SY24-5161-1
File No. S370-36

Systems

OS/VS1 Job Management Logic

VS1 Release 2

IBM

Second Edition (January 1973)

This edition applies to Release 2 of IBM System/370 OS/VS1 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *System/360 and System/370 SRL Newsletter,* GN24-0360, for the editions that are applicable and current.

This edition is a major revision of, and obsoletes, SY24-5161-0 and Technical Newsletter SN24-5459.

Requests for copies of IBM publications should be made to your IBM representative or to the 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 Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York, 13760. Comments become the property of IBM.

Summary of Amendments for Y24-5161-1 VS1 Release 2

This revision reflects the availability of OS/VS1 Release 2. Included is support for I/O Load Balancing, Greenwich Mean Time, and the Remote Entry Subsystem(RES). Miscellaneous changes are also included.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

I/O Load Balancing

I/O load balancing increases system performance by allocating data sets to devices to equalize the amount of I/O contention on each device. I/O load balancing accumulates information about the speed of the devices, counts the number of I/O events to each device, and compares the characteristics of different devices in determining the device to be allocated.

Greenwich Mean Time

For certain applications, it is advantageous to use a time clock that is independent of local time. Using the Store Clock STCK instruction, the time of day can be obtained in Greenwich Mean Time.

Remote Entry Subsystem (RES)

RES is a logical and functional extension of the Job Entry Subsystem (JES). It extends the JES function so that remote users can be attached to OS/VS1. Using RES, you can submit jobs from remote terminals, and output can be routed back to the terminals. RES makes OS/VS1 available to teleprocessing terminals so that more users can have direct access to the central system for their job submission. Data transmission is via Binary Synchronous Communication (BSC) Data-Link Facilities.

Miscellaneous Changes

The sectionalized index is changed to a standard index. A new cross-reference, listing module names and the figure number(s) in which they can be found, is added to the manual. Additional data areas are added to *Section 5: Data Areas.*

Preface

This publication supplements the program listings and makes the information in the listings easier to access. It is for persons involved in program maintenance, providing them with:

- hierarchal overview of the job management functions to show how they operate.
- functional system flow to illustrate the functions performed by the various job management modules.

The user of this publication should be familiar with OS/VS1 concepts and terminology, which can be found in the prerequisite publications listed here. This publication contains six sections:

Section 1: Job Management Concepts introduces the functions and major elements of job management.

Section 2: Method of Operation gives a hierarchal view of job management logic, by function, and provides a cross reference to the functional flow diagrams in Section 3 for more detail.

Section 3: Program Organization uses the functional flow diagrams to trace, step by step, the functions of the job management modules. Only the most representative cases are documented.

Section 4: Cross-Reference Directories shows modules, load modules, entry points, and CSECT relationships. A module name and figure number cross-reference is also provided.

Section 5: Data Areas illustrates the data areas used only by the VS1 scheduler.

Section 6: Diagnostic Aids lists debugging aid information, register usage by module, message/generating module relationship, return codes and their meanings by module, and pointers to the various control blocks.

An appendix, *Dictionary of Abbreviations,* is also included.

This publication assumes a knowledge of the OS/VS1 job managment functions. The section *Method of Operation* presents a hierarchal view of the job managment functions and serves as a directory to the more detailed figures in the section *Program Organization.* The *Program Organization* section can be used alone or with the figures in the section *Method of Operation* to follow the functions performed by the modules.

The data areas for a function are usually referred to only by name in the *Method of Operation* and *Program Organization* sections. The data area fields, flags, and masks are illustrated for easier information retrieval.

Prerequisite Publications

Knowledge of the information in the following publications is required for an understanding of this publication:

OS/VS1 Planning and Use Guide, GC24-5090

OS/VS JCL Services, GC28-0617

OS/VS JCL Reference, GC28-0618

OS/VS Supervisor Services and Macros, GC27-6989

Associated Publications

Publications the user will find helpful and which are referred to within this publication include:

OS/VS1 Supervisor Logic, SY24-5155

OS/VS System Data Areas, SY28-0605

OS/VS Job Control Language Reference, GC28-0618

OS/VS System Management Facilities, GC35-0004

OS/VS Debugging Guide, GC24-5093

OS/VS Open/Close/End-of-Volume Logic, SY27-3785

OS/VS Recovery Management Support Logic, SY27-7239

OS/MFT, OS/MVT, and OS/VS1 CRJE Logic, GY30-2011

Operator's Library: OS/VS1 Operator's Reference, GC38-0110

OS/VS Message Library:

OS/VS1 System Messages, GC38-1001

System Codes, GC38-1003

Routing and Descriptor Codes, GC38-1004

Contents

Section 1: Job Management Concepts

CONTENTS

Job management is the first and last portion of the OS/VS1 (Operating System/Virtual Storage, Option 1) control program to interface with user-submitted jobs. The primary function of job management is to prepare jobs for execution and to direct the disposition of data sets used during execution. To do this, job management must:

- ensure that the required I/O devices for the job are available, and that the appropriate data sets and scratch volumes are mounted.
- oversee the operation and scheduling of jobs (for example, canceling a job or changing the order in which jobs are executed).

Figure 1-1 shows the routines that comprise job management. A brief description of the purpose of each routine is included. Many of the names are the same as those found in other existing functions, such as OS MFT--Operating System Multiprogramming with-a-Fixed-number-of-Tasks. However, the functions performed are new or modified for OS/VS1.

The problem-program partitions defined for the installation are a part of the total virtual storage, but they must begin on segment boundaries (integral multiples of 64K). Therefore, a change in partition size is made in virtual storage and not in real storage as in OS MFT. The size of the virtual storage partition in OS/VS1 can, therefore, be greater than the real storage size of the CPU. However, programs are still executed in real storage; that is, main storage from which the CPU directly obtains instructions and data, and in which the CPU stores data. Pages residing in virtual storage are transferred into and out of real storage as needed under the control of the page supervisor.

Each initiator has a direct-access storage Scheduler Work Area Data Set (SWADS) that contains the necessary scheduler tables for the problem program in that partition. The SWADS is a sequentially organized and reusable data set; that is, the tables for executing one job overlay those of a previous job.

Job Scheduling Services

The job-scheduling services (job management) direct and control the flow of one or more jobs through the computing system. The job-control statements provide the communication link between the user and job management. These services include:

- *Data Set Integrity:* Ensuring that jobs do not access data sets that are not available.
- *Analysis of the Input Stream:* Scanning the input data to identify control statements; interpreting and analyzing the control statements; and preparing the necessary control tables that describe each job to the system.
- *Allocation of I/O Devices*: Ensuring the allocation of all necessary I/O devices; ensuring the allocation of direct-access storage space, as required; and ensuring that the operator has mounted any required tape and direct-access volumes. Allocation issues mount messages for the tape volumes. However, tape volumes are verified at OPEN time by data management.
- *Overall Scheduling:* Selecting jobs for execution, by class and by priority within a class.
- *Communication* between the operator, programmer, and the system.

Figure 1-2 illustrates the VS1 job scheduling services and how these services relate to problem programs. The problem program is a separate task to the system. The system also has its own tasks to perform; that is, each activity the system must perform is a task. Some tasks, such as those shown in Figure 1-2 in the Job Entry Subsystem (JES), are performed in parallel. Other tasks, such as those shown in the user problem-program partition, can be performed sequentially or in a multitasking environment. The initiator, termination, and the problem program, however, all operate under the same task control block.

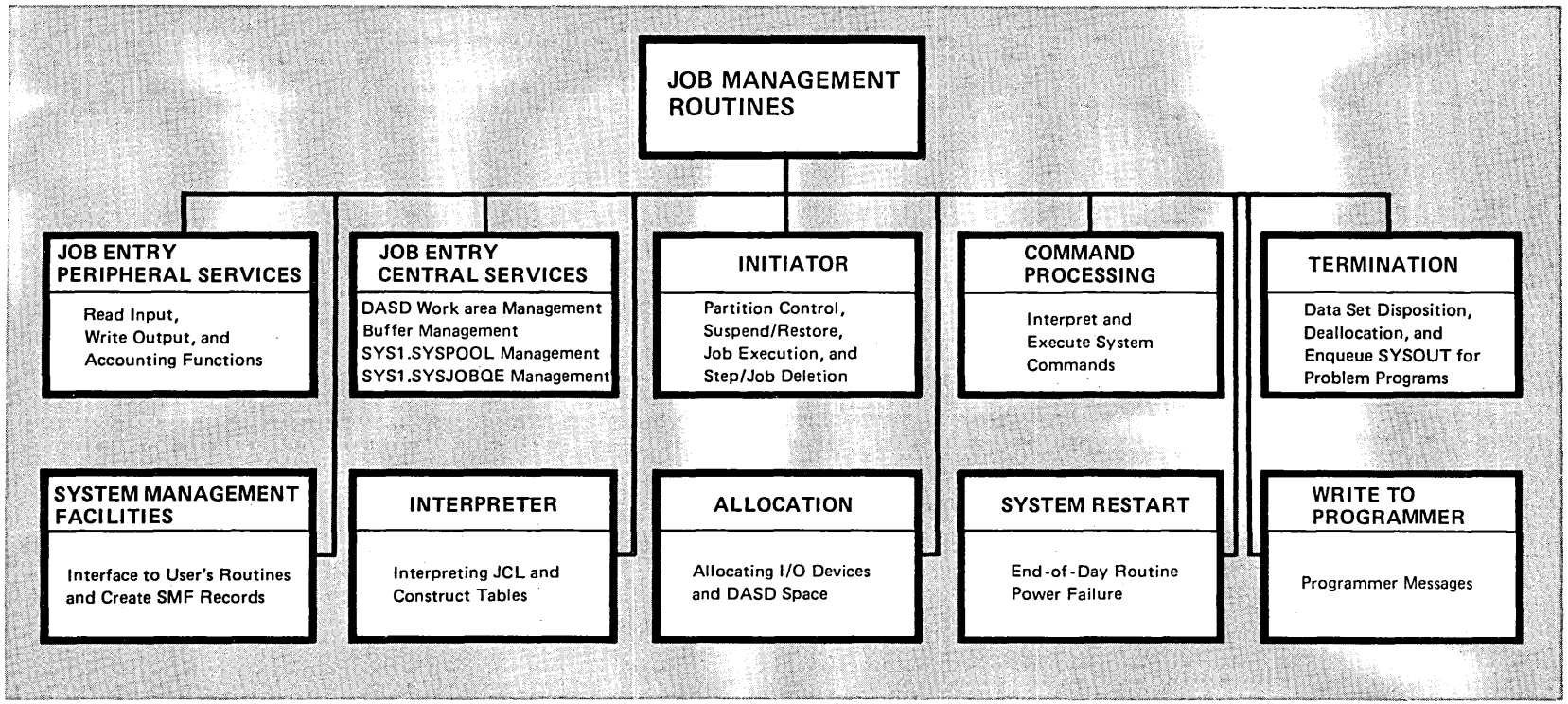


Figure 1-1. Routines That Perform VS1 Job Management Functions

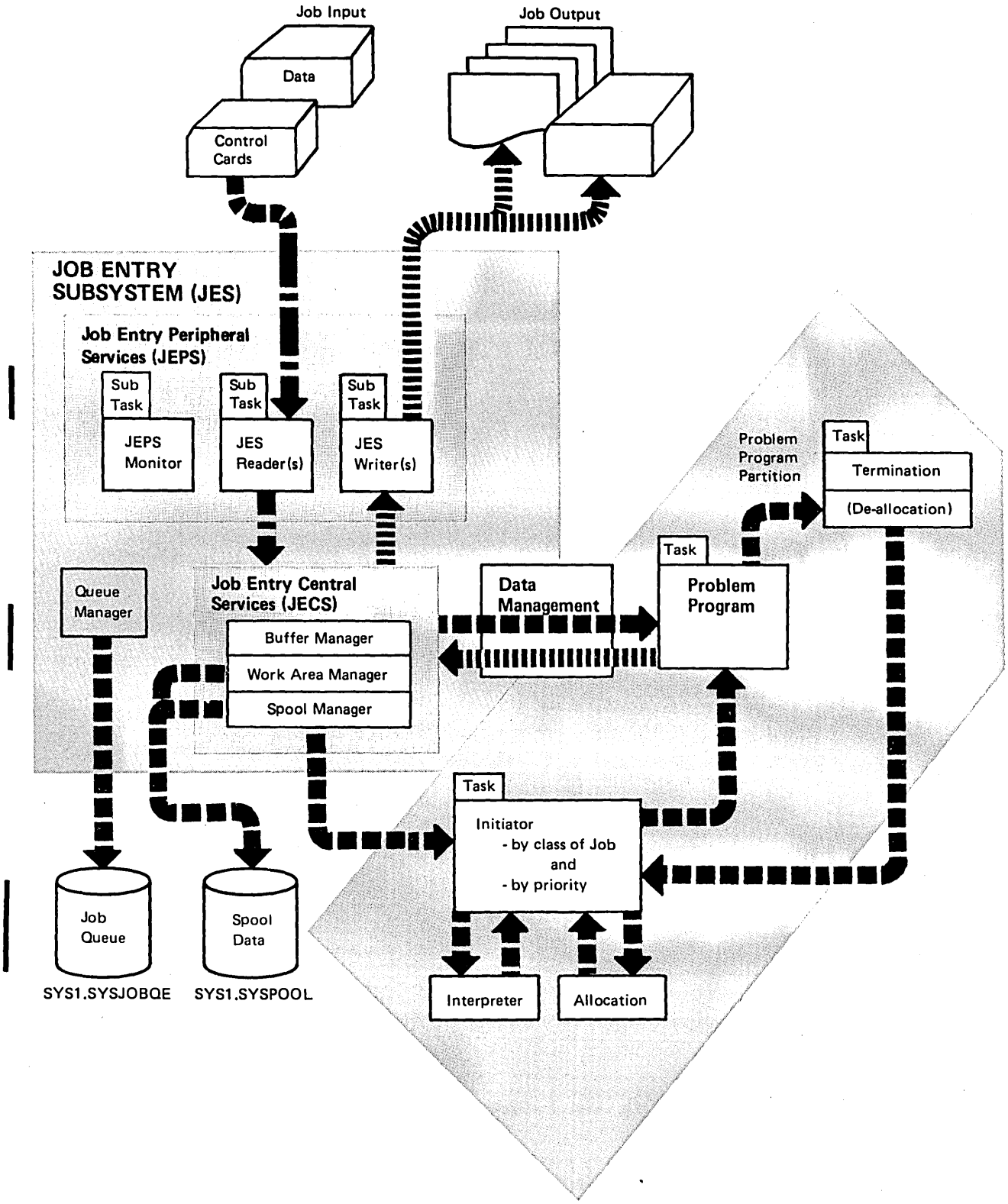


Figure 1-2. Relationship of Job Scheduler Services and Problem Programs

Job Entry Subsystem

The Job Entry Subsystem (JES) is the name given to that portion of job management that spools primary input and output streams and schedules resources. JES performs three basic functions:

1. All primary input streams are read from the input device and stored on a DASD data set (SYS1.SYSPOOL) in a format convenient for later processing by VS1 and problem programs.
2. System (and selected user) printed and punched output is similarly stored on SYS1.SYSPOOL until a convenient time for producing a hard copy on a printer, punch, or tape device.
3. Where there is contention for system resources (for example, buffer assignment), JES schedules the contending activities to assure the greatest system throughput.

