC DUMP is requesting a Suspend summary dump but is not in an enabled unlocked state and therefore the system trace tables could not be captured earlier. The SRB issues the SNAPTRC macro to get a snapshot of the system trace tables and then schedules an SRB to the DUMPSRV address space to initiate the dumping process.



01 Establishes two recovery exits (SRBFRR1 and SRBFRR2).

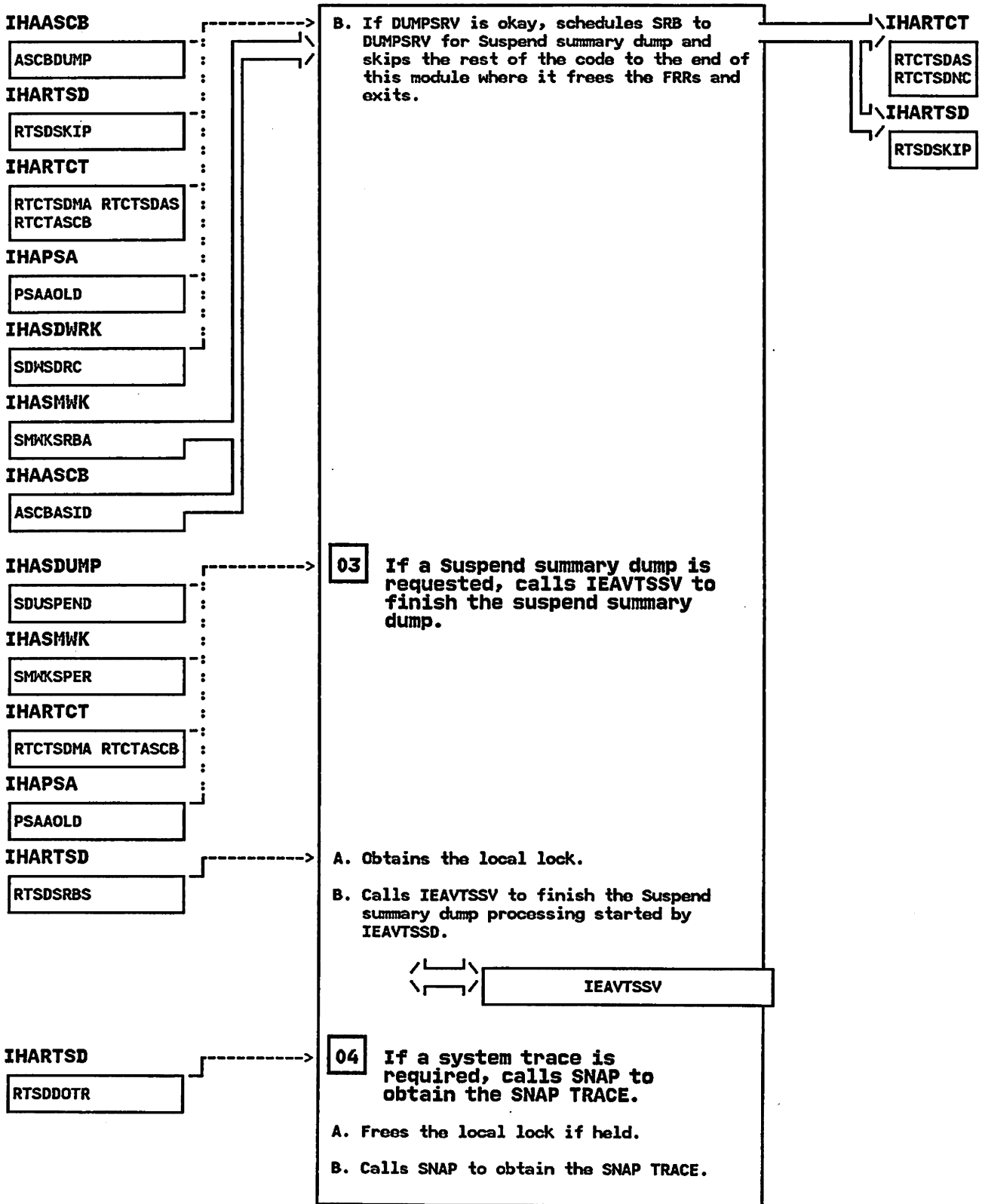
02 If this is a special trace SRB, performs the SNAPTRACE function and schedules the SRB to DUMPSRV.

A. Calls SNAP to obtain the SNAP TRACE.



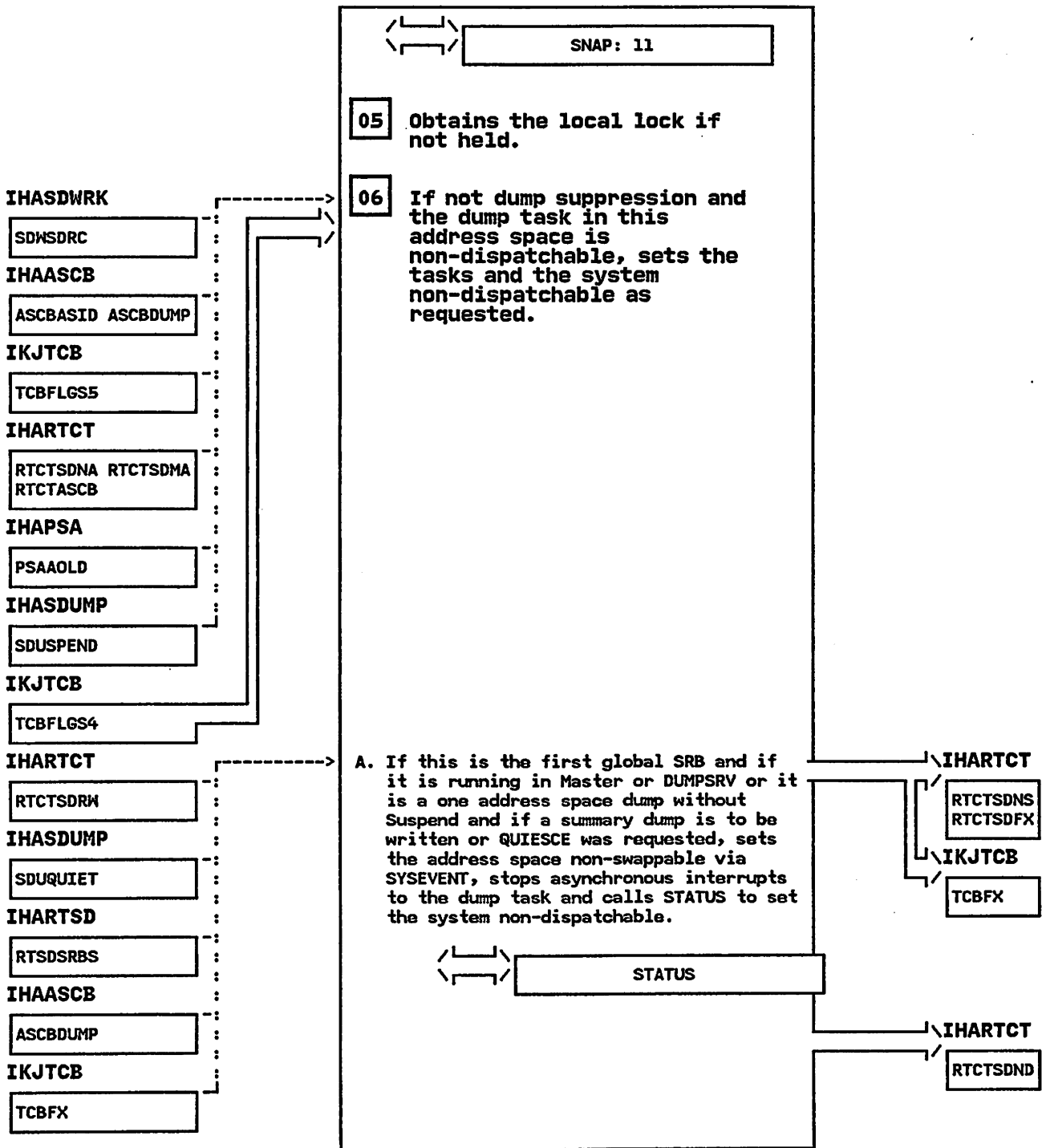
IEAVTSRB - SVC Dump SRB Routine

STEP 02B



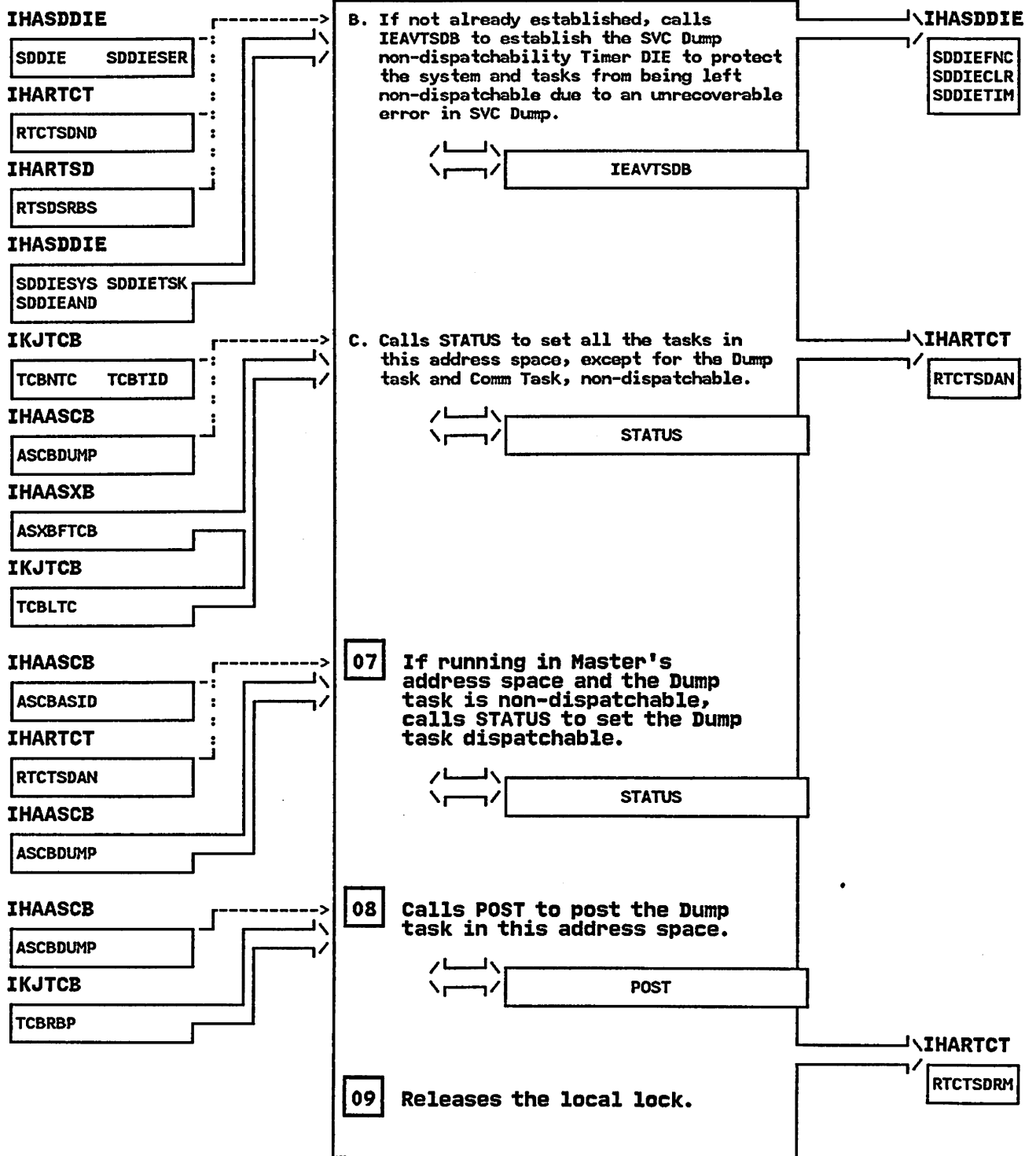
IEAVTSRB - SVC Dump SRB Routine

STEP 05



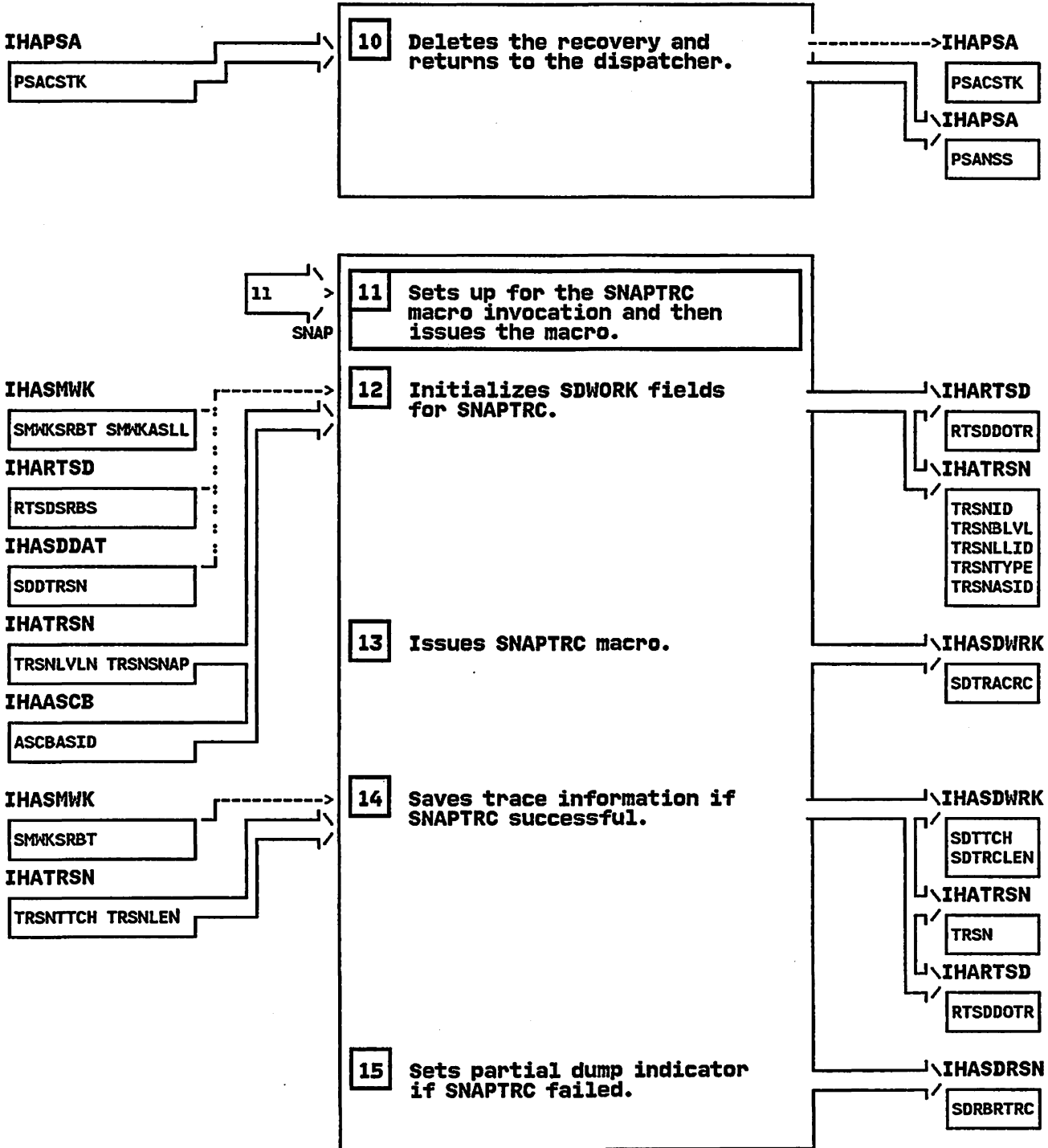
IEAVTSRB - SVC Dump SRB Routine

STEP 06B



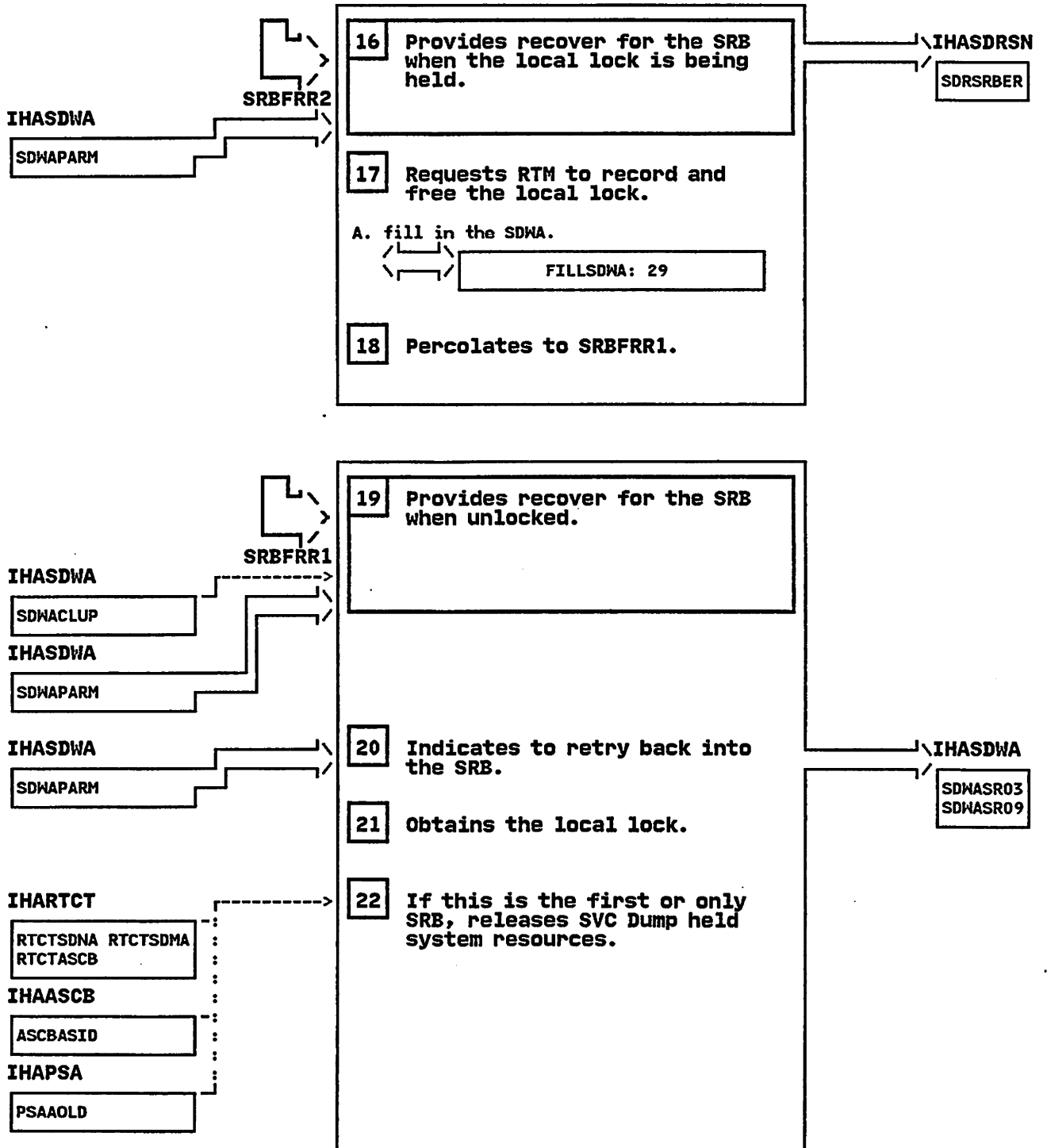
IEAVTSRB - SVC Dump SRB Routine

STEP 10



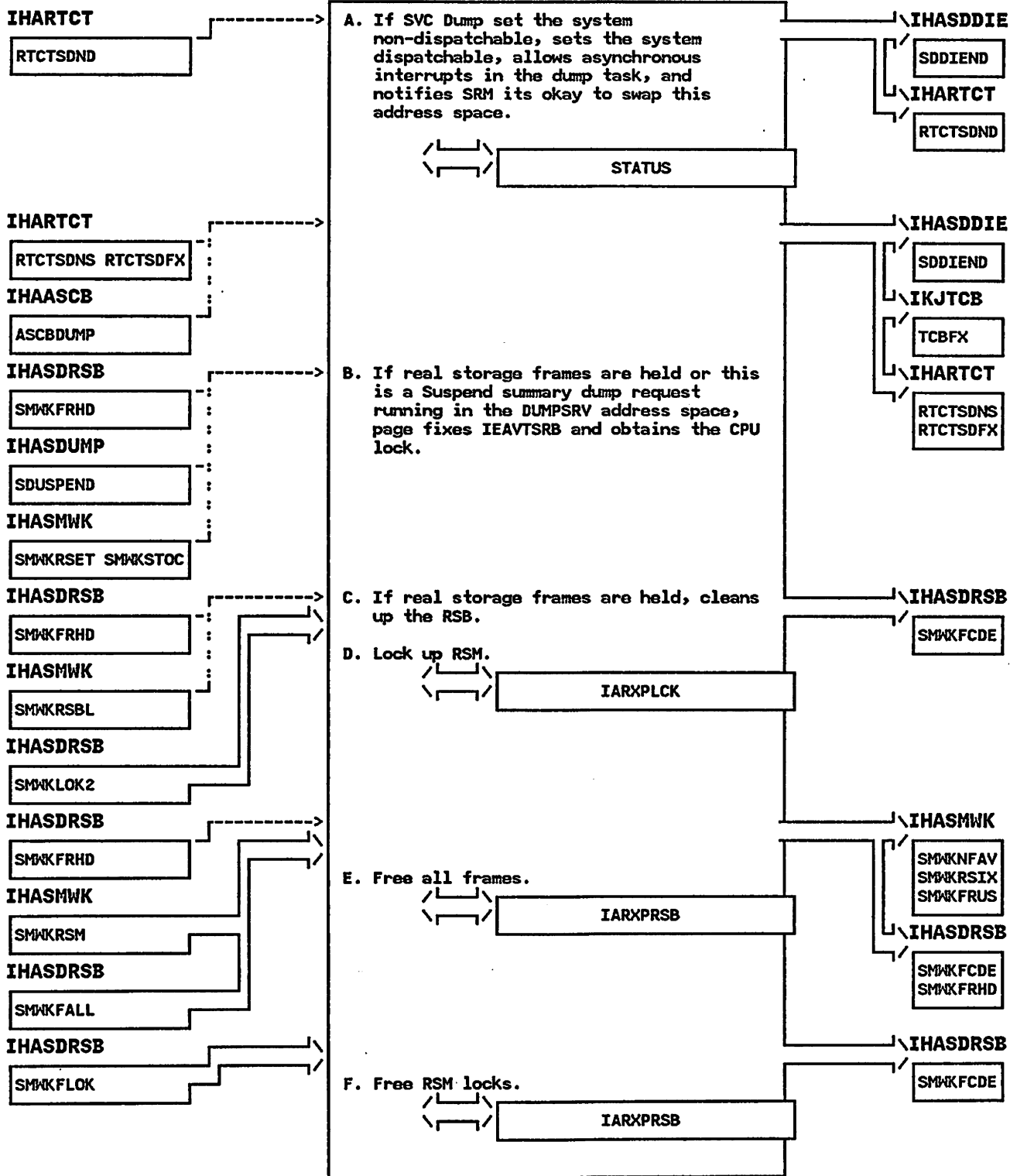