The first two functions, basic to JES, are called *spooling,* the advantages of which are:

1. Using nonsharable devices, generally unit-record devices, at rated speeds if enough buffers are available.
2. Using nonsharable devices only for the time required to read, write, or punch the data.
3. Processing jobs or problem-program output in any order. This ability to control system work is called *job queueing.* Jobs can be scheduled by class and by priority within a class.

In VS1 the reading function is not slowed by the interpreter function which operates as a subroutine of the initiator within the problem-program partition. Therefore, there is no requirement in VS1 for transient, floating, resident, or dedicated reader/interpreter or output-writer partitions. This, together with virtual storage, results in improved system work throughput.

The method of handling JCL by MFT and VS1 comprises a basic difference between the two systems. In MFT, the system interprets a job's JCL as it is being read. In VS1, the system interprets the JCL *after* the job is selected for execution.

Job Flow

A job flows through VS1 (Figure 1-2) by passing through two functional areas:

- The Job Entry Subsystem (JES) where it is under VS1 control from the time it is read into the system until it is written out.
- The problem-program area where program instructions are executed.

Before executing a problem program, VS1 requires some direction from the user. He must tell the system about the job and job steps through the job-control language. The job-control language provides the vehicle to supply the system with job and program information, data characteristics, and device requirements at the time the program begins execution.

Reading System Input

After a START reader command, the JES reader reads jobs and data into the system. The reader is a part of the JES component called *Job Entry Peripheral Services* (JEPS).

Next, jobs and data are spooled on a system data set (SYS1.SYSPOOL) and records pertaining to their processing requirements are placed on the system job queue (SYS1.SYSJOBQE). When a job and its data have been completely read, an entry is made into SYS1.SYSJOBQE signifying that it is ready to be selected by an initiator. Based on a priority and class scheme that the user specifies in the job control language, the initiator, which resides in the problem-program partition, will later select a job for execution. Figure 1-3 shows the process of reading and spooling job input and data.

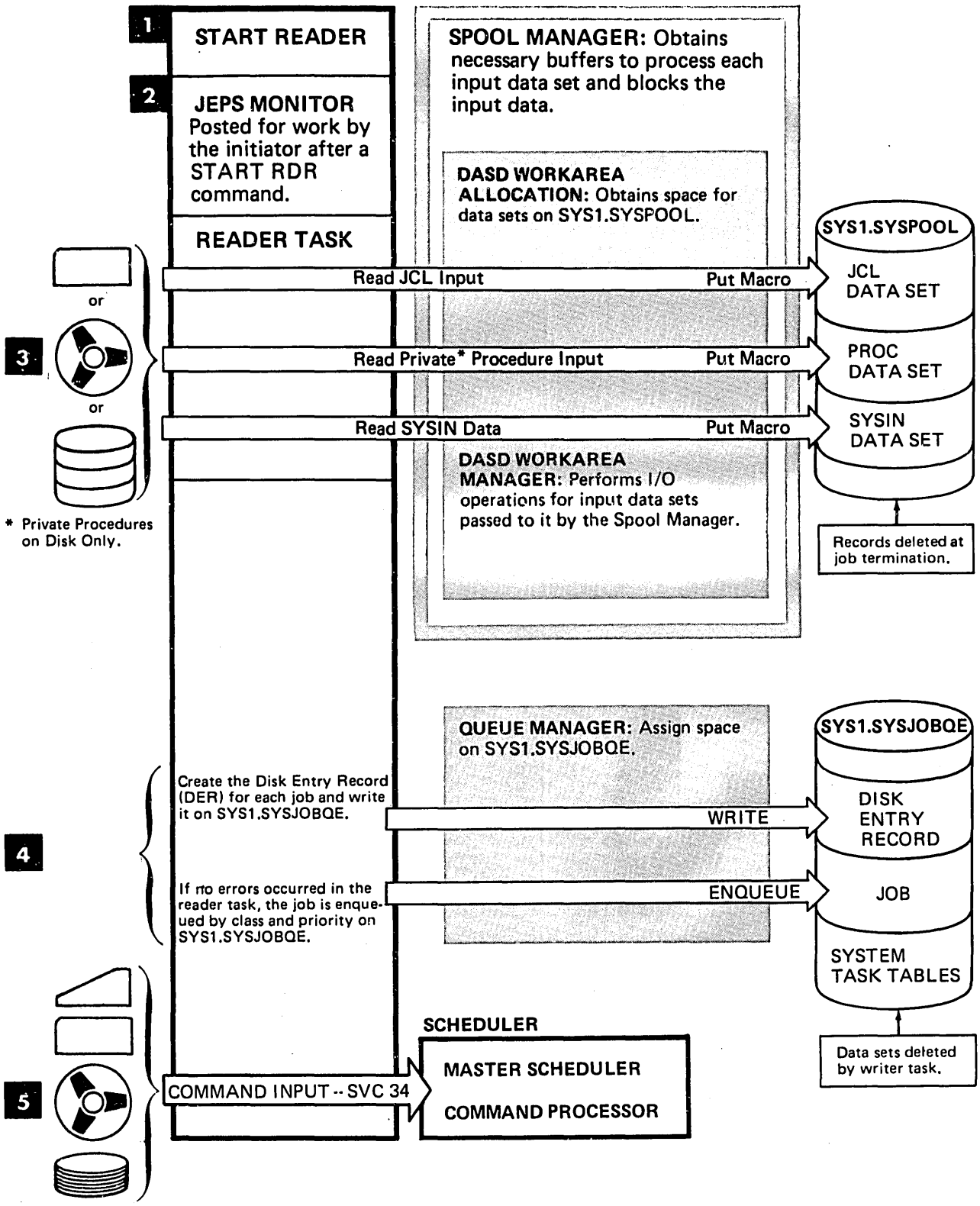


Figure 1-3. Reading and Spooling Job Input

Job and Job Step Processing

Once a problem program is selected for execution, the initiator invokes the interpreter. The interpreter analyzes the job-control language statements for the job and builds a series of tables that describe the required job-processing steps. These tables are then written on the Scheduler Work Area Data Set (SWADS) used by the initiator. The allocation routines then fill any I/O device requirements for the job. Once the initiator has completed its preparation activities, the problem program can begin execution and use the necessary access method and supervisor services for data and program management.

All system output from a problem program is spooled to an output data set on SYS1.SYSPOOL (unless direct system output is specified) and is enqueued on SYS1.SYSJOBQE when the job terminates. A JES writer task processes all output data sets residing on SYS1.SYSPOOL. If direct system output is specified, all output, except system message data sets, goes directly to the specified device and is not spooled to SYS1.SYSPOOL. Figure 1-4 illustrates the steps required to start a program executing.

When a program completes the execution phase, the necessary termination and device deallocation functions are performed. The initiator is then notified that execution of the next program may begin.

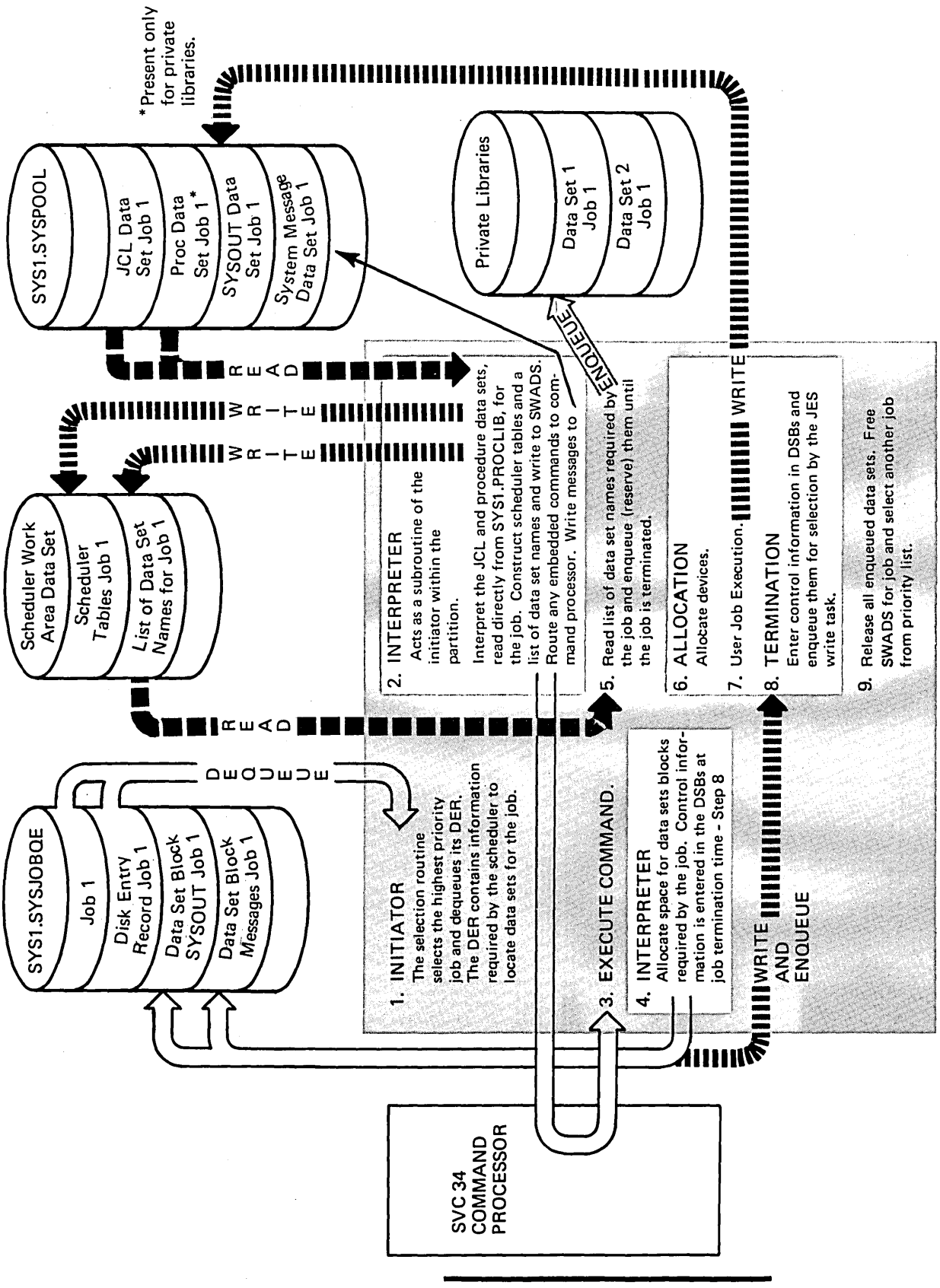


Figure 1-4. System Flow for Selecting a User-Submitted Job on an Input Queue and Starting Program Execution

Writing System Output

System output is divided into one or more classes. For example, there might be one class for printer output, another class for punch output, and a third class for tape output. The two ways to write system output are:

- The operator may use the START command to establish the JES writer task. He may change the set of classes via the MODIFY command.
- The operator may use the START command to indicate Direct System Output (DSO) processing for one or more classes. He may change the set of classes via a MODIFY command or by START or STOP commands.

If DSO processing has been started, a job containing a DSO class writes directly on the appropriate unit record or tape device. The JES writer task writes system output in other classes if a writer is active for that job class.

When the system processes all output for a job, the accounting linkage routine is called for final job accounting, and the job is removed from the system. At this time the space held by the SYSOUT data sets on SYS1.SYSPOOL is released. The JES writer task also provides information about the SYSOUT data set to the SMF accounting routines. Figure 1-5 shows the functions of the JES writer.

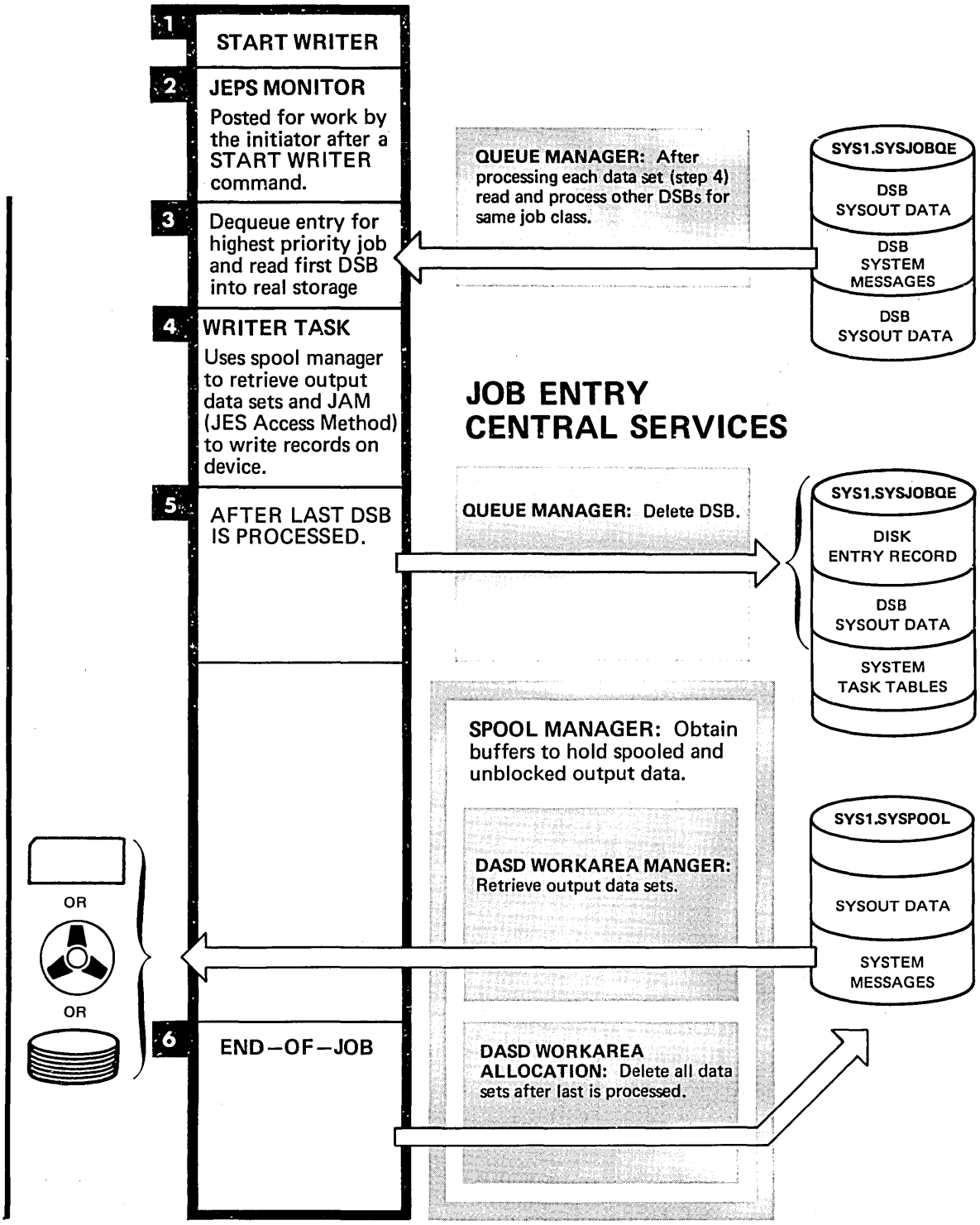


Figure 1-5. Retrieving and Writing a Job's Output

System Restart

System restart allows the user to restart the system after normal end-of-day shutdown or after a system failure without losing all the jobs remaining on the system queues. To invoke system restart, omit the F from the SET parameter, 'Q=(,F)'. Jobs in execution are restarted at the beginning of the current step. A job can restart only if all of the following are true:

- One of its steps was in execution.
- Correct restart parameter was specified in the JCL.
- Authorization is given by the operator.