IEAVTSRB - SVC Dump SRB Routine

STEP 16



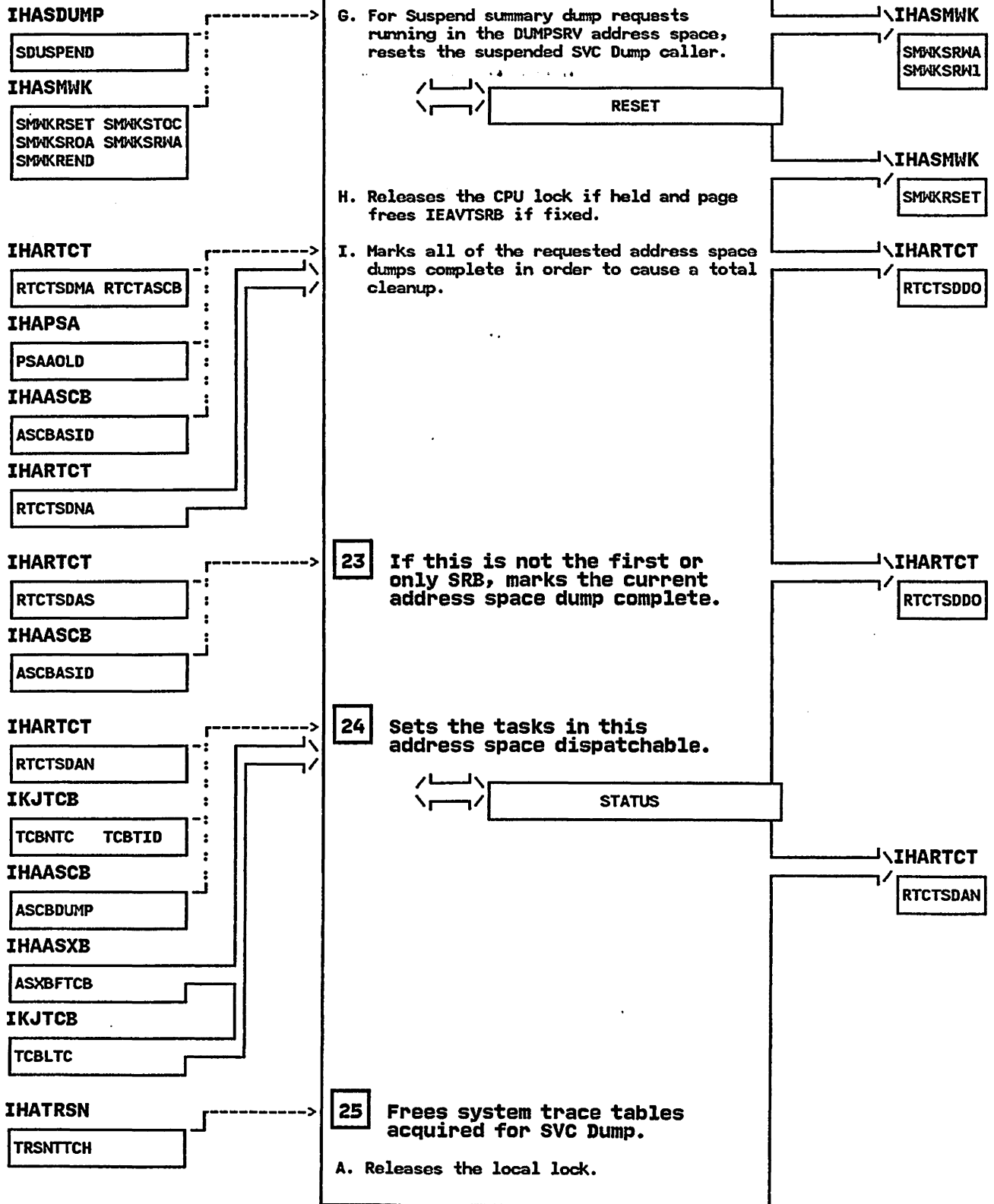
IEAVTSRB - SVC Dump SRB Routine

STEP 22A



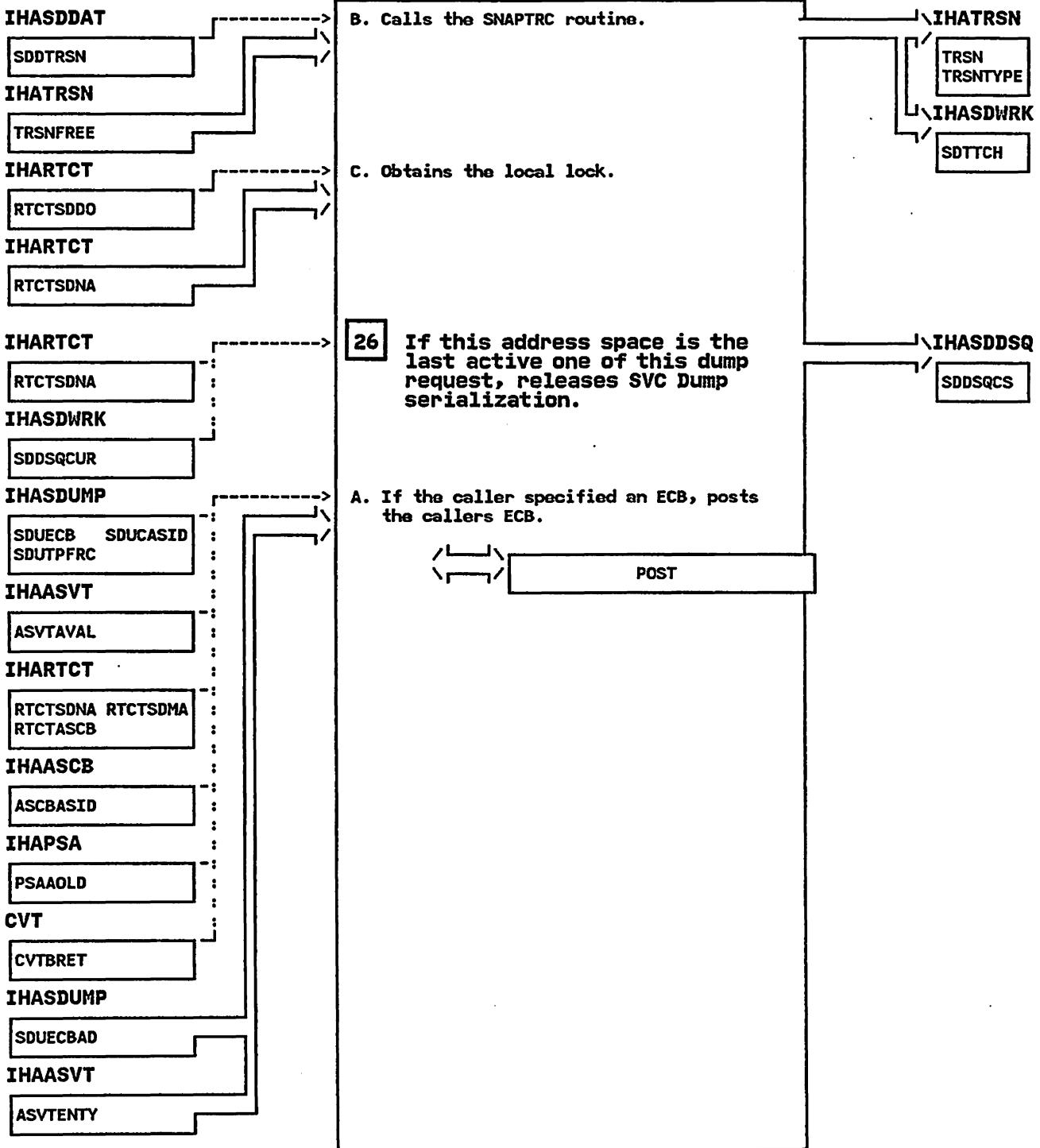
IEAVTSRB - SVC Dump SRB Routine

STEP 22G



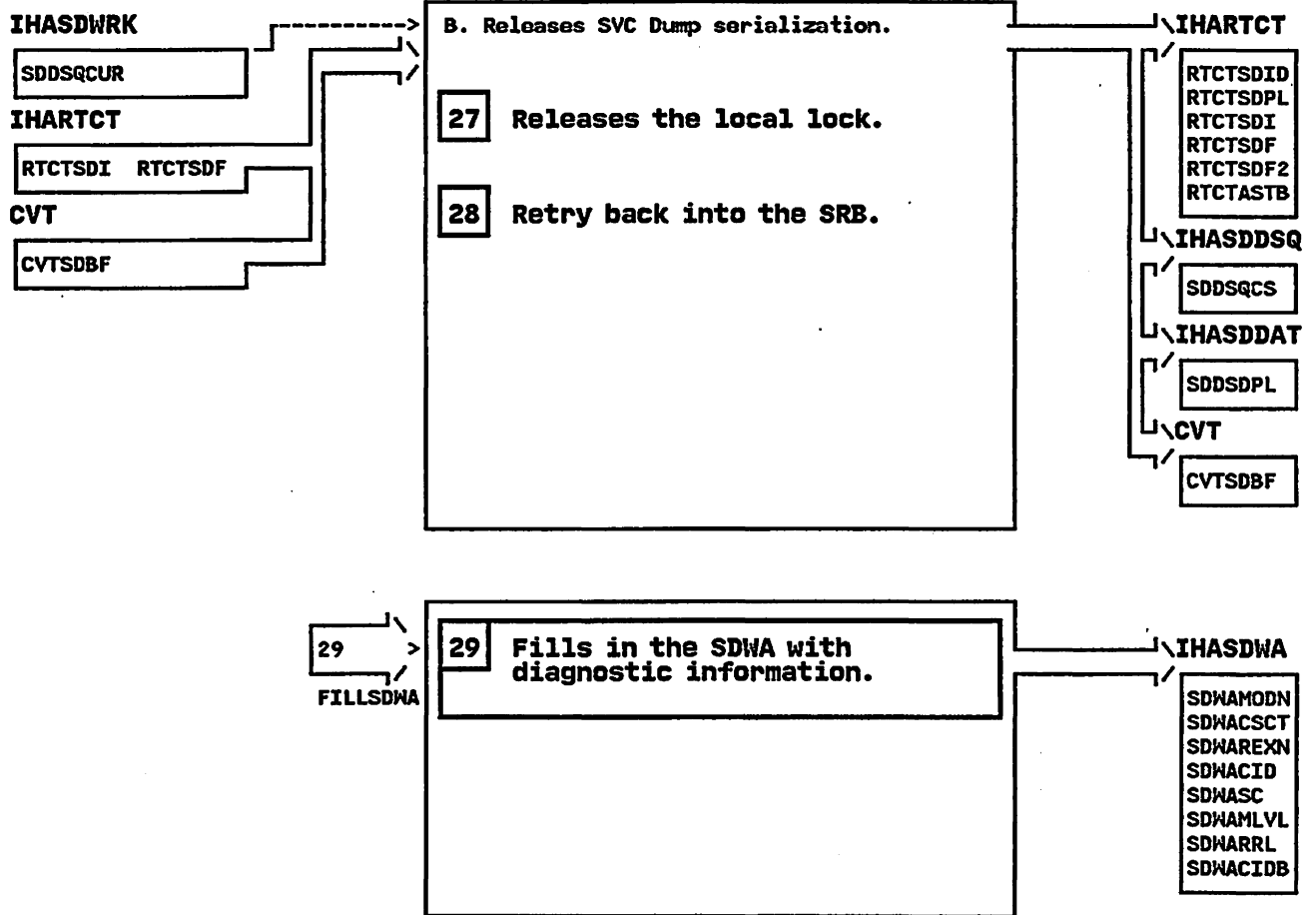
IEAVTSRB - SVC Dump SRB Routine

STEP 25B

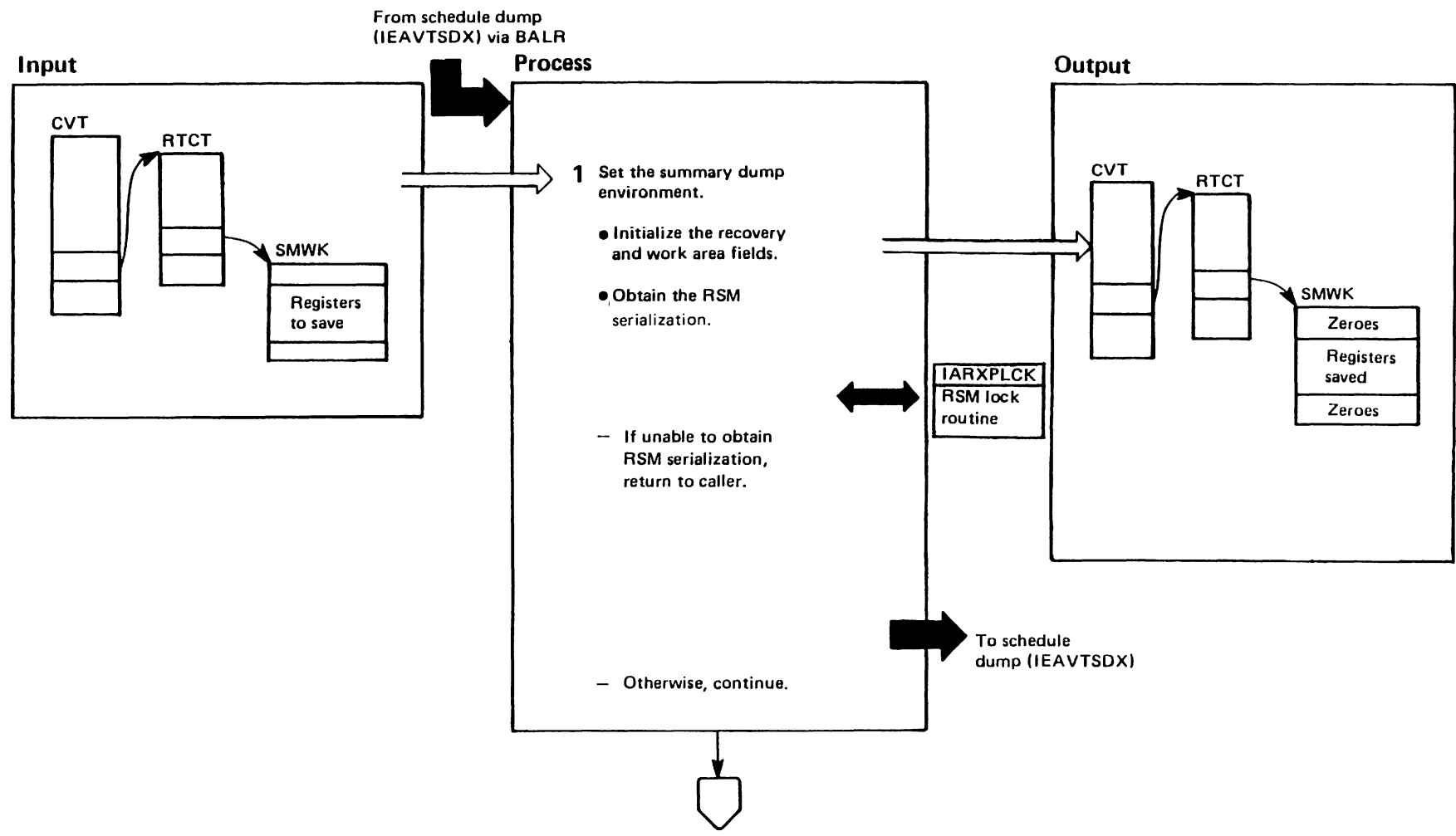


IEAVTSRB - SVC Dump SRB Routine

STEP 26B



IEAVTSSD - Summary Dump Processing (Part 1 of 8)