Active output is re-scheduled from the beginning of the SYSOUT data set being processed at the time of failure.

Jobs that cannot be restarted must be resubmitted.

Automatic Step and Checkpoint/Restart

The restart reader is executed for automatic step and automatic checkpoint/restart processing. This processing occurs for jobs that fail with an eligible abend code, such as jobs with a restart parameter specified in the JCL or a restart specified by the operator.

For automatic step restart processing, jobs are restarted at the beginning of the abnormally ended step. For automatic checkpoint/restart processing, the abnormally ended job receives control at the instruction immediately following the last successful checkpoint within the job step. System restart uses the restart reader to perform automatic step or checkpoint/restart functions for jobs in execution at the time of system failure.

Write-to-Programmer

Write-to-programmer (WTP) allows messages from the problem program or system tasks to be written to the JECS System Message Data Set. The spool manager routines process all I/O for the WTP message set.

System Service Elements

You have read a general description on how a single job flows through this portion of VS1. Now you can look at the various elements as facilities. This part describes those components of VS1 that are responsible for scheduling programs for execution.

Job Entry Subsystem

Figure 1-2 illustrates the various elements that make up JES, which consists of Job Entry Peripheral Services (JEPS) and Job Entry Central Services (JECS). JES reads jobs into the system, divides them into various segments (job control statements and data, for example), and stores them. When the jobs are selected for execution, the JCL is compiled separately. When execution of a job is complete, JES spools the output data set onto a DASD device.

Job Entry Peripheral Services

JEPS consists of a monitor, a JES reader, and a JES writer. Their functions are described here.

JEPS Monitor: The JEPS monitor controls the starting and stopping of all JES readers and writers. The maximum number of JES readers and writers is set at system generation time, but may be overridden at Initial Program Load (IPL) time by altering JESPARMS.

At IPL time, an initialization routine in the JEPS monitor receives control and:

1. Initializes portions of the JES communications table.
2. Waits for posting following a START or STOP command for a reader or writer, or end-of-file condition on a non-unit record device.

When the JEPS monitor initialization is complete, the monitor is ready to receive control whenever a reader or writer is started or stopped.

JES Reader: All JCL and job stream SYSIN data entering VS1 except console-entered commands, is processed by the JES reader task. The START command initializes the JES reader task for each input stream. The reader task continues to service an input device until stopped or until it encounters an end-of-file on a device other than a card reader.

When a JES reader is started, the JEPS monitor initializes work area storage for input stream de-

pendent information and attaches the reader task as a subtask for each input stream started. The reader task examines the input stream records and separates them into commands, job control, and data.

The JES reader task:

1. Scans each job statement for valid jobname, class, type of run, and priority keywords to determine a job's position on the job queue.
2. Obtains a time stamp containing the start time as each job is encountered in the input stream.
3. Gives an internal identification consisting of a number concatenated with the job name.
4. Sends to the command processing section of job management all commands not within a job. The command is processed according to a specified disposition that may:
   a. Execute the command.
   b. Write the command on the console and execute it.
   c. Write the command on the console and ask the operator whether or not to execute it.
   d. Ignore the command.
5. Puts the JCL for the job into a JCL spool data set SYS1.SYSPOOL).
6. Enqueues the job on SYS1.SYSJOBQE.
7. Take a job-end time stamp when the job has been completely read, and calculates the real time in the JES reader for the job. (The total number of statements read for the job is also available at this time.)
8. Complete processing for the current input stream when an end-of-file is encountered on any device other than a card reader. (The JEPS monitor is then invoked to indicate that the reader has been closed.)

JES Writer: The JES writer processes all spooled output (SYSOUT) data with the exception of direct output data. The data includes output data sets and data sets containing system messages, and the job-control statements. A START command creates a writer task for each output stream. It continues to service an output device until terminated by a STOP command.

When a JES writer task is started, the JEPS monitor initializes work area storage for output stream dependent information and attaches the writer as a subtask for each output stream that is started.

The JES writer task uses JECS to obtain output data and system messages in the same order as they appear in the specific output class queue. The writer prints one copy of a job's JCL, system messages, and output data set unless otherwise specified. Multiple copies can be obtained by using the writer command or by specifying "COPIES=" in the JCL.

When all output for a job is processed, an accounting linkage routine is called for final job accounting, and the job is completely removed from the system. At this time, the space held on SYS1.SYSPOOL is freed.

Job Entry Central Services

Spool management and queue management are the two external services residing within JECS that provide management functions for JES.

Spool Management: The spool manager is a central facility that handles sequential data on a logical record basis. It is a block/deblock routine in which the system processes records sequentially. Specific records can also be directly read, thereby automatically switching spool management processing to the new record location within the data set. Records are then processed normally from this point by issuing the standard spool manager request without specifying the record address.

The data handled by spool management includes (but is not limited to) job control language text, in-line input data, procedure libraries, system messages, and spooled output data. This data can be accessed only through the spool manager.

Job Queue Management: The job queue manager is the central facility maintaining the job queues (SYS1.SYSJOBQE) and the SWADS. The job queue manager uses a set of routines that:

1. Assign space to problem programs.
2. Read and write queue records and tables.
3. Enqueue and dequeue work queue entries.
4. Delete work queue entries.

These work queues are the temporary storage areas that maintain information about the job in the system. The SWADS is the auxiliary storage

area for scheduler work tables pertaining to a batch problem program. The corresponding tables required for system tasks reside on SYS1.SYSJOBQE.

Scheduler Work Area Data Set: The SWADS is a work data set that holds the tables created for a problem program. A SWADS is associated with each initiator and is allocated when the START command for the initiator is processed. It is deallocated when the STOP command is processed.

Each SWADS holds the tables for the problem program currently handled by that particular initiator. Thus, it is a sequential data set that is reusable; that is, the tables for a job overlay those from the previous job.

Initiator

When a START command is entered from the console specifying an initiator procedure, the initiating task is established in the specified partition to schedule job execution. The initiator job selection routine then attempts to dequeue the highest priority job from the first job input queue associated with its partition. If there are no jobs to be initiated on any input queues, the operator is notified, and the partition is placed in a wait state.

Once a job is selected, the initiator enqueues on all data sets required by that job. Data set names are obtained from records constructed by the interpreter (see Figure 1-4). If all requested data sets are available, they are reserved for the job. If any data sets are not available, the system notifies the operator and asks him to reply to a system message with either RETRY, HOLD, or CANCEL.

- RETRY: the initiator again attempts to enqueue on the data sets.
- HOLD: the job is placed on the HOLD queue for later release.
- CANCEL: job execution is canceled.

The system fetches a program required for execution from the program library and loads it into virtual storage. Program execution begins after the entire program's relative addresses (created by the linkage editor) are resolved into absolute addresses.

When the final step of the job is terminated, or if it is bypassed, any reserved data sets are returned to the system. The initiator is ready to select another job.

Interpreter: The interpreter operates as a subroutine of the initiator in a partition. It analyzes the contents of job-control statements and builds tables used during the initiation, termination, and execution of job steps.

Allocation: The allocation function operates as a subroutine of the initiator. It analyzes the I/O device requirements of job steps, allocates devices to them, issues volume mounting instructions, and verifies that the volumes are mounted on the correct device. (For magnetic tapes, the OPEN routines verify this.)

VS1 allocation also supports dedicated data sets (workfiles). Dedicated data sets eliminate some of the time required to schedule a job step because these data sets are always allocated.

Termination

The termination function operates as a subroutine of the initiator. When a problem program completes execution, the termination routines free (deallocate) all resources used by the program, perform the data set disposition and termination, and enqueue SYSOUT to allow the system to continue functioning for other problem programs.

Command Processing

Command processing is the reading, scheduling, and executing of commands from the programmer or the operator. These commands can be entered into the system via the console or the input job stream. Some commands must be entered via console only. Others may be entered via the console or the input job stream (Figure 1-6). The functions of these commands, regardless of their source, allow the user to override specifications or values that were specified at system generation time, at IPL time, or by other commands.

Command processors also provide manipulation of control blocks associated with commands. These additional functions are invoked via the QEDIT (CIB) and MGCR (CSCB) macros. The functions available include:

- Adding or removing a CSCB to/from the chain.
- Setting the CIB count in the CSCB.
- Adding or removing a CIB to/from a CSCB.

All MFT commands are available in VS1 with one notable enhancement: the ability for the oper-

ator and RES work-station users to manipulate SYSOUT data sets. This new command facility can only be entered from the console or an RES work-station. It enables the user to:

- Obtain multiple copies of job output on a data set or job basis.
- Immediately stop the job output stream and start writing again from the beginning.
- Forward space or backward space the output data.
- Single-, double-, or triple-space the output.
- Go to the next data set or restart the writing of the current data set.
- Suspend the writing of a job's output data and replace it on the output queue to be written out later.

MFT command facilities available in VS1 include:

- Providing the flexibility to manipulate jobs by displaying the class, priority, and the number of jobs to be processed; suspending the execution of certain jobs or classes of jobs; releasing suspended jobs; direct canceling of a particular job; and changing the priority or class of a particular job. This facility is available to RES work-station users for manipulation of their own jobs only.
- Redefining the size of a partition. In a virtual storage environment the partitions actually reside in virtual storage. Therefore, a change in partition size is made in virtual storage and not in real storage as in MFT.
- Preparing the system for end-of-day shutdown by enabling the operator to save important statistics and data records.
- Modifying certain processing characteristics, such as changing output writer classes and conditions under which the output writer pauses for servicing (this facility is also available to RES users); changing job classes associated with each system initiator; changing the job classes and output classes associated with DSO.
- Starting a job, called via the console, from a procedure library, to interrupt the normal selection of jobs entered via the input job stream.
- Establishing the date, time of day, or the device used as the input work queue and whether this queue is formatted, as well as specifying the location of the library containing certain program procedures (procedure library), and which automatic commands you wish to override.
- Assigning mount and use attributes to volumes.
- Placing input/output devices, other than a communication line, into an online or offline status.

RES users have command facilities that provide for:

- Logging on and off from RES work stations.
- Listing of NOTICES messages.
- Listing of MAIL messages sent to them.
- Sending messages to other RES users.
- Placing messages into the system log.
- Routing their SYSOUT data to other RES users.

When commands are entered within a job, they are executed when that job is selected for execution (see Figure 1-4). When commands are entered between jobs in a job stream, or via the console, they are executed immediately.

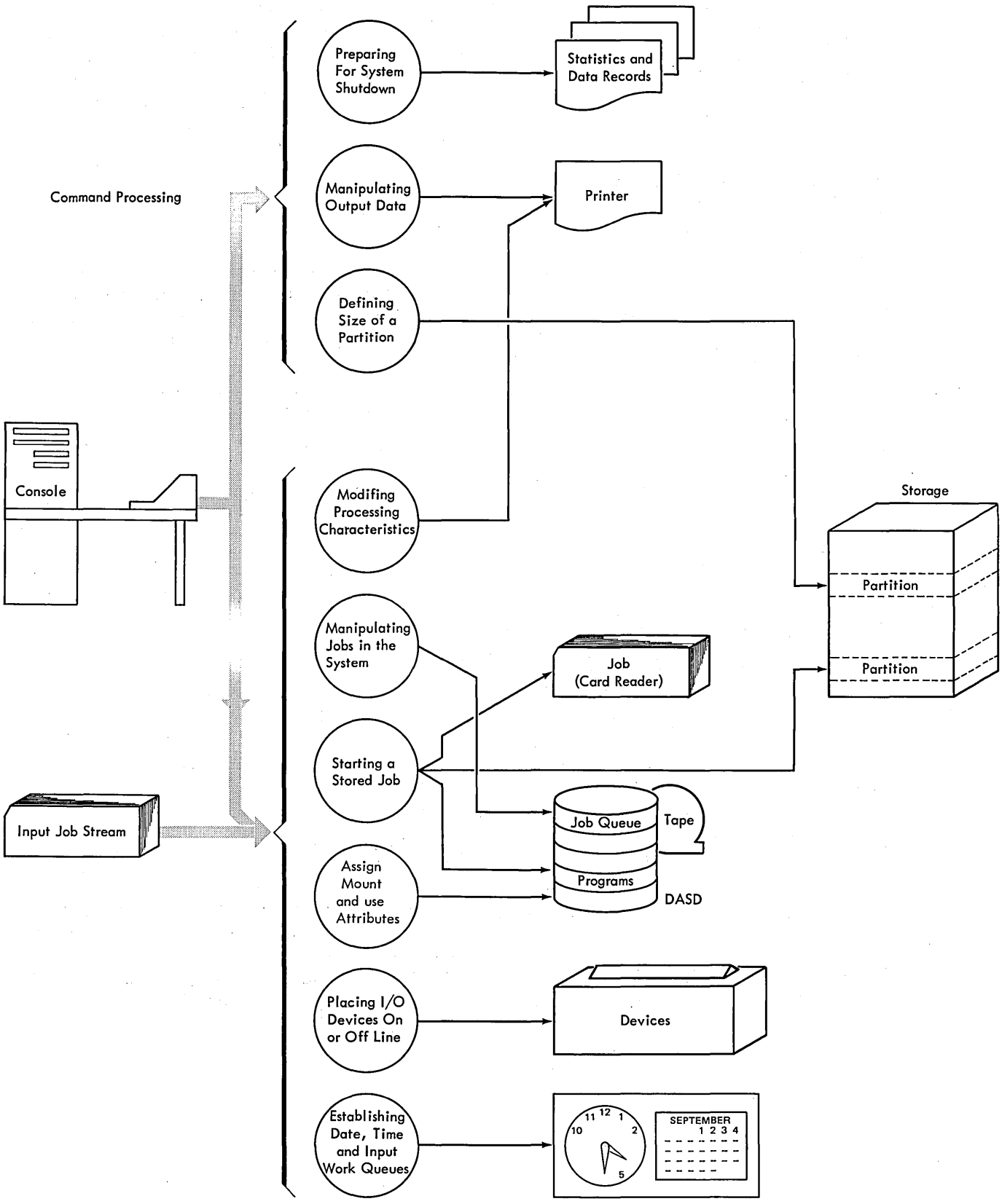


Figure 1-6. Console and Input Job Stream Command Processing

Section 2: Method of Operation

CONTENTS

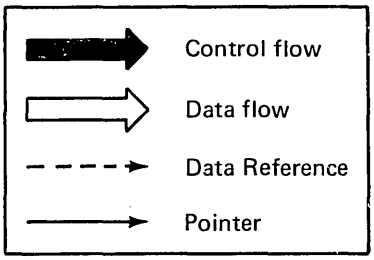
Introduction

This section contains figures organized according to the job management functions. Functional overviews are provided by graphic tables of contents.

Four types of arrows used in these figures are explained in the legend below.

Each diagram has columns labeled *Extended Description, Module, and Figure*. When a step number on the diagram has a corresponding entry in the extended description, this kind of number appears: **3** When no extended description exists, this kind of number is used: **3**

Legend



The module and figure columns are filled in, when applicable. The module column gives the name of the module that performs the activity stated in the extended description.

The figure column references a figure in *Section 3, Program Organization*. The figure details the flow of control that occurs within the function.

A *Dictionary of Abbreviations* appears at the end of this manual.

Figure 2-1. Master Scheduler Initialization (Part 1 of 2)

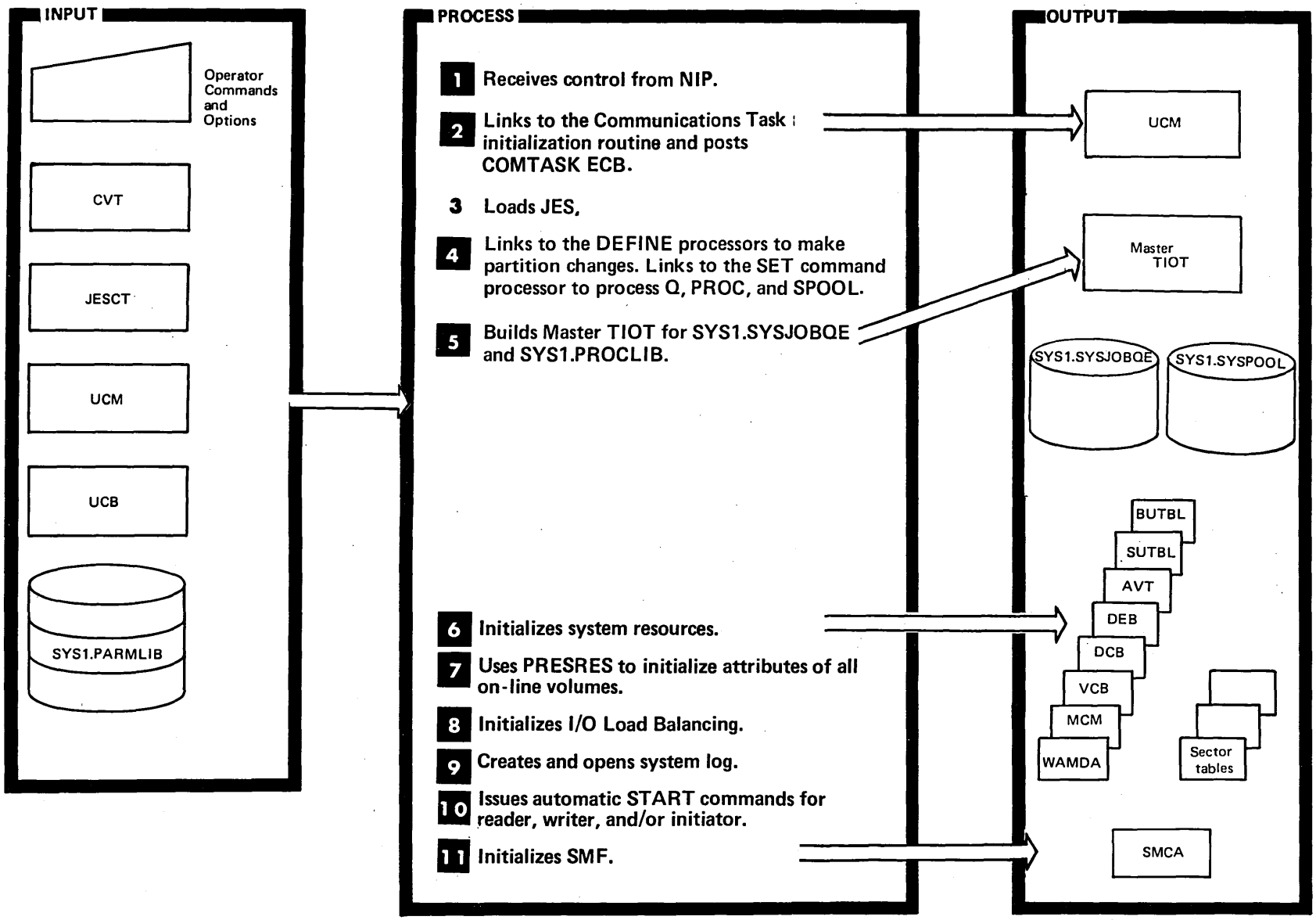


Figure 2-1. (Part 2 of 2)

Extended Description	Module	Figure
1 Module IEFSD569 controls master-scheduler initialization.	IEFSD569	3-56
2 The communications task is initialized because it must be available for console communications during master-scheduler initialization.	IEEVCTI	3-56
4 At IPL, the master-scheduler initialization module links directly to the DEFINE command processor to define the partition configuration. The system then issues a READY message.	IEEDFIN1	3-56
5 The locations of SYS1.SYSJOBQE and SYS1.PROCLIB are verified. SYS1.PROCLIB is cataloged and the Master TIOT is built.	IEFSD569	3-56
6 System resources are initialized by various routines:		
• Work area manager	IEFWARIN	3-56
• Spool manager	IEFSMINT	3-56
• Buffer manager	IEFBMINT	3-56
• Queue manager	IEFSQINT	3-56
7 Volume attribute initialization is based on the PRESRES member of SYS1.PARMLIB.	IEFPRES	3-56
8 The EXCP counts tables are initialized and the EXCP measurement interval is determined.	IEEMB800	3-56
9 The system log is created and opened. The Log Control Area (LCA) is created and initialized.		
10 This is a system-generation option. The commands are expanded and contained within the module itself.	IEFSD569	3-56
11 SMF records for IPL, online I/O, CPU wait time, and dynamic storage configuration are created.	IEFSMF12	3-56
The Systems Management Control Area is initialized.	IEFSMFIT	3-56

Extended Description	Module	Figure

Figure 2-2. Overview of Job Scheduling (Part 1 of 2)

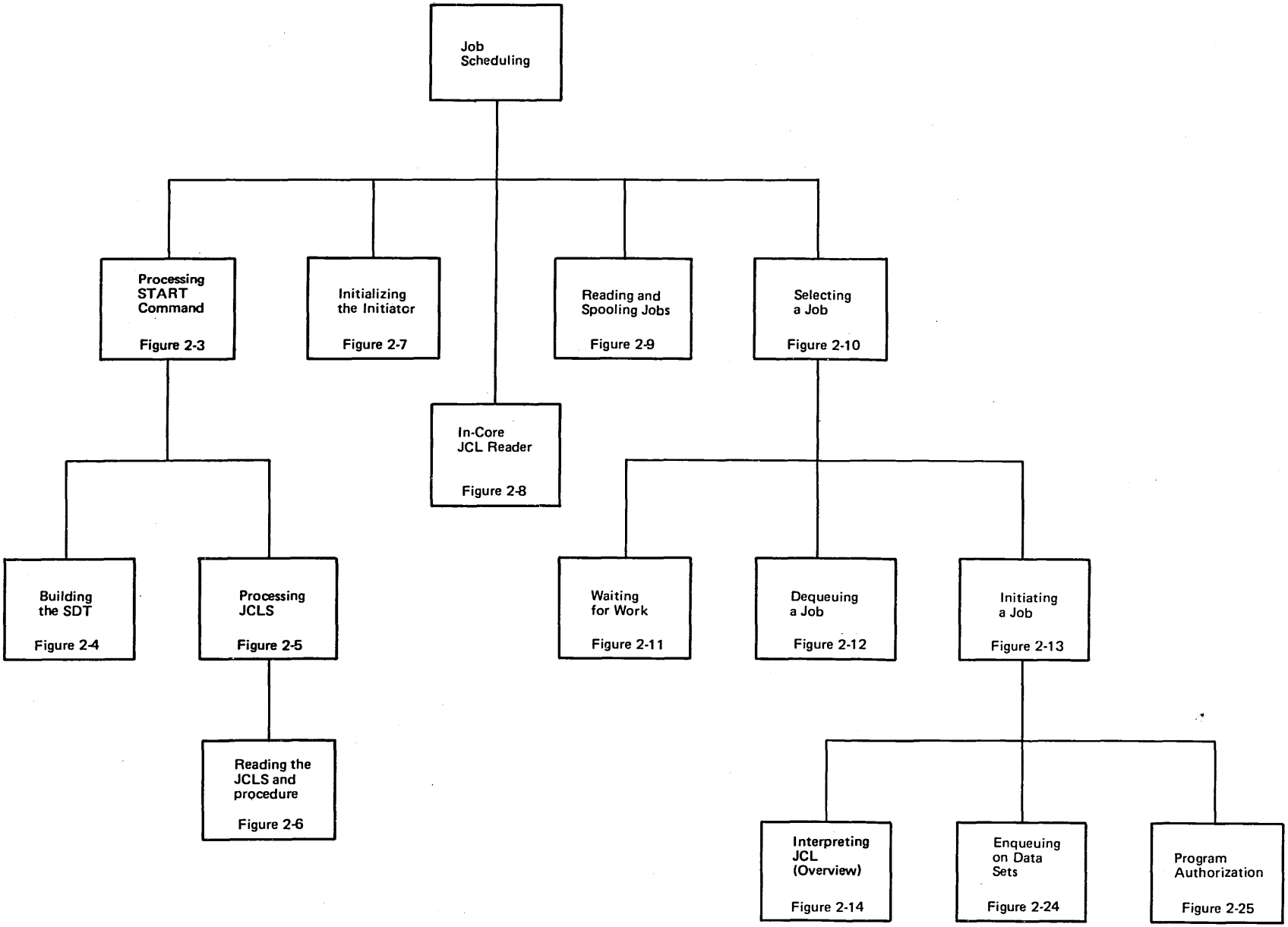


Figure 2-2. (Part 2 of 2)

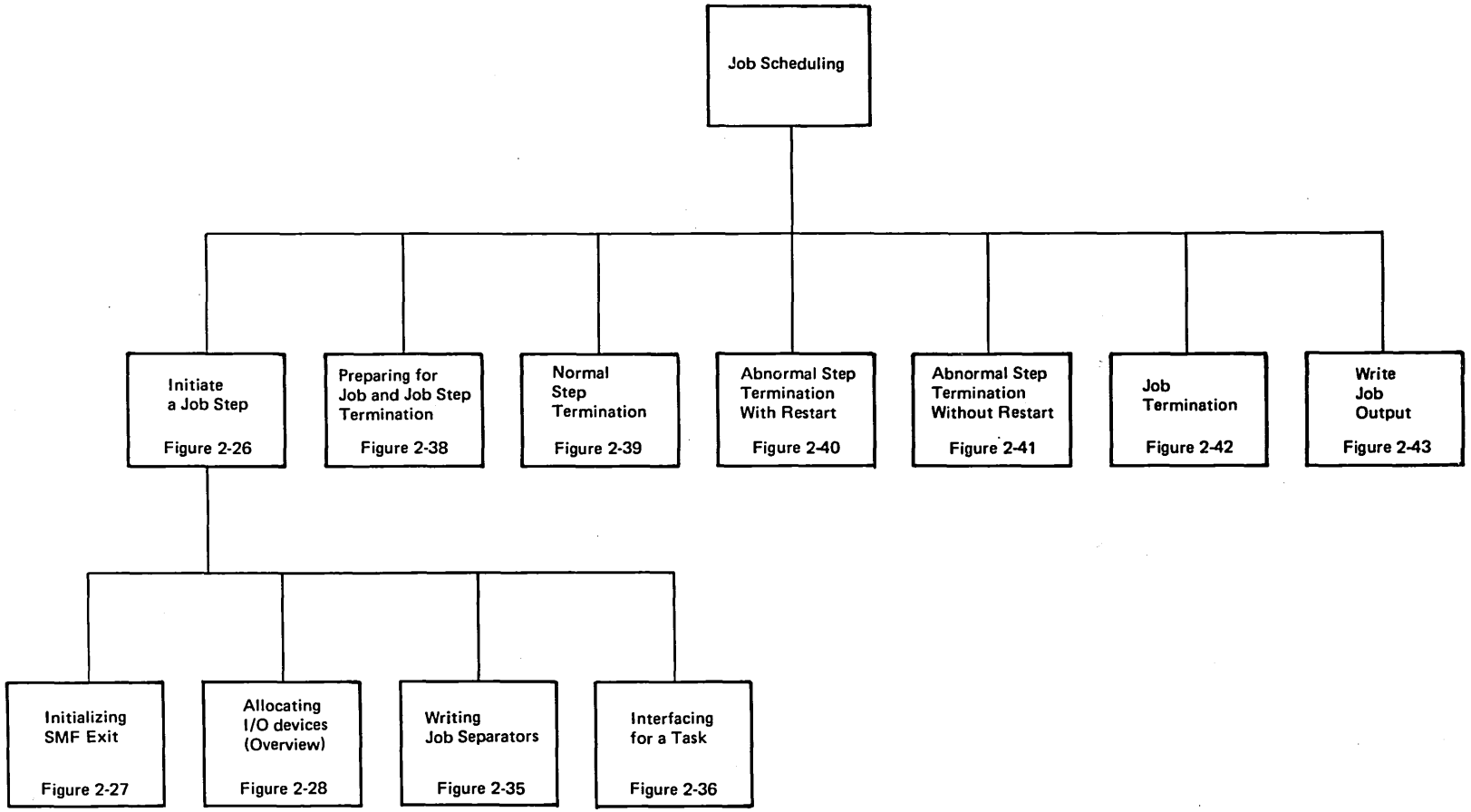


Figure 2-3. Processing the START Command (Part 1 of 2)