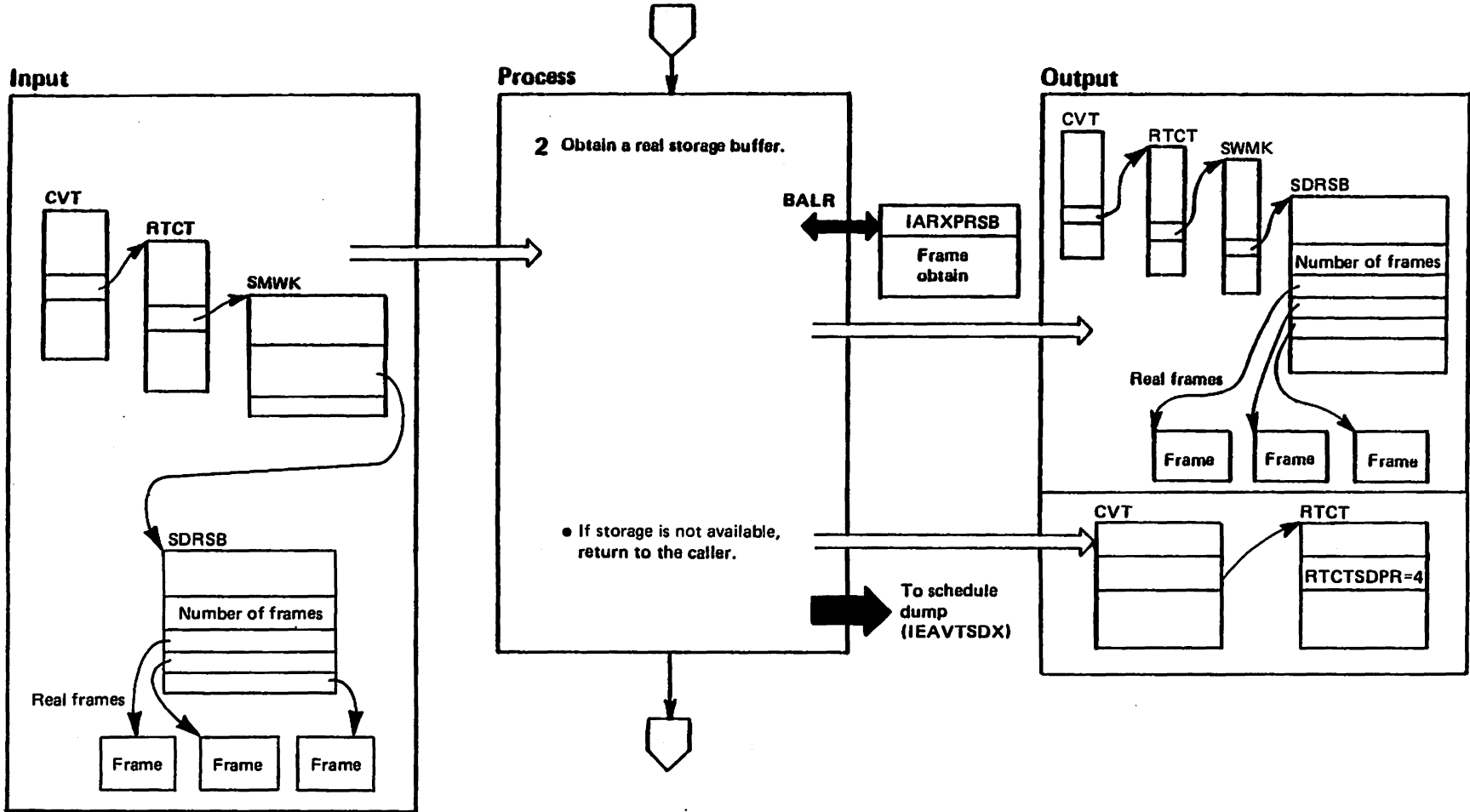


IEAVTSSD - Summary Dump Processing (Part 2 of 8)

Extended Description	Module	Label
<p>Summary dump processing provides a synchronous dump of volatile system data areas. Other system activity or the invoking recovery routine would change these areas before asynchronous SVC dump processing dumped them and they would not reflect the state of the system or the failing component at the time of the error. The volatile data areas summarize the state of the system.</p> <p>All SVC dump invocations where both the BRANCH=YES and SDATA=SUMDUMP have been specified use summary dump processing.</p> <p>IEAVTSSD supports two types of summary dumps, the disabled summary dump and the suspend summary dump. Disabled summary dump (SDATA=SUMDUMP, BRANCH=YES, SUSPEND=NO) uses a real storage buffer to save the volatile system data. Only paged-in data can be saved in this buffer. Disabled summary dump is meant to be used by components whose error related data is volatile because of other system activity. Since these callers have some type of global serialization, they require that the real storage buffer be used.</p> <p>Suspend summary dump (SDATA=SUMDUMP, BRANCH=YES, SUSPEND=YES) uses a virtual storage buffer in the DUMPSRV address space to save the volatile system data. Both paged-in and paged-out data can be saved in this buffer. Suspend summary dump is meant to be used by components whose error data is volatile because it will be changed by the calling recovery routine. Since these callers have some type of local serialization, they require that a virtual storage buffer be used. (The IEAVTSSV diagram documents in detail the areas dumped for a suspend summary dump.)</p> <p>Summary dump processing can be entered in cross memory mode. IEAVTSSD uses the addressing mode of the caller when referencing global system areas (such as the PSA and LCCA). IEAVTSSD simulates the addressing mode at the time the status was saved when dumping the data related to the saved system status (such as the SDWA and IHSA).</p>	IEAVTSSD	

Extended Description	Module	Label
<p>1 IEAVTSSD establishes an FRR to provide recovery for the disabled and suspend summary dump processing located in this module. See recovery processing at the end of this module's extended description for more details. IEAVTSSD saves the caller's registers and initializes the summary dump work area (SMWK).</p> <p>If IEAVTSDX did not obtain serialization for the real storage buffer (RSB) before, IEAVTSSD obtains the CPU lock to provide disablement and then calls IARXPLCK to obtain RSM serialization for the RSB. If IARXPLCK cannot obtain the RSM serialization, a partial dump reason flag is set in RTSDSRSN (mapped by SDRSN) and control returns to IEAVTSDX.</p>	IEAVTSSD	SUMFRR
	IEAVTSSD	SUMINIT
		SUMLOCK

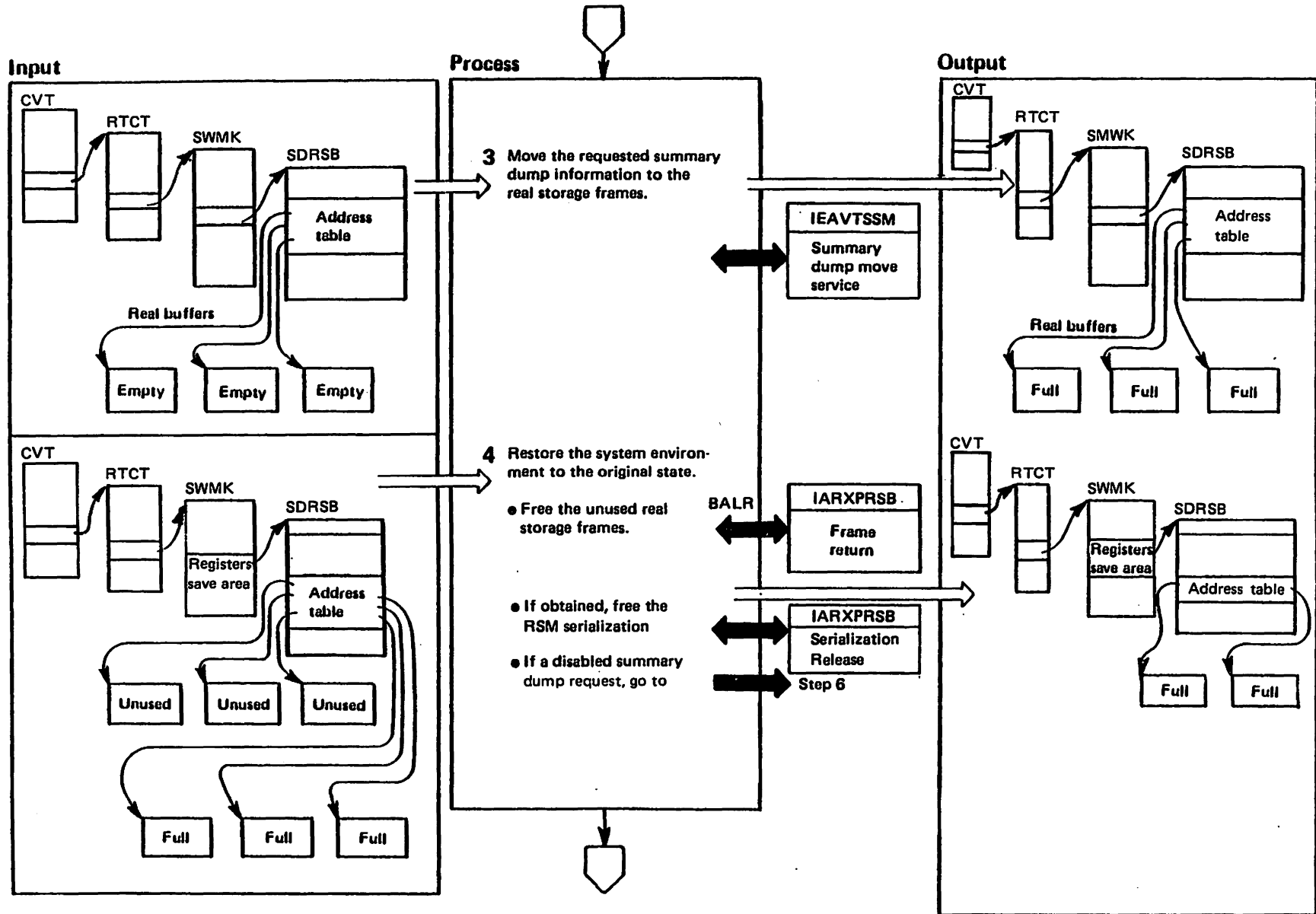
IEAVTSSD - Summary Dump Processing (Part 3 of 8)



IEAVTSSD - Summary Dump Processing (Part 4 of 8)

Extended Description	Module	Label
2 IEAVTSSD calls RSM module IARXPRSB to obtain the real storage buffer. Upon return from IARXPRSB, the SDRSB contains the number of real storage frames obtained and a table containing the real storage address of each frame. If IARXPRSB could not obtain any real storage frames, IEAVTSSD calls IARXPRSB to release the RSM serialization, releases the CPU lock, deletes the FRR and sets a partial dump reason flag in SDRSN (RTSDSRSN).	IEAVTSSD IARXPRSB	TOPSEG

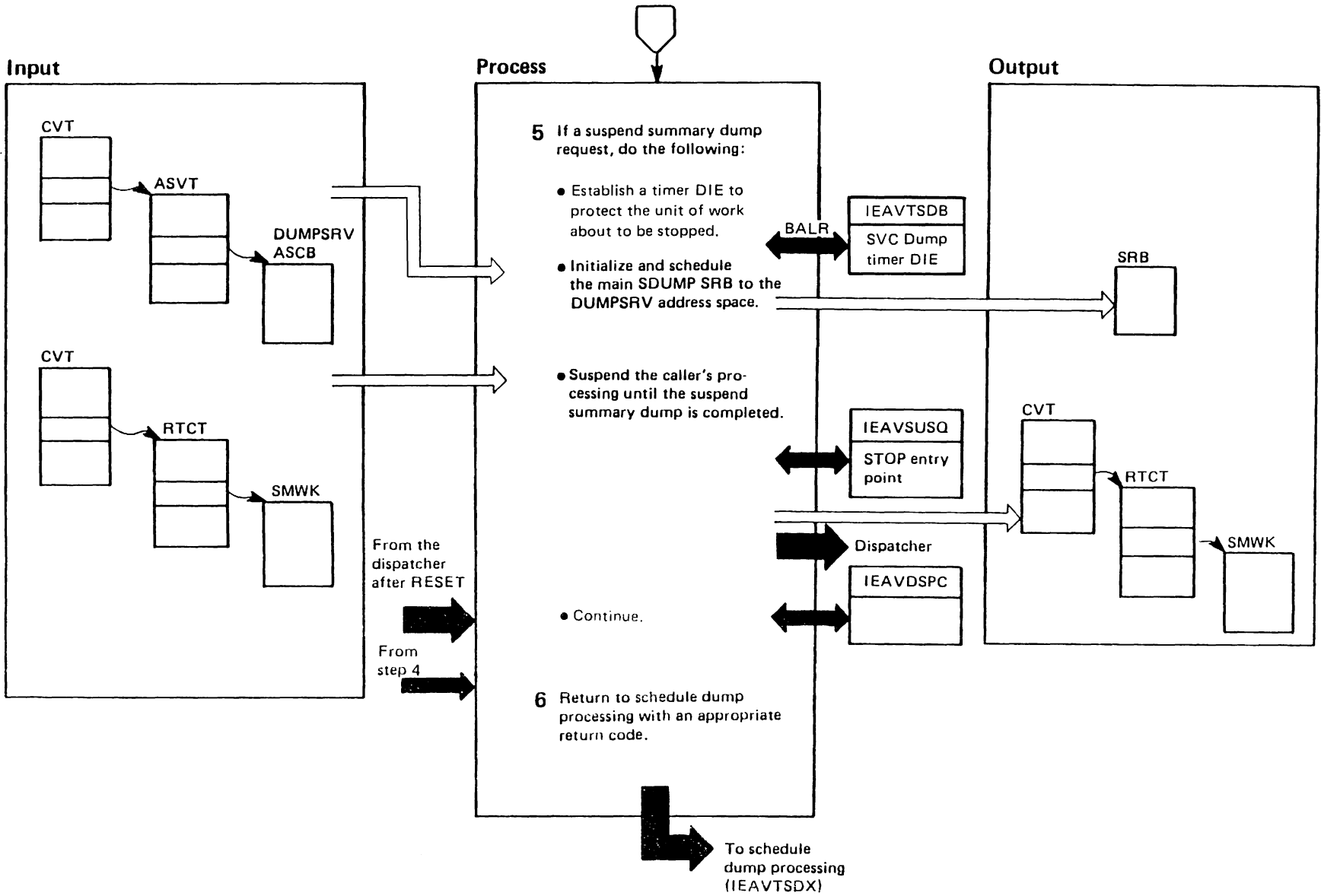
IEAVTSSD - Summary Dump Processing (Part 5 of 8)