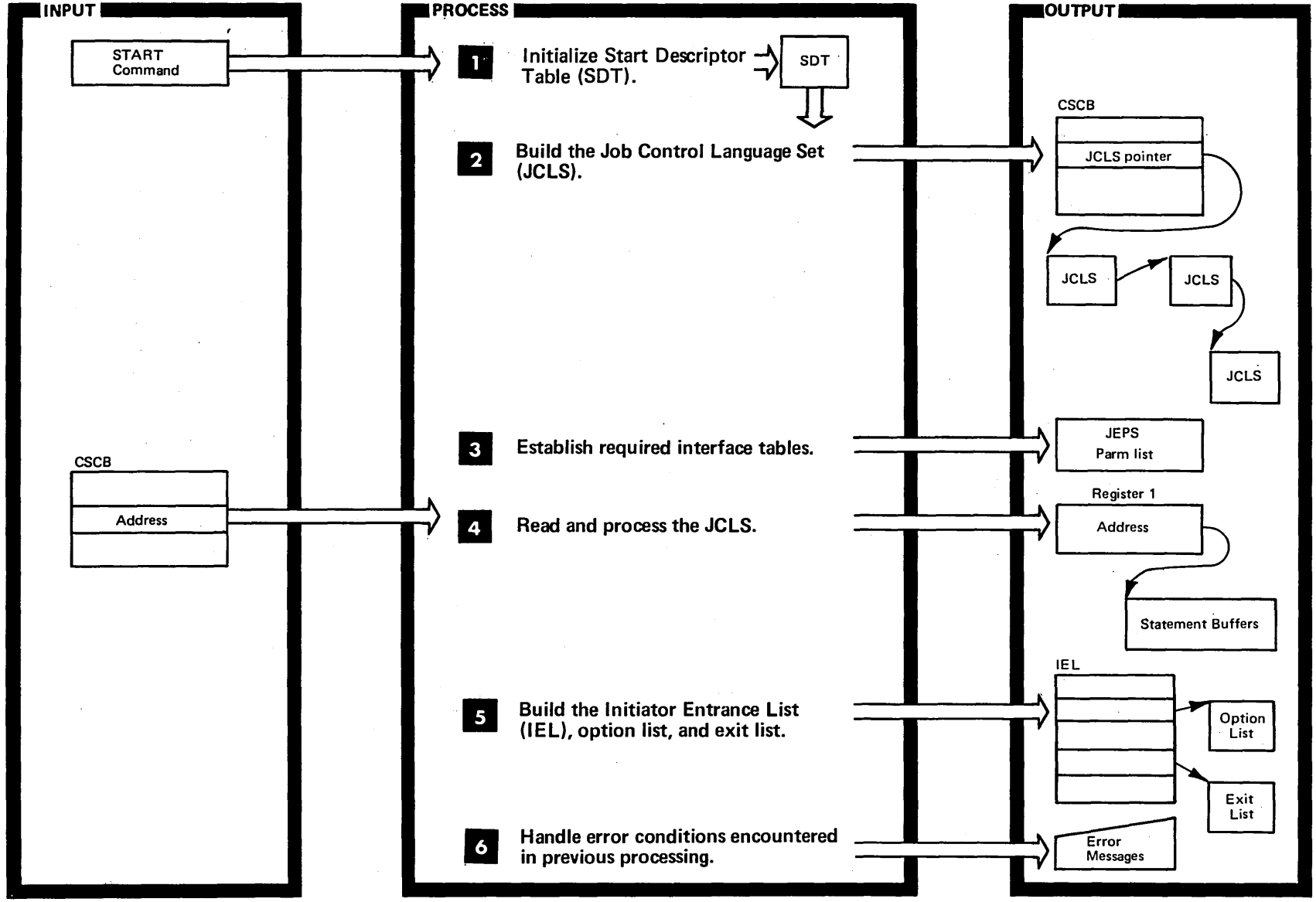
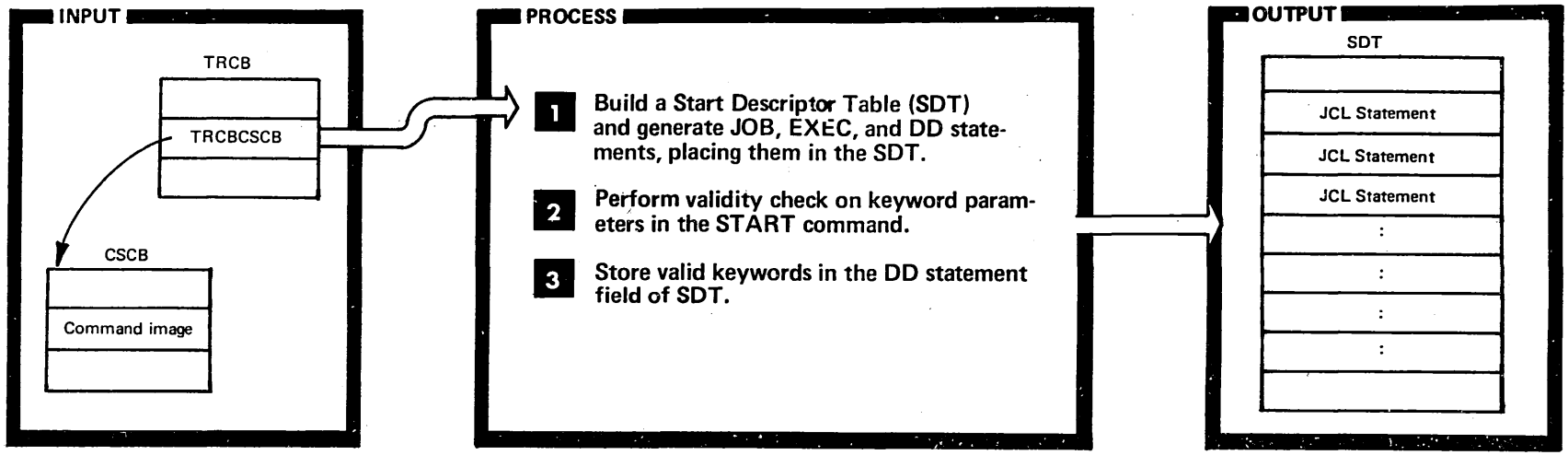


Figure 2-3. (Part 2 of 2)

Extended Description	Module	Figure
1 The START Syntax Check routine builds the SDT from parameters on the START command.	IEEVSTAR	3-11
2 The JCLS Build routine builds in-core JCL statements in 88-character buffers using the information in the Start Descriptor Table (SDT). The Command Scheduling Control Block (CSCB), built by command processing and associated with the START command, points to the first buffer.	IEEVSTAR	3-11
3 The JES Reader Interface routine passes pointers to tables required by the JEPS monitor and the Scan routine (IEEPSN) in a parameter list, and branches (BALR) to JEPS.	IEEVRCTL	3-11

Extended Description	Module	Figure
4 The JES reader (IEFVMC) reads the JCLS and passes procedure statements to the JES Reader Post Scan Exit routine (IEEPSN) which determines the type of task being started.	IEEVRCTL IEEVRJCL IEFVMC IEEPSN	3-11 3-40 3-40 3-40
5 The initiator options list constructed by the Initiator Interface routine contains switches that indicate the processing the initiator is to perform. The Interface routine passes control to the Initiator Initialization routine (IEFSD160) (Figure 2-7). After the started task terminates or is stopped, the initiator returns control to IEEVRCTL which then returns control to the Partition Controller (IEFSD510).	IEEVRCTL	3-11
6 The JES Interface routine handles errors that it finds, as well as errors found by the JES Reader Post Scan Exit routine. It calls the Message Writing routine (IEEVSMSG) to write the error message to the console.	IEEVRCTL	

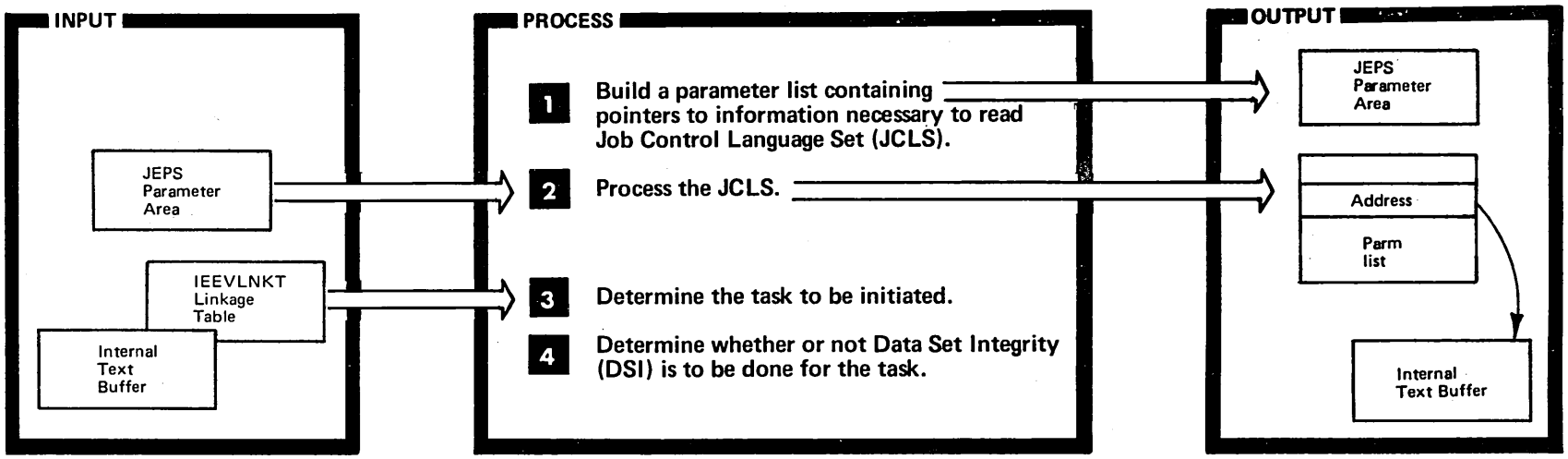
Figure 2-4. Building the SDT



Extended Description	Module	Figure
1 When the SVC 34 routine determines that the command is a START command, it posts the partition controller (IEFSD510) which passes control to the START Syntax Check routine which builds the SDT. The SDT provides up to seven entries which contain: First entry: the JOB statement. Second entry: the EXEC statement that calls the procedure specified in the START command. Remaining entries: a DD statement and continuations of the EXEC and DD statements.	IEEVSTAR	3-11

Extended Description	Module	Figure
2 The Syntax Check routine compares keyword parameters in the START command with a list of allowable DD statement keyword parameters. If a keyword does not correspond to the list of allowable parameters, the routine assumes a symbolic parameter keyword. It does not check DD subparameters.	IEEVSTAR	3-11
3 The DD statement stored in the SDT overrides the IEFRDER DD statement in the procedure specified in the START command. A symbolic parameter keyword is placed in the EXEC statement in the SDT.	IEEVSTAR	3-11

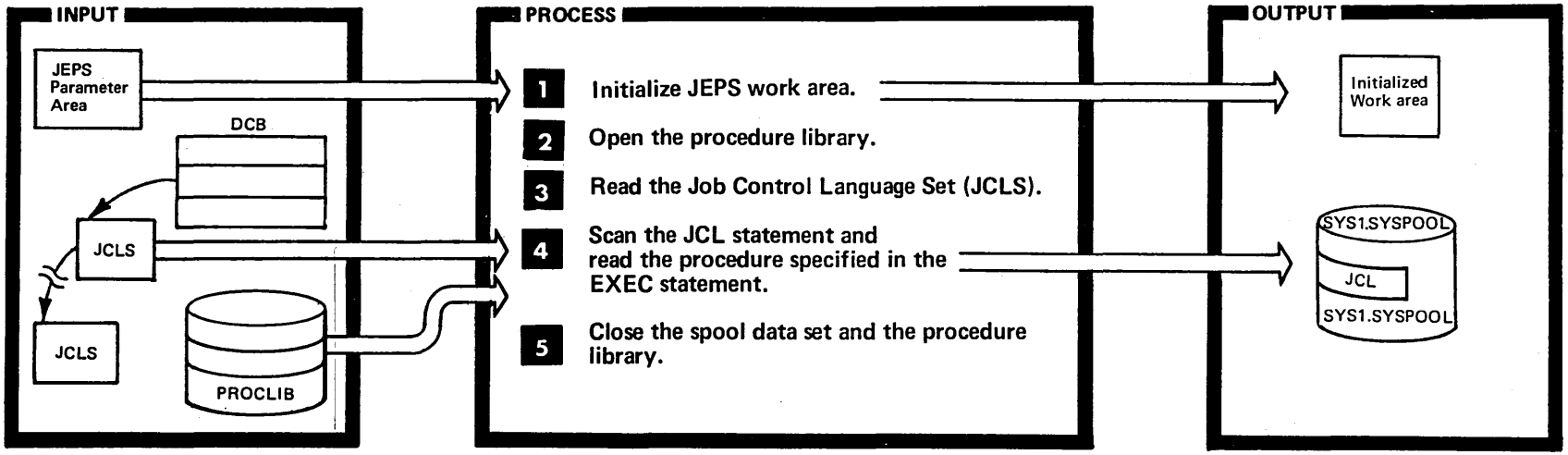
Figure 2-5. Processing the JCLS



Extended Description	Module	Figure
1 The JEPS Interface routine builds the parameter list used by the JES reader to read the JCLS.	IEEVRCTL	3-11
2 The JES reader processes a statement from the JCLS. It determines and reads the procedure specified and passes each logical record from the procedure to the Post Scan Exit routine (IEEPSN).		