IEAVTSSD - Summary Dump Processing (Part 6 of 8)

Extended Description	Module	Label	Extended Description	Module	Label
<p>3 If space should run out in the real storage buffer, summary dump sets a reason flag in RTSDSRSN mapped by IHASDRSN.</p> <p>Both disabled and suspend dump requests cause data to be moved to the allocated real storage buffer. For disabled summary dump requests, IEAVTSSD moves the following areas to the real storage buffer in the order shown.</p> <ul style="list-style-type: none"> • Areas described by the SUMLIST or SUMLISTA storage list addresses (if the SUMLIST or SUMLISTA, keywords were specified on the SDUMP request) • The PSA, LCCA and PCCA for each functioning processor. • The current PCLINK stack elements (pointed to by PSASEL). • The relevant interrupt handler save area (IHSA), its associated XSB, and the PCLINK stack elements pointed to by this XSB. • The super FRR stack and the current system diagnostic work area (FRRS, SDWA). • 4K of storage around non-duplicated areas pointed to by the registers saved in the current SDWA and IHSA. IEAVTSSD uses the addressability in effect when the status was last saved in the SDWA and IHSA. If the primary and secondary ASIDs are different, both will be used to dump the storage related to the saved register values. • The global, processor, and relevant local work save area vector tables. • The global, processor, and relevant local work save areas. • 4K of storage around the instruction counter values of the external old, program check old, I/O old, and SVC old PSW values saved in the LCCA and PSA for all processors. IEAVTSSD uses the caller's primary ASID to reference these values. • 8K of storage around the PSW and general purpose registers from both primary and secondary address spaces when the caller supplies a PSWREGS area. 	IEAVTSSD	TOPSEG	<p>For suspend summary dump requests, IEAVTSSD moves the following areas to the real storage buffer in the order shown:</p> <ul style="list-style-type: none"> • The PSA, PCCA, and LCCA for all active processors. • The relevant interrupt handler save area (IHSA), its associated XSB, and the PCLINK stack elements pointed to by this XSB. <p>IEAVTSSD calls IEAVTSSM to physically move the data to be dumped to real storage buffer frames.</p> <p>4 After moving the requested data areas to the real storage buffer, IEAVTSSD calls RSM to free any unused portions of the real storage buffer. If the RSM serialization was obtained in IEAVTSSD, IEAVTSSD calls module IARXPRSB to release it. IEAVTSSD releases the CPU lock.</p> <p>Disabled summary dump processing continues at step 6. Suspend summary dump processing continues with step 5.</p>	IEAVTSSD	IARXPRSB

IEAVTSSD - Summary Dump Processing (Part 7 of 8)

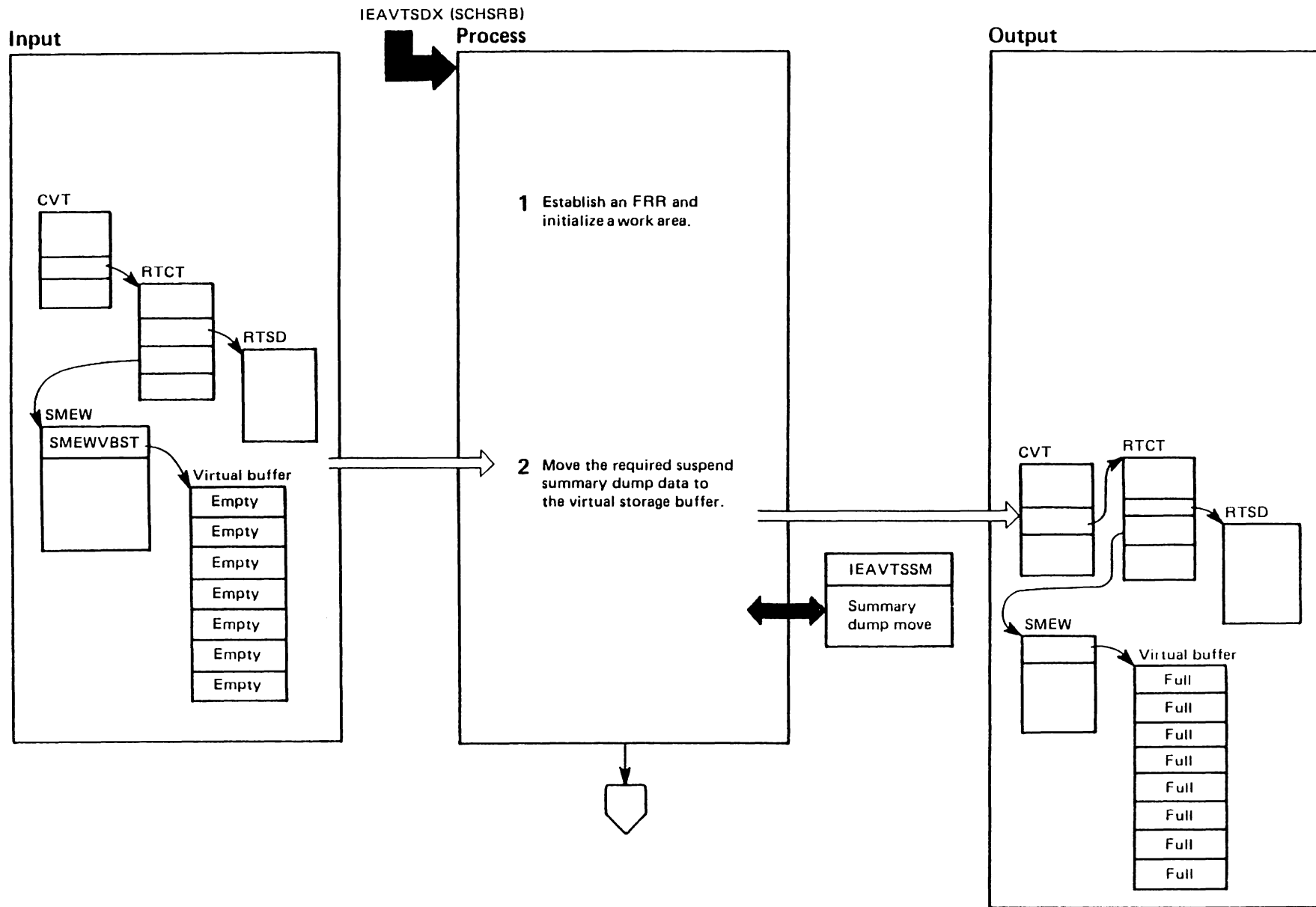


IEAVTSSD - Summary Dump Processing (Part 8 of 8)

Extended Description	Module	Label
<p>5 Only suspend summary dump requests execute this step. Suspend summary dump, running enabled as an SRB routine in the dumping service's address space (DUMPSRV), moves pageable volatile system data to a virtual storage buffer located in the DUMPSRV address space. IEAVTSSD requires the new unit of work in order to address the virtual storage buffer and to handle page-faults when referencing the data being dumped.</p> <p>IEAVTSSD calls IEAVTSDB to establish a timer DIE to protect the unit of work that is stopped during suspend summary dump processing. The timer DIE resets the stopped unit of work if it detects that SVC Dump has halted processing.</p> <p>IEAVTSSD initializes the SRB block passed to this module by IEAVTSDX. IEAVTSSD obtains the LOCAL lock (if a local lock is not already held) in order to make the current home address space non-swappable. IEAVTSSD obtains the SALLOC lock to satisfy the interface requirement for serializing the stop (IEAVSUSQ). Schedule dump processing sets the SRB to run as a global SRB in the DUMPSRV address space. Details of the SRB process are shown in the Schedule Dump SRB Processing (IEAVTSDX) diagram.</p> <p>IEAVTSSD saves the current register values and builds a model PSW in the SMWK. The system restores these values when summary dump processing resumes. IEAVTSSD calls IEAVSUSQ to suspend the current unit of work (SRB or TCB). On return from IEAVSUSQ, IEAVTSSD saves the address of the suspended unit of work in the SMWK, releases the SALLOC lock, and gives control to the dispatcher (at entry point IEAVDSPC).</p>	IEAVTSSD	SSDSPND
		IEAVSUSQ

Extended Description	Module	Label
<p>6 Control passes to this step at the completion of both suspend and disabled summary dump processing. IEAVTSSD deletes the FRR recovery routine from the current FRR stack, using the SETFRR macro.</p> <p>Recovery Processing:</p> <p>RTM gives an FRR routine control if errors occur during summary dump processing. If RTM allows, this FRR will retry. The FRR always records the pertinent diagnostic data saved in the SDWA and the SDWA VRA for this error to SYS1.LOGREC. In cases where retry cannot be performed, this FRR releases all summary dump resources that it holds.</p> <p>When RSM serialization is held, an RSM FRR remains on the FRR stack. If the RSM FRR ever receives control, the RSB is cleaned up and the RSM serialization freed.</p>	IEAVTSSD	TOPSEG SUMFRR

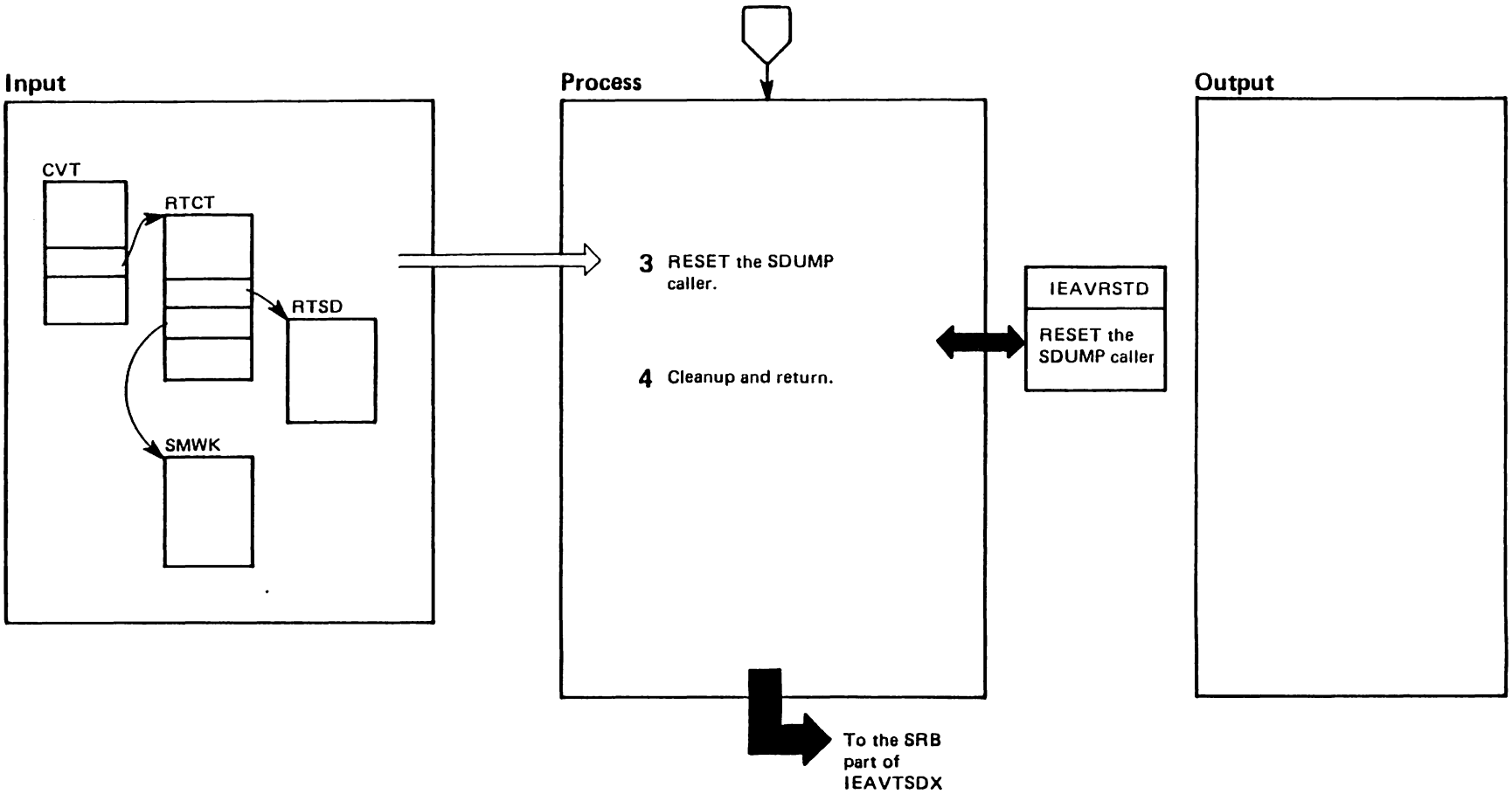
IEAVTSSV - Suspend Summary Dump Processing (Part 1 of 4)



IEAVTSSV - Suspend Summary Dump Processing (Part 2 of 4)

Extended Description	Module	Label	Extended Description	Module	Label
<p>Suspend summary dump processing provides a synchronous summary type dump of volatile system areas. Suspend summary dump moves this volatile system data to a virtual storage buffer located in the dumping services address space (DUMPSRV). IEAVTSSV receives control enabled, in SRB mode, and holding the LOCAL lock of the DUMPSRV address space. IEAVTSSV holds the LOCAL lock so that the DUMPSRV address space will not be swapped out during suspend summary dump processing. Being enabled allows page-fault processing to continue, this means that volatile data that is paged-out when the dump is taken can still be dumped. Accessing this data is important to solving a software problem. This volatile data must be saved now because the recovery routines of the component encountering the error will clean up this data before the regular, scheduled SVC dump could be completed. All SVC dump requests specifying the SDATA=SUMDUMP, BRANCH=YES, and SUSPEND=YES keywords use suspend summary dump. Subpool 229 of the dumping services address space (DUMPSRV) contains the suspend summary dump's virtual storage buffer. A field in the SMEW control block (SMEWVBST) points to the buffer.</p>			<p>2 IEAVTSSV moves the following areas to the virtual storage buffer in the order shown. (If space runs out in the virtual storage buffer, suspend summary dump sets a flag in the SDUMP work area (SDWSDRSN mapped by SDRSN). A partial dump message will result (IEA911E).)</p> <ul style="list-style-type: none"> ● Storage defined by the address ranges specified using the SUMLIST keyword. ● Storage defined by the address ranges specified using the SUMLSTA keyword. ● The SVC dump caller's home address space ASCB. ● The suspended unit of work: <ul style="list-style-type: none"> – For TCB mode callers this is the TCB, RB, and XSB. – For SRB mode callers this is the SSRB and XSB. ● The PCLINK stack associated with the suspended unit of work. ● The error SDWA. ● For task mode callers the RTM2WAs associated with the suspended TCB. ● The SDUMP caller's register save area. ● 4K of storage around the unique register values saved in the SDWA and the caller's register save area. ● 4K of storage around the PSWs saved in the SDWA. 		
		TOPSEG			TOPSEG
		SSVFRR			SSVSMMLST
					SSVSMMLTA
					TOPSEG
					SSVTCB
					SSVSRB
					SSVSDWA
					SSVTCB
					TOPSEG
					SSVSAREA
<p>1 The schedule dump SRB routine (IEAVTSDX) gives suspend summary dump control enabled, in SRB mode, and with the LOCAL lock held. IEAVTSSV saves the caller's registers in the provided register save area and establishes an FRR to cover the suspend summary dump process. This FRR is located in module IEAVTSSV (SSVFRR label).</p>					

IEAVTSSV - Suspend Summary Dump Processing (Part 3 of 4)

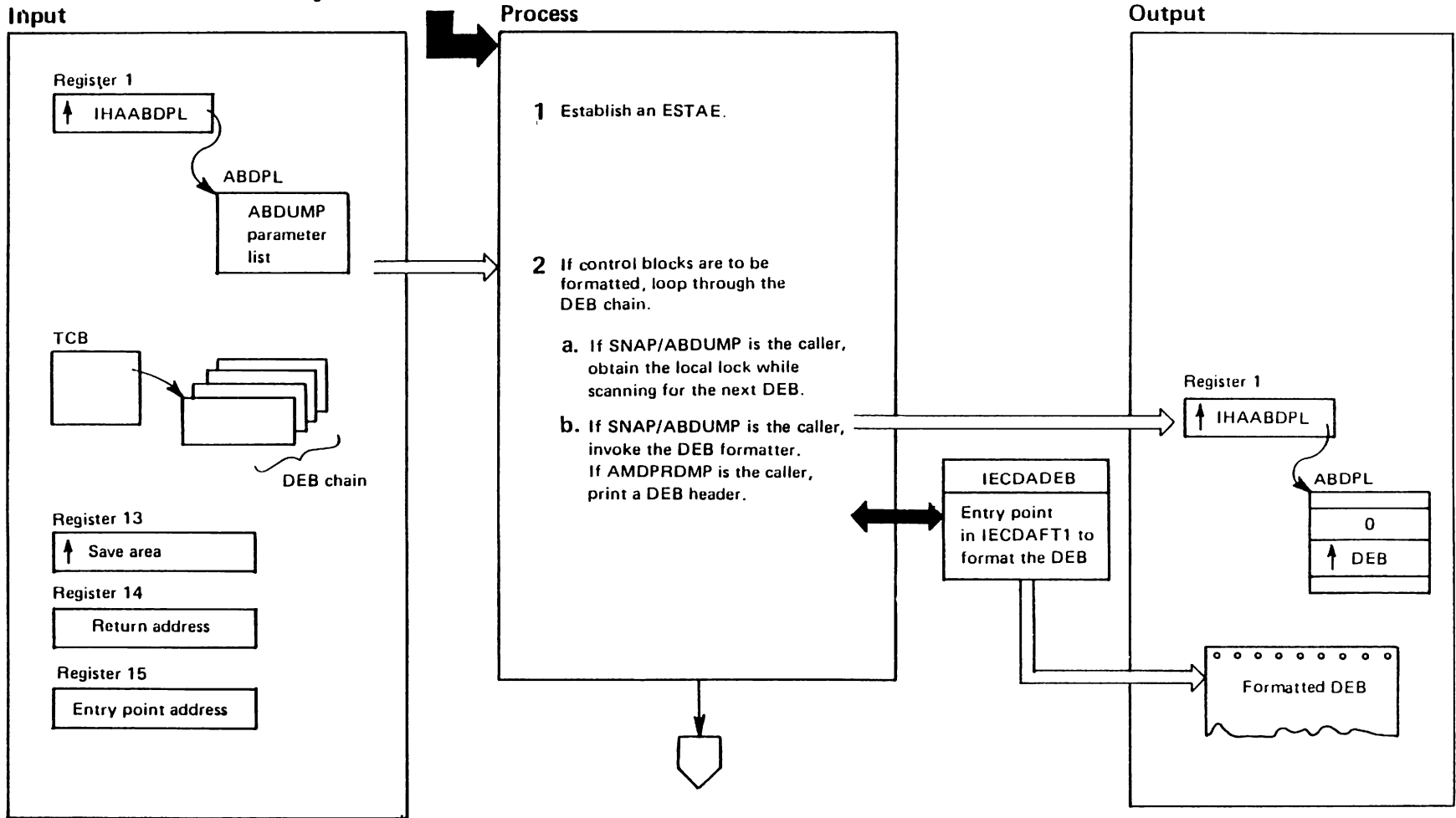


IEAVTSSV - Suspend Summary Dump Processing (Part 4 of 4)

Extended Description	Module	Label
3 IEAVTSSV obtains the address of the suspended unit of work and of the caller's ASCB from the SMWK. IEAVTSSV passes these values as input to IEAVRSTD. IEAVRSTD RESETs the stopped process in module IEAVTSSD. Suspend summary dump processing is now complete so the caller's recovery routines can now modify control blocks and data areas without damaging the contents of the dump.	IEAVTSSV IEAVRSTD	TOPSEG
4 IEAVTSSV issues a SETFRR to delete the IEAVTSSV FRR entry from the current FRR stack. IEAVTSSV restores all the saved registers and returns control to schedule dump SRB processing (IEAVTSDX).	IEAVTSSV	TOPSEG

IECDAFMT - Data Management Formatter Mainline (Part 1 of 4)

Entered as a TCB exit from AMDPRDMP or from SNAP/ABDUMP
to format the data management control blocks



IECDFMT - Data Management Formatter Mainline (Part 2 of 4)

Extended Description

Module Label

This routine operates as a TCB exit for the service aid AMDPRDMP or is directly invoked by SNAP/ABDUMP. This routine locates the data management control blocks to be formatted and invokes the appropriate formatter.

1 IECDAFMT establishes an ESTAE that will retry at a designated retry address for expected errors and will print an ABEND message for unexpected errors. For invocations by SNAP/ABDUMP an error is expected whenever a storage reference is made. If the error is not expected, IECDAFMT places an abend message in the buffer and attempts to retry in the mainline (IECDFMT) at the point where the message is printed. IECDAFMT returns to the caller.

IECDFMT ESTAERIN

2 IECDAFMT determines if control blocks are to be formatted. For a call from SNAP/ABDUMP, IECDAFMT uses the ABDUMP parameter list to make this determination. For print dump (AMDPRDMP), IECDAFMT uses the TCB. The input parameter list must contain the CVT address, the TCB address, and the system whose control blocks are being formatted must be MVS.

IECDFMT

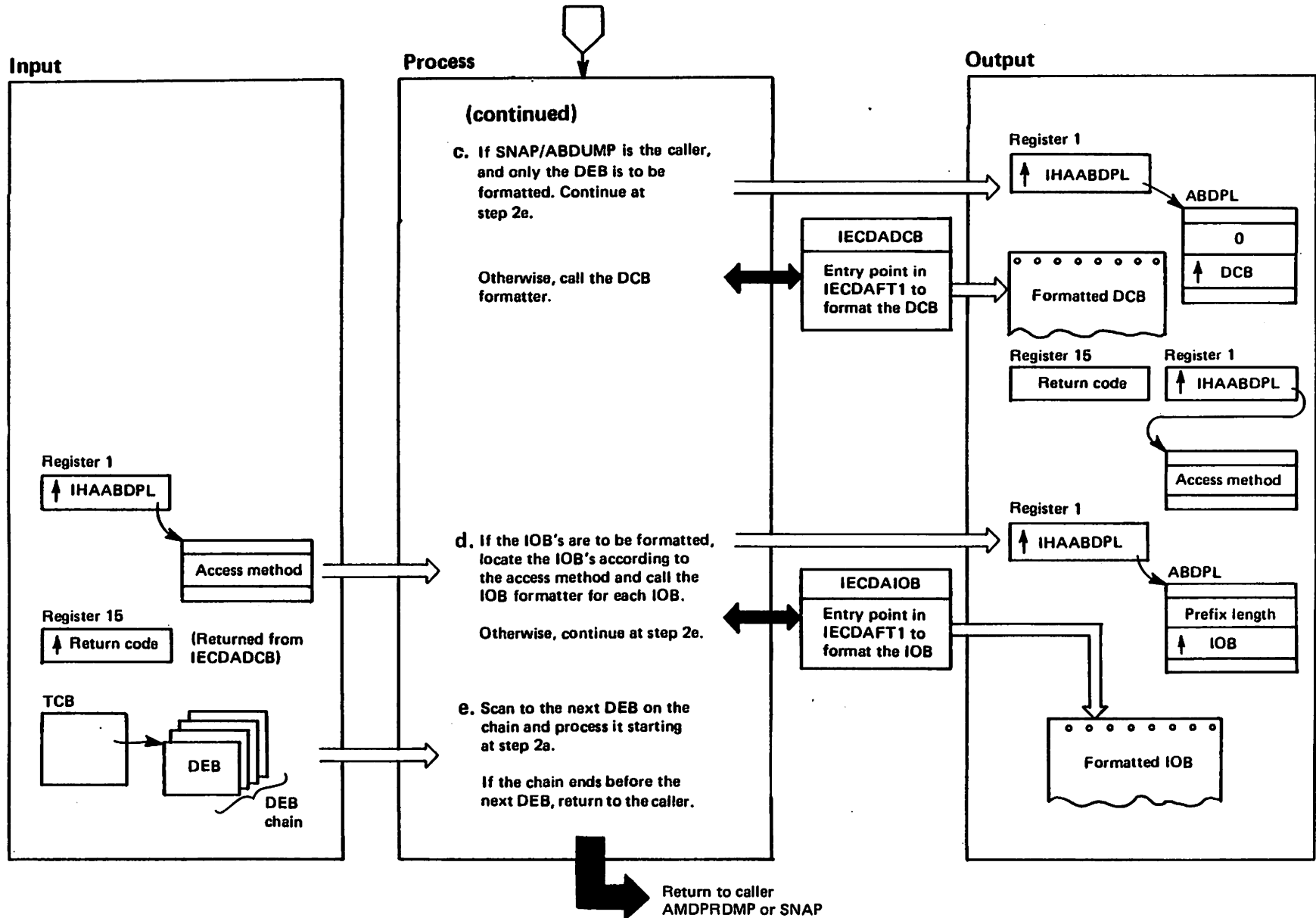
a. IECDAFMT obtains the LOCAL lock if SNAP/ABDUMP is the caller to ensure that the DEB chain does not change while it is formatting the DEBs.

IECDFMT

b. If the caller is AMDPRDMP the DEB has already been formatted. IECDAFMT only needs a DEB header.