Extended Description	Module	Figure
3 The Post Scan Exit routine searches the internal text buffer for an EXEC statement entry that specifies the program (PGM) name. It then compares the name against a list in the PGM name table containing names of all valid system tasks initiated by a START command. If the program name is not found, it is assumed to indicate problem program initiation. A "valid" name found in a multistep job causes an error return code to pass to the JEPS interface routine. Flags are set in the IEEPSN work area to indicate which task is being started: a system task, problem program, initiator, DSO, reader, writer, Generalized Trace Facility (GTF), or RTAM.	IEEPSN	3-39
4 The Post Scan routine checks a table containing a list of all tasks for which data set integrity processing is not to be done. If the name of the task is found, a 'no DSI' indicator is set in the IEEPSN work area.	IEEPSN	3-39

Figure 2-6. Reading the JCLS and Procedure



Extended Description		Module	Figure
1	The system task control routine (IEEVRCTL) passes control to the JEPS monitor. The JEPS monitor obtains a work area and initializes it with information from the JEPS parameter area. It also sets up a pointer to the JCLS in the Data Control Block (DCB).	IEFVMF	3-40
2	The JEPS monitor opens the procedure library (PROCLIB).	IEFVMF	3-40
3	The in-core JCL reader passes the in-core JCL statements to the JES reader.	IEEVRJCL	3-40

Extended Description		Module	Figure
4	The JES reader processes the statements as follows: JOB statement: the reader obtains a unique job number and opens a JCL data set on SYS1.SYSPOOL. EXEC statement: the reader reads the procedure and passes each statement to the Post Scan Exit routine (IEEPSN) for validity checking (see Figure 2-5). Other statements: the reader spools them to the data set on SYS1.SYSPOOL.	IEFVMB	3-40
5	When the reader detects the end of the JCLS, it closes the data set and returns to the monitor which closes the procedure library and passes control back to the system task control routine (IEEVRCTL).	IEFVMB IEFVMF	3-40

Figure 2-7. Initializing the Initiator (Part 1 of 2)

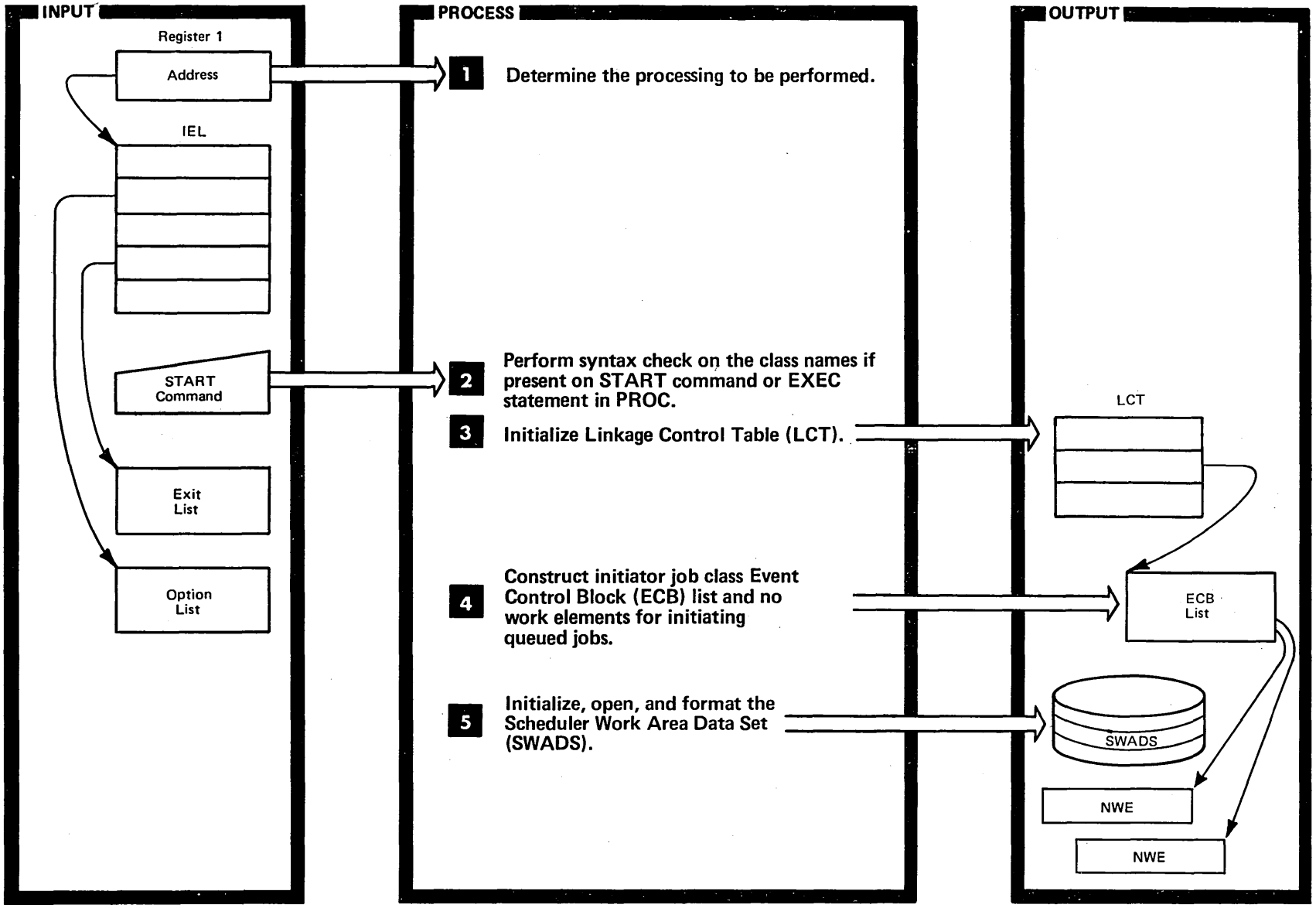


Figure 2-7. (Part 2 of 2)

Extended Description	Module	Figure
1 The Initiator Initialization routine enters from the JES Reader Interface routine (IEEVRCTL) to use the initiator as a subroutine to initiate a system task or a problem program started from the console. It also enters from the Initiator Interface routine (IEFIIC) to initiate queued jobs. The routine determines the necessary processing using information in an Initiator Entrance List (IEL), an initiator options list, and, when entered from IEEVRCTL, an initiator exit list. When the MODIFY command is issued to the initiator, the Job Selection routine (IEFSD161) calls the initialization routine to reinitialize the Life-Of-Task block (LOT) and build a new Event Control Block (ECB) list.	IEFSD160	3-2
2 If the class names are not on the START command, they are taken from the Partition Information Block (PIB). If the routine finds a syntax error, the initiator task is terminated.	IEFSD160	

Extended Description	Module	Figure
3 The initialization routine initializes the Linkage Control Table (LCT) with the queue manager parameter area (QMPA) pointers, Unit Control Block (UCB) pointer, the initiator options from the initiator options list, and the address of the initiator exit list (*only* when called by IEEVRCTL). It also initializes the LCT with the addresses of the Disk Entry Record (DER) and the Command Scheduling Control Block (CSCB) if the initiator is initiating a system task, and the return address of the SVC 3 exit routine when called by IEFIIC.	IEFSD160	
5 The initiating task terminates if the OPEN or the SWADS format is not successful, or if the format routine returns a code indicating that the device allocated to SWADS was not direct access. The initiator routine places a pointer to the ECB list in the LCT and passes control to the Job Selection routine.	IEFSD160 IEFINTQA	

Figure 2-8. In-Core JCL Reader (Part 1 of 2)

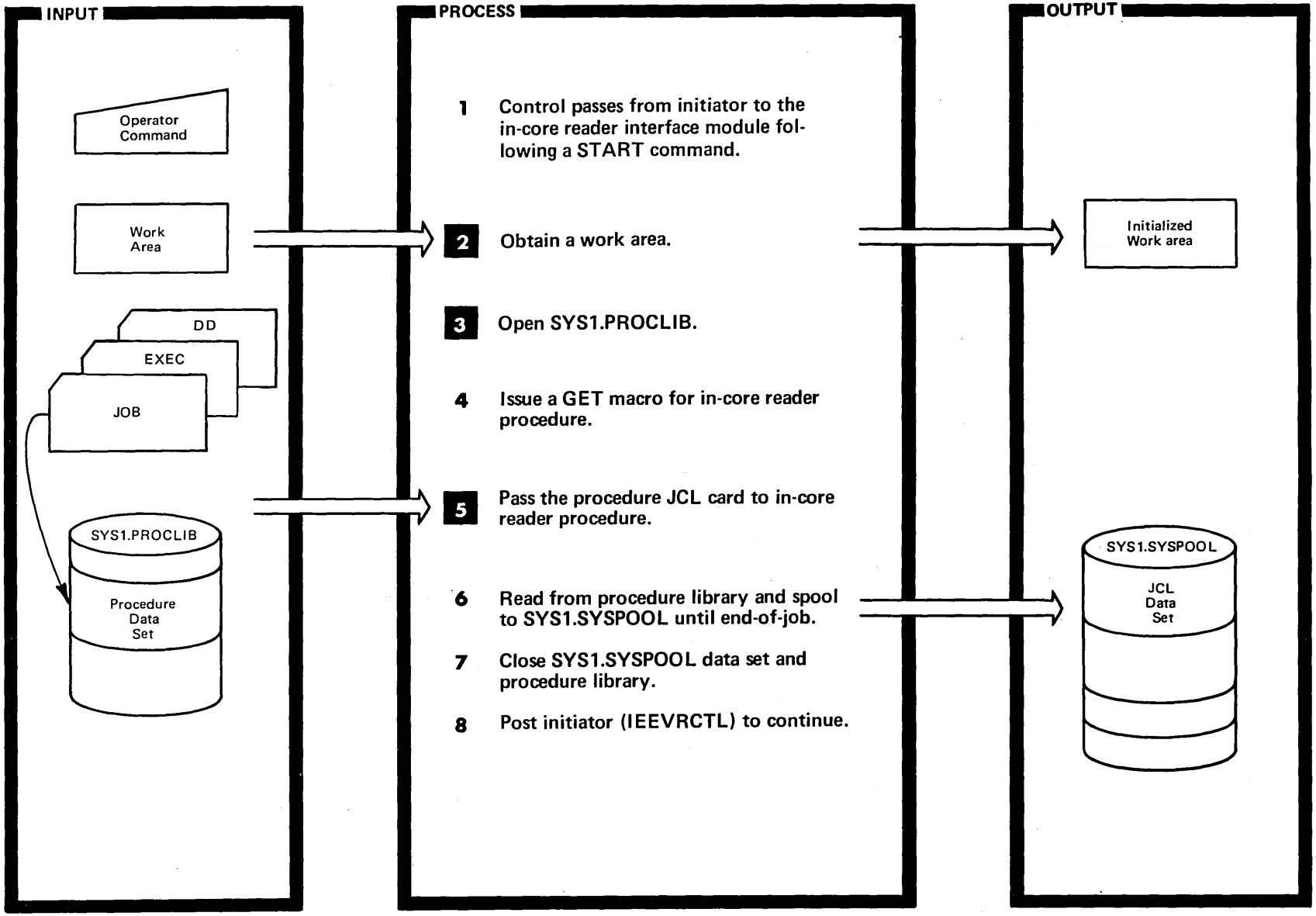


Figure 2-8. (Part 2 of 2)

Extended Description	Module	Figure
2 The in-core reader interface module obtains a work area and initializes it with information in the initiator parameter area.	IEFVMF	3-40
3 The procedure library is a partitioned data set containing JCL for procedures that are frequently used.		
5 If the input card is a JOB card, obtain a unique job number and open a JCL data set on SYS1.SYSPOOL.	IEFVMB	3-40
If the input card is an EXEC card, issue a READ macro instruction to the procedure library. Control then passes to the scan routine (IEEPSN).	IEFVMC	3-40
If the input card is other than a JOB or EXEC card, spool the statement to SYS1.SYSPOOL data set.		

Figure 2-9. Reading and Spooling an Input Job (Part 1 of 2)

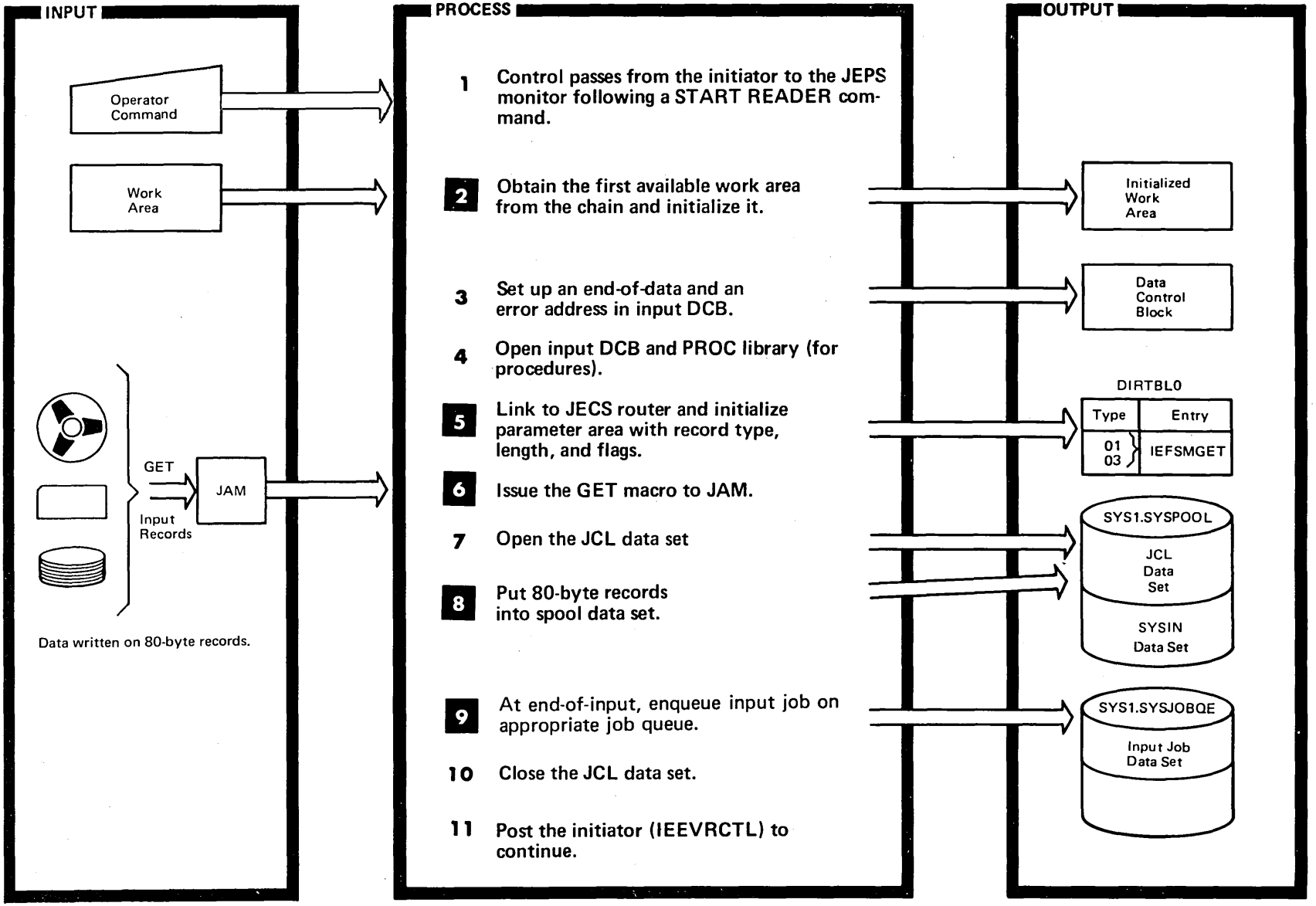


Figure 2-9. (Part 2 of 2)

Extended Description	Message	Label
2 JEPs monitor obtains a work area and initializes it with information in the initiator parameter area. The default parameters in the reader EXEC statement are checked. If spooled procedures are required, post bit in reader work area.	IEFVMF IEFVME	3-40 3-38
5 Input to JECS Router: Register 1 points to RPL; RPL points to LRCB.	IEFSMIFC	3-47
6 The GET macro uses the JES Access Method (JAM) to obtain an 80-byte record from the input device and pass it to the reader procedure (IEFVMB).	IEFVMB	3-41

Extended Description	Message	Label
8 The spool manager obtains the necessary buffers to process each input data set, and blocks the input data. DASD workarea allocation obtains space on SYS1.SYSPOOL for the data set. The DASD work area manager performs the necessary I/O operation.	IEFSMPUT	3-50
9 The job is enqueued according to CLASS and PRIORITY as specified on the JOB card, or according to the defaults if CLASS and PRIORITY are not specified.	IEFJESQM	3-54

Figure 2-10. Selecting a Job

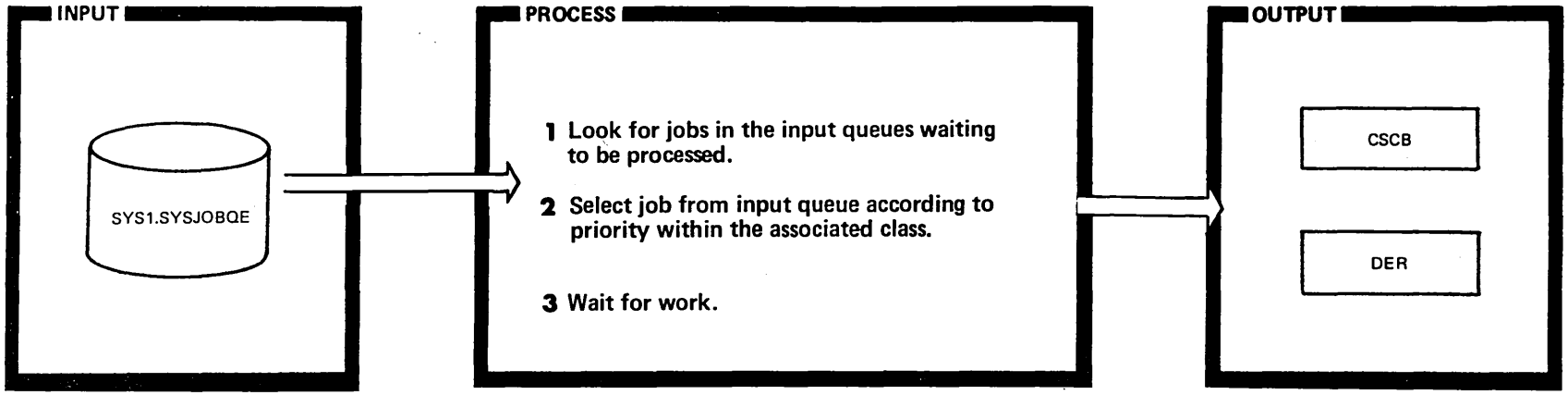
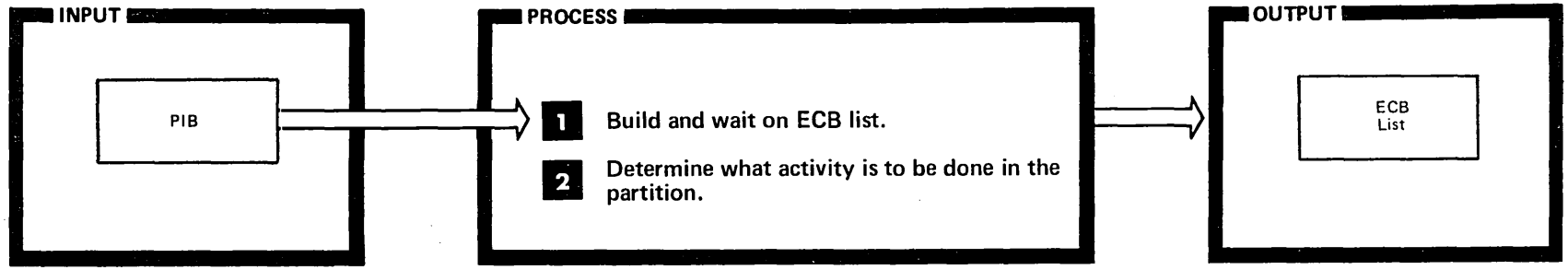
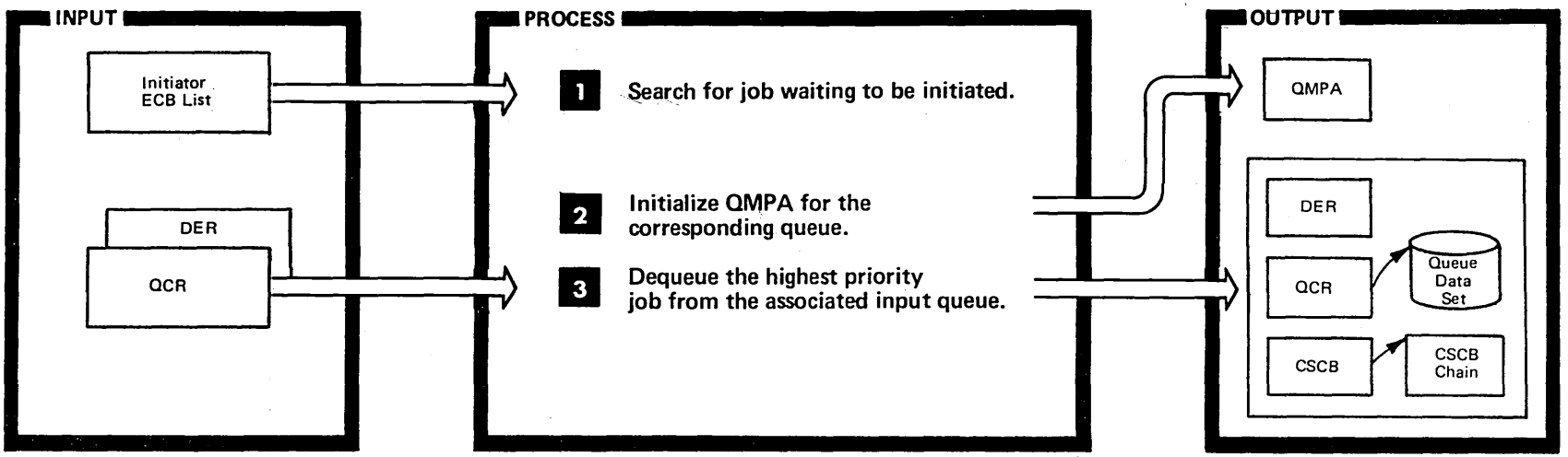


Figure 2-11. Waiting for Work



Extended Description	Module	Figure
1 When scheduling is complete for any job or command, the Job Selection routine (IEFSD161) or the JEPS Interface routine (IEEVRCTL) passes control to the Partition Controller (IEFSD510), which waits for an Event Control Block (ECB) in the initiator's ECB list to be posted. The first ECB in the list is posted by SVC 34 for a STOP or MODIFY command, by JEPS for a STOP READER/WRITER, and by any module that calls IEFSD510. The other ECBs, the job class ECBs, are posted by the Queue Management Enqueue routine (IEFQMJ02) when a job is enqueued on the corresponding queue.	IEFSD161 IEFSD510 IEFQMJ02	3-54
2 After being posted, IEFSD510 checks for: • restart reader to be run • partition to be redefined • STOP issued for reader or writer • STOP or MODIFY issued for initiator or DSO • START issued for reader or writer • MOUNT command • START for initiator, DSO, generalized START, and any START issued to a partition • checkpoint/restart job on internal hold queue • job class ECB posted It goes to the appropriate module if any tests are positive; otherwise, it issues the message, IEF005E.	IEFSD510	

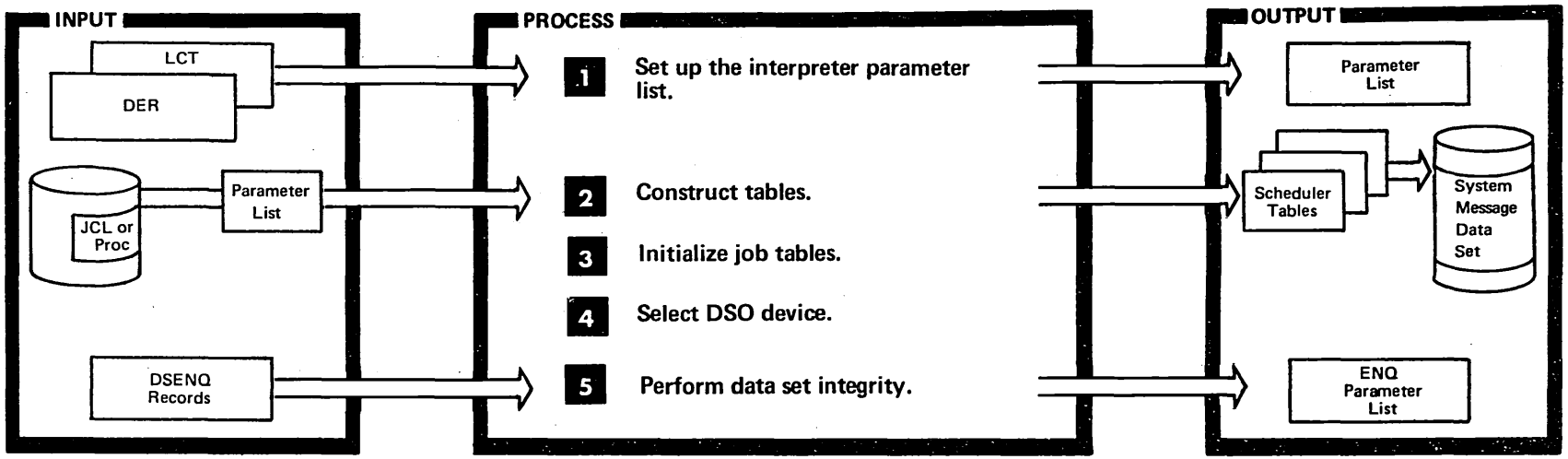
Figure 2-12. Dequeuing a Job



Extended Description		Module	Figure
1	The Job Selection routine scans the initiator job class Event Control Block (ECB) list to determine which job class ECB is posted.	IEFSD161	
2	When it finds the posted job class ECB, the Job Selection routine builds a queue manager parameter area (QMPA) for that queue, giving queue management information necessary to dequeue a job.	IEFSD161	

Extended Description		Module	Figure
3	The Queue Management Dequeue routine issues an ENQ macro and reads in the queue control record (QCR). However, if there are no jobs in the queue. the routine adds the appropriate ECB in the initiator ECB list to the no-work chain, issues the DEQ macro, and returns to searching for jobs (step 1). If a job is found, the priority pointers in the input QCR are updated, a CSCB is created and chained, the DEQ on the QCRS is issued, and the DER is read into real storage.	IEFJESQM	3-54

Figure 2-13. Initiating a Job



Extended Description	Module	Figure
1 After the Job Selection routine dequeues a job, it initializes the parameter list with information necessary for the construction of tables. It then passes control to the interpreter.	IEFSD161	3-4
2 The interpreter uses the parameter list passed by the initiator to locate the spooled JCL data set and the procedure data set, if present. This data set information is used to construct the scheduler tables for the job.	IEFIRC	3-4
3 The Job Selection routine uses queue management to read in the appropriate tables for job initiation.	IEFSD161 IEFJESQM	3-4

Extended Description	Module	Figure
4 The DSO Selection routine determines whether or not a DSO device processing the job's output class is available by comparing the classes with those classes in the Direct System Output Control Block (DSOCB). If a device is available, it is selected for the job.	IEFDSOSL	
5 If data set integrity is required, the DSENQ records are read in, the ENQ parameter list is constructed, and the data sets, if available, are reserved.	IEFSD161 IEFMF102	3-7