IECDFMT FORMATIT

IECDFMT - Data Management Formatter Mainline (Part 3 of 4)



IECDFMT - Data Management Formatter Mainline (Part 4 of 4)

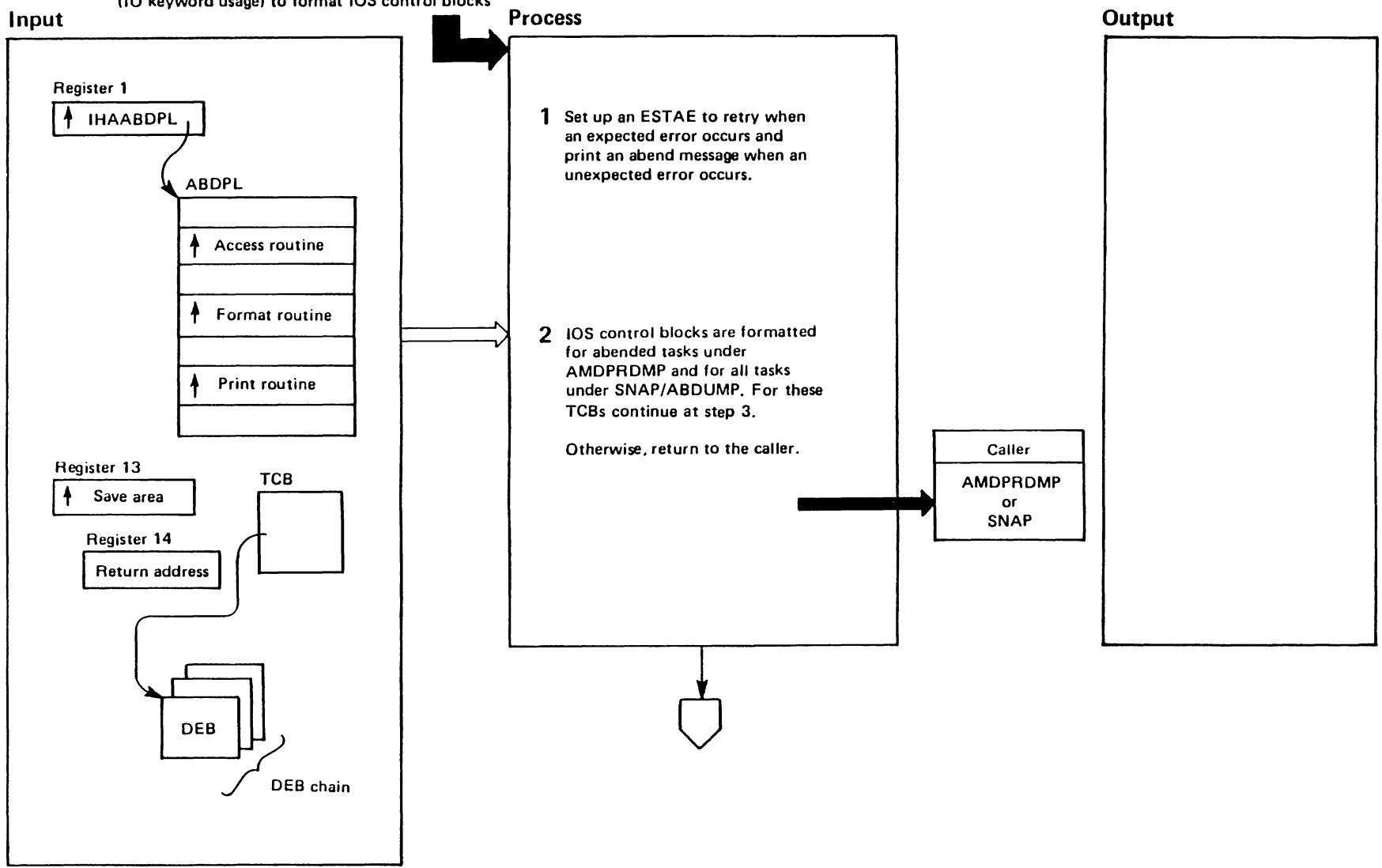
Extended Description	Module	Label
----------------------	--------	-------

Continued

- | | | |
|---|---------|----------|
| <p>c. When the user specified SNAP/ABDUMP with the DM verb, then DCBs and IOBs as well as DEBs are to be formatted. When AMDPRDMP is the caller, the TCB must have been abended in order to have access to the DCBs and IOBs. The TCB has abended if the completion code is non-zero and the ABEND in progress flag is one. The IECDADCB entry point in IECDAFMT formats the DCB.</p> | IECDFMT | FORMATIT |
| <p>d. If a zero return code is returned from the DCB formatter (step 2c), then IOBs (ICBs, LCBs) are to be formatted. IECDAFMT invokes the IOB formatter (the IECDIOB entry point in IECDAFMT) for each IOB (ICB, LCB).</p> | IECDFMT | IOBRTN |
| <p>e. IECDAFMT scans the DEB chain starting from the TCB. The chain could have changed since the last scan. Once a new DEB is found, IECDAFMT releases the LOCAL lock and continues to format that DEB at step 2a.</p> | IECDFMT | |

IECIOFMT - IOS Formatter Mainline (Part 1 of 6)

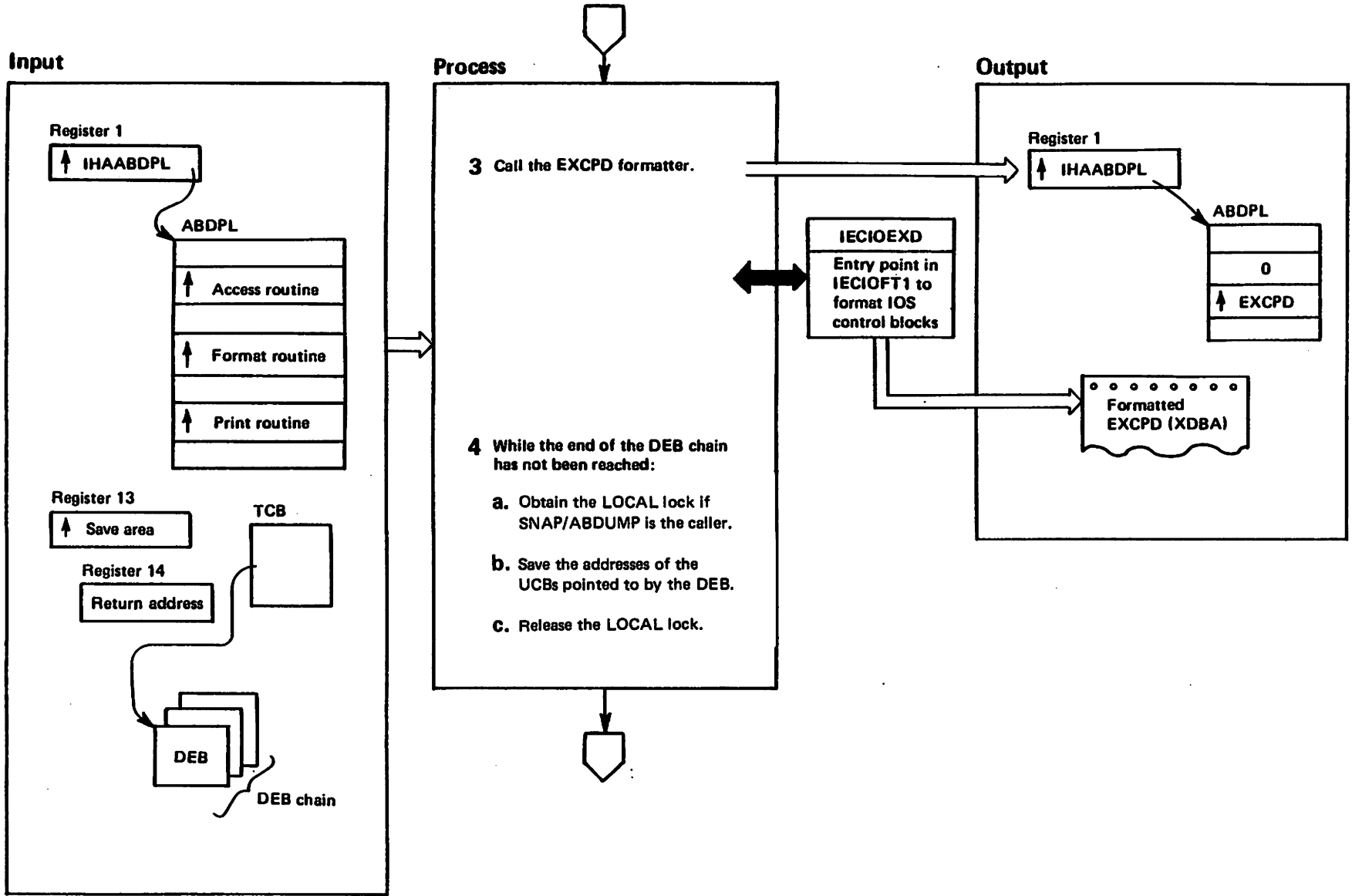
Entered from AMDPRDMP as a TCB exit or from SNAP/ABDUMP (IO keyword usage) to format IOS control blocks



IECIOFMT - IOS Formatter Mainline (Part 2 of 6)

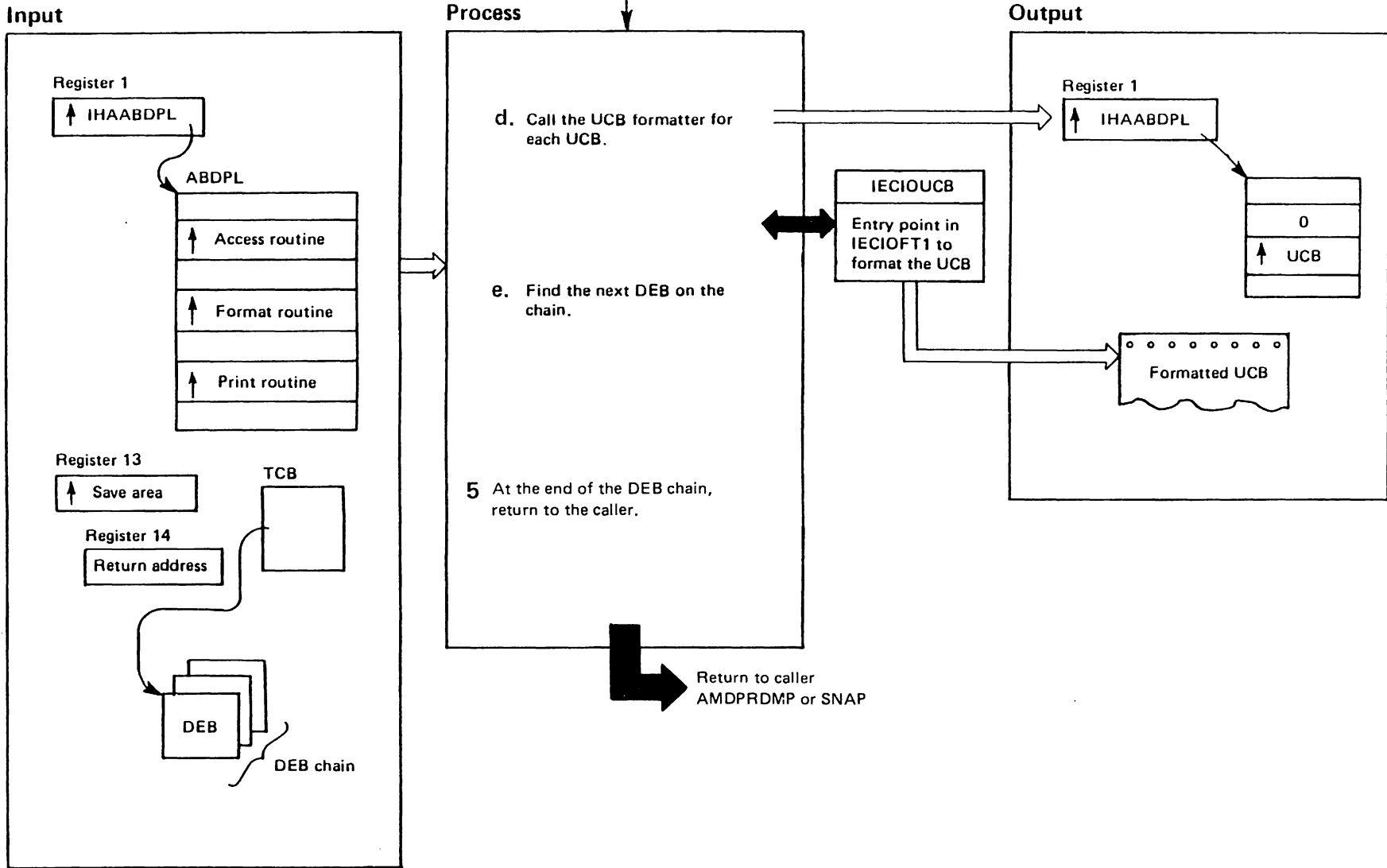
Extended Description	Module	Label
AMDPRDMP and SNAP/ABDUMP (when the user specifies the IO verb) call this module as a TCB exit. It locates the IOS control blocks (XDBA, UCB) that are to be formatted with the appropriate formatter.		
1 IECIOFMT sets up an ESTAE that puts out a message indicating the abend code for an unexpected error. This ESTAE handles expected 0C4 abends, if the invoker is SNAP, and retries at the address supplied by the routine expecting the abend. If the abend code is not 0C4 (even for expected errors), this ESTAE prints the abend message and returns control to the caller.	IECIOFMT	ESTAERTN
2 When invoked by AMDPRDMP, IECIOFMT only formats the control blocks if the completion code in the TCB is non-zero or if the TCB abend in progress flag is one. When SNAP/ABDUMP is the caller, IECIOFMT always formats the IOS control blocks when they are available.	IECIOFMT	IECIOFMT

IECIOFMT - IOS Formatter Mainline (Part 3 of 6)



IECIOFMT - IOS Formatter Mainline (Part 4 of 6)

Extended Description	Module	Label
3 Using the pointer from the TCB, IECIOFMT places the address of the EXCPD (XDBA) in the parameter list passed by the caller and invokes the EXCPD formatter.	IECIOFMT	EXCPDRTN
4 IECIOFMT scans the DEB chain looking for the next DEB. IECIOFMT starts the scan at the pointer from the TCB. The chain could have been altered since the last scan.		UCBRTN
a. In order to insure the DEB chain does not change while it is being scanned, IECIOFMT obtains the LOCAL lock.		UCBRTN
b. Using the addresses from the DEB, IECIOFMT saves the pointers to the UCB's in a table to be formatted, even if the DEB is taken off the DEB chain while the UCB's are being formatted.		UCBRTN
c. IECIOFMT releases the LOCAL lock in order to allow I/O processing to continue.		RELLOCK



IECIOFMT - IOS Formatter Mainline (Part 6 of 6)

Extended Description	Module	Label
d. IECIOFMT calls the UCB formatter (IECIOUCB) for each UCB address saved in the table.	IECIOFMT	UCBRTN
e. IECIOFMT scans the DEB chain looking for the next DEB starting from the TCB. If the DEB chain ends prior to the next DEB, the scan ends and IECIOFMT returns control to the caller. If the DEB chain does not end, IECIOFMT selects the next DEB and continues formatting UCBs.		
5 At the end of the DEB chain, IECIOFMT returns to the caller.		

IEECB867 - MODULE DESCRIPTION

DESCRIPTIVE NAME: SVC Dump Reason Code Text Processor

FUNCTION:

To use the reason code passed by the caller to locate the corresponding text and copy that text into the callers buffer. The set of reason codes handled by this module are the reason codes returned by SVC Dump when a requested dump is not taken (SVC Dump return code of 8).

ENTRY POINT: IEECB867

PURPOSE: see FUNCTION

LINKAGE: BALR

CALLERS: None

INPUT:

Two arguments -
1-reason code
2-address of the callers buffer

OUTPUT: None

EXIT NORMAL: Returns to caller.

EXTERNAL REFERENCES:

ROUTINES: IEECB868

DATA AREAS:

RCTEXT - table in IEECB868 containing the reason code text.

SERIALIZATION: Whatever is held by the caller.

IEECB867 - MODULE OPERATION

1. If the reason code is not a valid one, IEECB867 returns to the caller with a return code of 4.
2. If the reason code is a valid one, IEECB867 copies the text corresponding to the inputted reason code into the callers buffer and then returns to the caller with a return code of 0.

IEECB867 - DIAGNOSTIC AIDS

ENTRY POINT NAME: IEECB867

MESSAGES: None

ABEND CODES: None

WAIT STATE CODES: None

RETURN CODES:

EXIT NORMAL:

0-no error
4-invalid reason code passed as first argument

REGISTER CONTENTS ON ENTRY:

REG 1 - pointer to parameter list
REG 13 - address of callers 72 byte save area
REG 14 - return address
REG 15 - entry point address

REGISTER CONTENTS ON EXIT:

EXIT NORMAL:

R0-R14 are restored
R15 contains the return code

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

INDEX

A

ABDUMP
 processing
 IEAVTABD DMP-82
abend codes
 issued by SVC dump DMP-8
ABEND dump processing
 control flow DMP-23
 introduction DMP-3
abnormal end-of-task termination
 control flow DMP-20
ASCB (address space control block)
 used to debug SVC dump errors DMP-14

B

BRANCH parameter
 SDUMP macro
 usage DMP-7

C

CHNGDUMP operator command DMP-4
console loop trace data
 obtained by IEAVTSDH DMP-116
control blocks
 formatting specific DMP-5
 used to debug SVC dump errors DMP-14

D

data management
 control block formatter DMP-5
 module flow DMP-47
 formatter mainline
 IECDFMT DMP-164
data mapping
 ERRWORK DMP-9
 SDEXPARM DMP-9
debug SVC dump errors
 control blocks used DMP-14
determining the SVC entry point DMP-8
diagnostic techniques
 control blocks used to debug SVC dump
 errors DMP-14
 important SVC dump entry
 points DMP-7
 resource cleanup for SVC dump DMP-16
 SVC dump
 debugging aids DMP-7
 error conditions DMP-8
 SYS1.LOGREC entries produced for SVC
 dump errors DMP-9

DMP

 See dumping services
DMP (Dumping Services)
 method of operation DMP-49
DSVCB (DUMPSRV control block)
 used to debug SVC dump errors DMP-15
dump (formatted)
 module flow DMP-28
dumping services (DMP)
 ABEND dumps
 introduction DMP-3

 data management control block
 formatter
 module flow DMP-47
 data management formatter mainline
 function DMP-164
 introduction DMP-3
 CHNGDUMP operator command DMP-4
 SVC dump overview DMP-4
 unformatted dump SVC DUMP DMP-4
IOS control block formatter
 module flow DMP-48
IOS formatter mainline
 function DMP-168
master trace dump exit processing
 module flow DMP-45
resident dump task processing
 function DMP-116
schedule dump processing
 function DMP-122
schedule dump SRB processing
 function DMP-130
scheduled unformatted dump
 module flow DMP-37
SNAP dump processing
 function DMP-68
summary dump processing
 function DMP-152
suspend summary dump processing
 function DMP-160
SVC post exit processor
 function DMP-134
SVC summary dump formatter
 module flow DMP-32
synchronous unformatted dump
 module flow DMP-33
SYSMDUMP
 introduction DMP-3
SYSUDUMP
 introduction DMP-3
DUMPSRV
 address space jobstep task
 IEAVTDSV DMP-92
 resource manager
 IEAVTSDR DMP-110

E

 entry points for DAE modules
 See module name
 ERRWORK data mapping DMP-9

F

 fixed data
 placed in SYS1.LOGREC DMP-9
 formatted
 and unformatted
 ABEND dumps DMP-3
 dump
 module flow DMP-28
 specific control blocks DMP-5
 FRR (functional recovery routine)
 for IEAVTSDB
 logic diagram DMP-108
 module description DMP-94
 functional recovery routine

See FRR

I

IEAVADIN
 module flow DMP-28
IEAVAD00
 function
 SVC dump processing DMP-56
 SVC 51 overview DMP-54
 module flow DMP-33
 records in the VRA DMP-9
IEAVAD01
 function DMP-68
 module flow DMP-28
IEAVSY50
 posted by IEAVTSDT DMP-118
 posts IEAVTSDX DMP-132
IEAVTABD
 function DMP-82
IEAVTDSV
 function DMP-92
 recovery processing DMP-93
IEAVTFSD
 module flow DMP-32
IEAVTRTM
 module flow DMP-25
IEAVTSDB
 diagnostic aids DMP-98
 logic diagram DMP-99
 module description DMP-94
 module operation DMP-96
 recovery operation DMP-96
 resource cleanup DMP-16
IEAVTSDC
 resource cleanup DMP-16
IEAVTSDR
 function DMP-110
 resource cleanup DMP-16
IEAVTSDT
 function DMP-116
 module flow DMP-40, DMP-42
 records in the VRA DMP-9
IEAVTSDU
 module flow DMP-45
IEAVTSDX
 function DMP-122, DMP-130
 module flow DMP-37, DMP-38
 records in the VRA DMP-9
IEAVTSD1
 posts IEAVTSDV DMP-92
IEAVTSEP
 function DMP-134
IEAVTSRB
 diagnostic aids DMP-140
 entry point in IEAVTSRB DMP-136
 logic diagram DMP-142
 module description DMP-136
 module operation DMP-137
IEAVTSSD
 function DMP-152
 records in the VRA DMP-9
 recovery processing DMP-37
IEAVTSSV
 function DMP-160
IEAOPT01
 posted by IEAVTSDR DMP-110
IECDFMT
 function DMP-164
 module flow DMP-47
IECIOFMT
 function DMP-168
 module flow DMP-48

IEECB867
 diagnostic aids DMP-176
 logic diagram DMP-177
 module description DMP-174
 module operation DMP-175
IEEMB880
 module flow DMP-45
IEE0403D
 module flow DMP-46
important SVC dump entry points DMP-7
input/output supervisor
 See IOS
introduction
 dumping services DMP-3
IOS (I/O supervisor)
 control block formatter DMP-5
 module flow DMP-48
 formatter mainline
 IECIOFMT DMP-168

K

key to logic diagrams DMP-49, DMP-50

L

logic for DMP DMP-49

M

master trace dump exit processing
 module flow DMP-45
method of operation
 DMP DMP-49
module flow
 RTM DMP-20

N

NDISPSRB
 entry point in IEAVTSDB
 module description DMP-94
 logic diagram DMP-105

O

operator command
 CHNGDUMP DMP-4

P

post exit processor
 SVC dump
 function DMP-134

R

recovery processing
 IEAVTDSV DMP-93
 IEAVTSSD DMP-37
recovery termination control table
 See RTCT
recovery termination manager
 See RTM
resident dump task processing
 IEAVTSDT DMP-116
resource cleanup for SVC dump DMP-16

**"Restricted Materials of IBM"
Licensed Materials - Property of IBM**

RSBSRB
entry point in IEAVTSDB
module description DMP-94
logic diagram DMP-100

RTCT (recovery termination control table)
used to debug SVC dump errors DMP-15

RTCT extension for SDUMP
See RTSD

RTM (recovery termination manager)
ABDUMP processing DMP-82
control flow
for ABEND/SNAP dump processing DMP-23
for abnormal task termination DMP-20
for SVC dump DMP-25
data management formatter mainline function DMP-164
dumping services
data management control block formatter module flow DMP-47
formatted dump module flow DMP-28
IOS control block formatter module flow DMP-48
master trace dump exit processing module flow DMP-45
RTM services DMP-46
scheduled unformatted dump module flow DMP-37
SVC summary dump formatter module flow DMP-32
synchronous unformatted dump, module flow DMP-33
introduction
ABEND/SNAP dump processing DMP-23
abnormal end-of-task DMP-3
data management control block formatter DMP-5
dumping DMP-3
formatted and unformatted ABEND dumps DMP-3
IOS control block formatter DMP-5
SVC dump overview DMP-25

IOS formatter mainline function DMP-168
resident dump task processing function DMP-116
schedule dump processing function DMP-122
schedule dump SRB processing DMP-130
SNAP dump processing DMP-68
summary dump processing function DMP-152
suspend summary dump processing function DMP-160
SVC post exit processor function DMP-134
SVC 51 overview DMP-54

RTM services
module flow DMP-46

RTSD (RTCT extension for SDUMP)
used to debug SVC dump errors DMP-15

S

SCHDISRB (label in IEAVTSDX)
module flow DMP-38

schedule dump
options DMP-7
processing
IEAVTSDX DMP-122
SRB processing
IEAVTSDX DMP-130
scheduled unformatted dump
module flow DMP-37
SDDIE (SVC dump DIE control block)
used to debug SVC dump errors DMP-15
SDEXPARM data mapping DMP-9
SDUMP macro
usage DMP-7
SDWORK (SVC dump work area)
used to debug SVC dump errors DMP-15
SMWK (summary dump work area)
used to debug SVC dump errors DMP-15

SNAP
dump processing
control flow DMP-23
IEAVAD01 DMP-68
used by ABEND dump introduction DMP-3

SRBFRR1
entry point in IEAVTSRB DMP-136

SRBFRR2
entry point in IEAVTSRB DMP-136

SUMDUMP formatter
module flow DMP-32

summary dump
processing
IEAVTSSD DMP-152
summary SVC dump formatter
module flow DMP-32

suspend
summary dump
processing, IEAVTSSV DMP-160

SVC
entry point
determining the DMP-8
post exit processor
function DMP-134
summary dump formatter
module flow DMP-32

13
module flow DMP-20

51
introduction DMP-3
overview DMP-54

SVC dump
/DUMPSRV resource manager
IEAVTSDR DMP-110
abend code issued DMP-8
control blocks used to debug errors DMP-14
debugging aids DMP-7
error conditions DMP-8
important entry points DMP-7
overview
RTM control flow DMP-25
overview in RTM DMP-4
processing
IEAVAD00 DMP-56
resource cleanup for DMP-16
title line indicator DMP-8
synchronous unformatted dump
module flow DMP-33

SYSDUMP
introduction DMP-3

SYSUDUMP
introduction DMP-3
SYS1.LOGREC
variable data placed in DMP-9
SYS1.LOGREC entries produced for SVC
dump errors DMP-9

V
variable data
placed in SYS1.LOGREC DMP-9
variable recording area
See VRA
VRA (variable recording area)
data recorded by DMP routines DMP-9

.T.
title line of SVC dumps
indicator DMP-8

1
133 abend code
issued by SVC dump DMP-8

U
unformatted dump
scheduled
module flow DMP-37
SVC dump DMP-4
synchronous
module flow DMP-33

2
233 abend code
issued by SVC dump DMP-8

LY28-1670-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

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.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

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

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

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 or you may mail directly to the address in the Edition Notice on the back of the title page.)

"Restricted Materials of IBM"
All Rights Reserved
Licensed Materials - Property of IBM
(Except for Customer-Originated Materials)
©Copyright IBM Corp. 1987, 1989
LY28-1670-1

S370-36

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

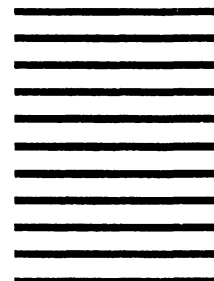
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 950
Poughkeepsie, New York 12602



Fold and tape

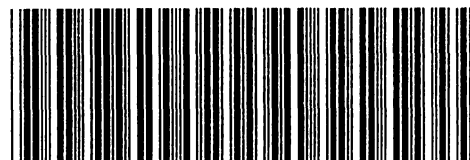
Please Do Not Staple

Fold and tape

Printed in U.S.A.



LY28-1670-01



MVS/Extended Architecture System Logic Library: Dumping Services

"Restricted Materials of IBM "
All Rights Reserved
Licensed Materials - Property of IBM
©Copyright IBM Corp. 1987, 1989
LY28-1670-1

S370-36

Printed in U.S.A.