Figure 2-14. Overview of Interpretation

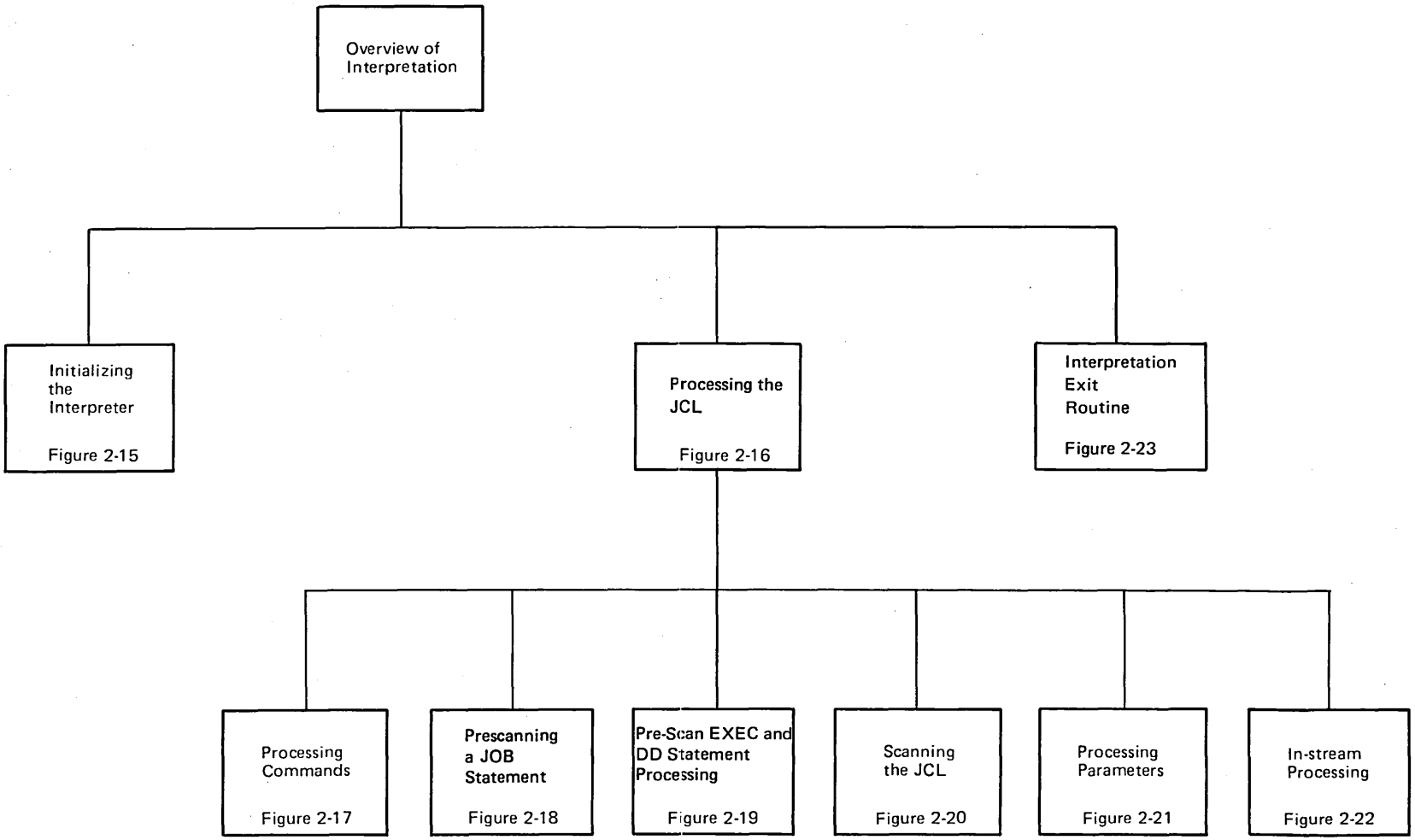
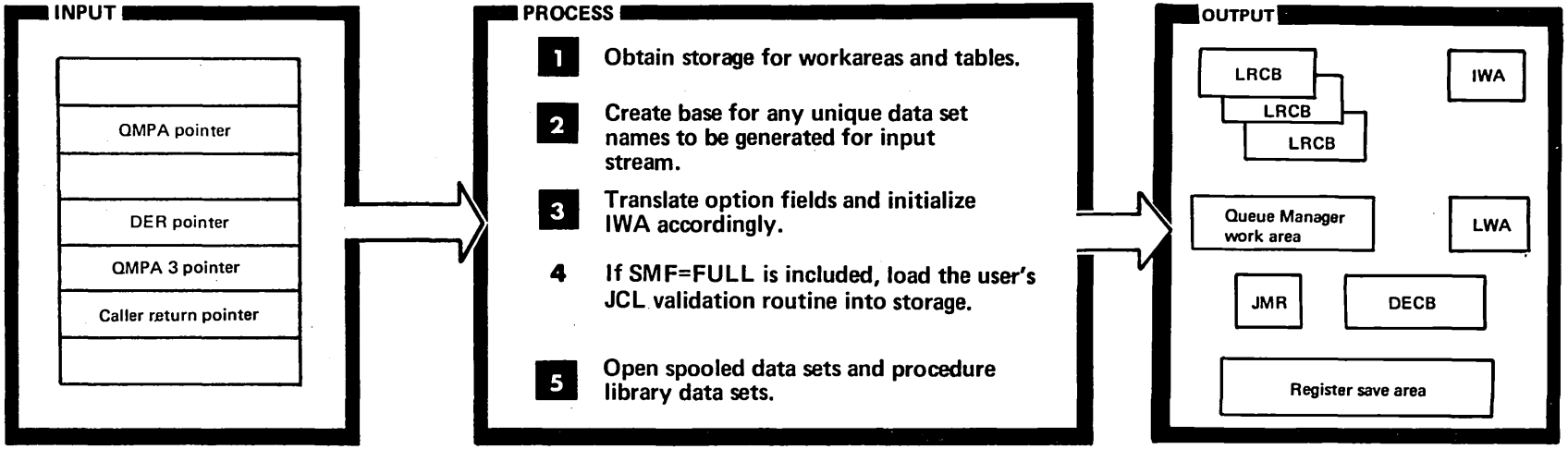


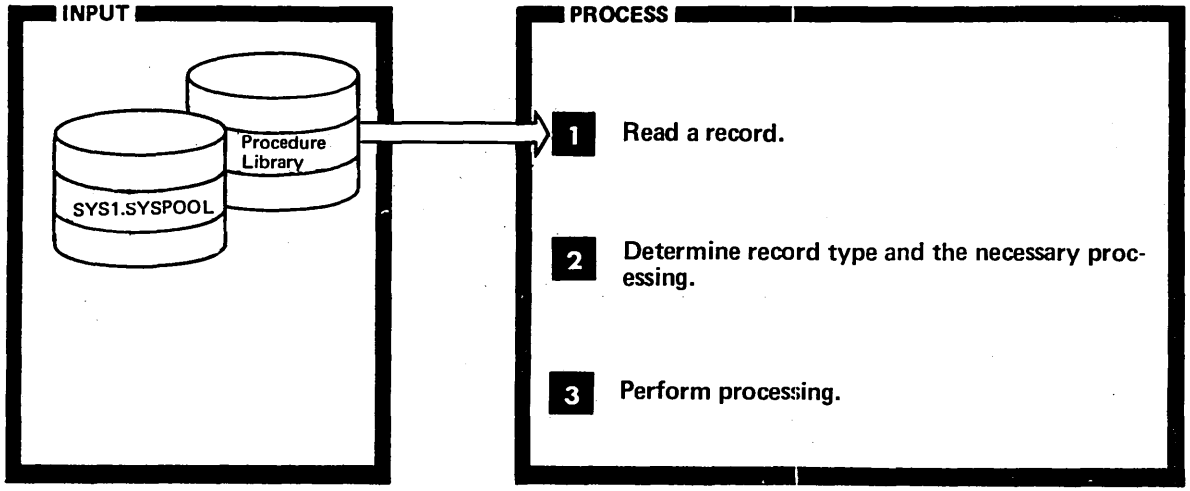
Figure 2-15. Initializing the Interpreter



Extended Description	Module	Figure
1 When the Interpreter Initialization routine is entered, it obtains storage for the Interpreter Work Area (IWA) and the Local Work Areas (LWA). The IWA contains information shared by two or more interpreter routines. The LWA is mapped by each individual routine using it and contains information that need not be preserved outside the routine. The initialization routine also obtains main storage for the blocks shown under output. Storage for a DCB, IOB, DEB, and dummy TIOT is also obtained if a private procedure library is not specified.	IEFVH1	

Extended Description	Module	Figure
2 The initialization routine issues a TIMER macro to create the base.	IEFVH1	
3 The initialization routine sets up a unique name which may be generated during interpretation, converts the EXEC parameter list of the RDR procedure (found in the disk entry record) to binary, and stores this information in the IWA to be used as JCL default values.	IEFVH1	
5 If a private procedure library is specified in the reader procedure, the procedures have been spooled. The initialization routine opens the procedure library if a private procedure library was not specified.	IEFVH1	

Figure 2-16. Processing the JCL (Part 1 of 2)



Extended Description	Module	Figure
1 The Interpreter Get routine (IEFVHA) uses the READ macro to obtain records from the procedure library, or, if an alternate procedure library is used, it uses JECS macros to read from the spooled data set. An end-of-data condition causes control to pass to the Continuation Check routine (IEFVHC) which passes control to the Verb Identification routine (IEFVHCB). When this module recognizes the end-of-data condition, it passes control to the Null Statement routine (IEFVHL) which examines the conditions under which it was entered and passes control accordingly.	IEFVHA IEFVHCB IEFVHL	3-15 3-16 3-17 3-18 3-19 3-20

NULL statement represents	Additional records to process	Control passed to
End of input stream	from procedure (procedure record not in buffer)	Read (IEFVHA)

Extended Description	Module	Figure

End of procedure	from input stream	Verb Identification to process current record from input stream
No more records		Job Validity Check
No more input stream records	from procedure (procedure record already in buffer)	Router routine

When the job is complete and all tables are written, control passes to the Interpreter Termination routine (Figure 2-22). — IEFVHN

Figure 2-16. (Part 2 of 2)

Extended Description	Module	Figure
2 When the Get routine encounters a record containing // in the first two positions, control passes to the Continuation Check routine (IEFVHC). If the record is expected to be a continuation of the preceding statement, control passes to the Prescan Preparation routine (IEFVHEB). If the record is a comment statement, control passes to the interpreter Get routine (IEFVHA). Otherwise, control passes to the Verb Identification routine (IEFVHCB).		
3 The Verb Identification routine identifies the type of statement being processed and passes control accordingly.	IEFVHCB	3-14 through 3-21

Statement Type	Action Taken	Control passed to:
JOB		Router
EXEC		Router
DD		Router
NULL		NULL statement
PROC (first statement in in-stream procedure)		In-stream Procedure Router

(Continued)

Extended Description	Module	Figure

Statement Type	Action Taken	Control passed to:
PROC (first statement in cataloged procedure		Pre-scan Preparation
PROC (not first statement)	Set job-failed bit in JCT. No further processing performed on statement.	Interpreter Get
PEND (not part of in-stream procedure)	Put out error message.	Interpreter Message
PEND (part of in-stream procedure)	Statement converted to NULL statement.	Continuation Check
Command		Command routine

Figure 2-17. Processing Commands

PROCESS

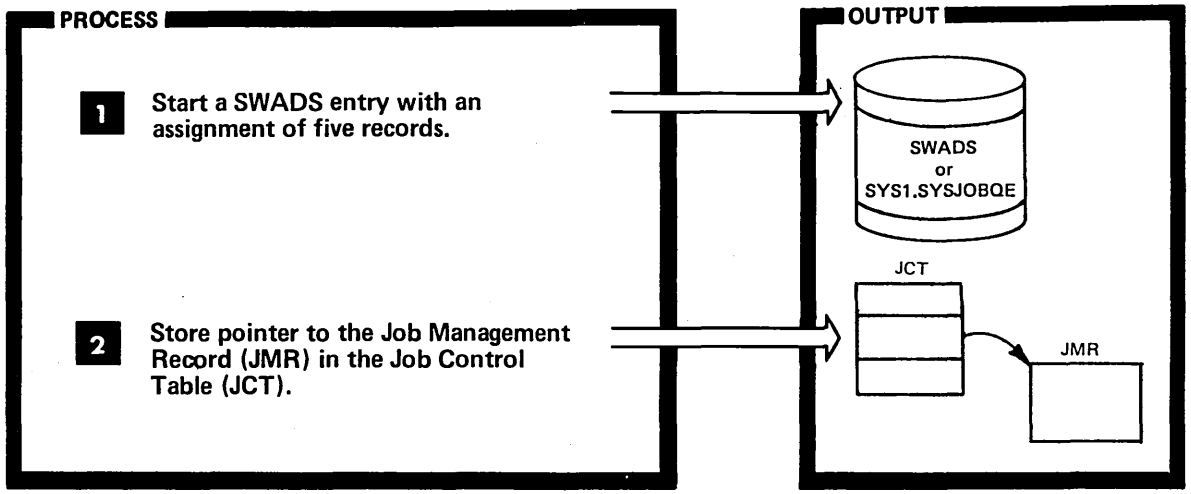
1. **Verify that verb is allowed in the input stream.**
2. **Determine the disposition and process accordingly.**

Extended Description	Module	Figure
1 The Command routine checks the command verb against a list of allowable commands. If the verb is not allowed, the message 'UNIDENTIFIED OPERATION FIELD' is issued and control passes to the Interpreter Get routine (IEFVHA).	IEFVHM	
2 The routine processes the command as follows:	IEFVHM	

Disposition	Action
0	Causes command to be executed.
1	Causes command to be executed, issues WTO displaying command.
2	Requests authorization for command execution via WTOR. Executes command if reply is positive.
3	Ignores command, returns control to Interpreter Get routine.

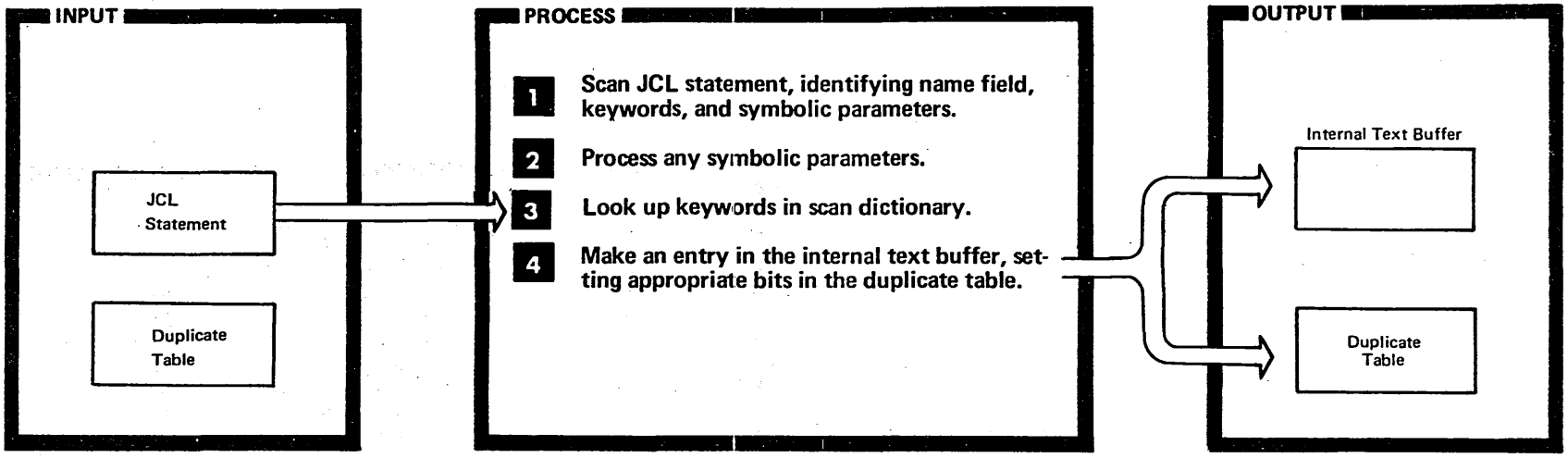
To execute the command, the Command routine places the command authority of the reader into register 0, makes the register negative to indicate that the command was in the jobstream, and issues an SVC 34. It then passes control to the Interpreter Get routine.

Figure 2-18. Prescanning a JOB Statement



Extended Description		Module	Figure
1	The Prescan routine uses the Queue Management Assign and Start routine to start the SWADS entry.	IEFVHEB	3-17 3-19
2	The reader assigns and initializes the JMR. The Prescan routine passes control to the User JCL Validation routine (IEFUJV), if one is supplied. If the return code from the Validation routine indicates that the job is to be cancelled, the Prescan routine sets the job-failed bit in the JCT. It then uses the Message Writing routine (IEFVGM) to copy the JOB statement into the system message data set. It then passes control to the Scan routine (IEFVFA) (Figure 2-19).	IEFVHEB	

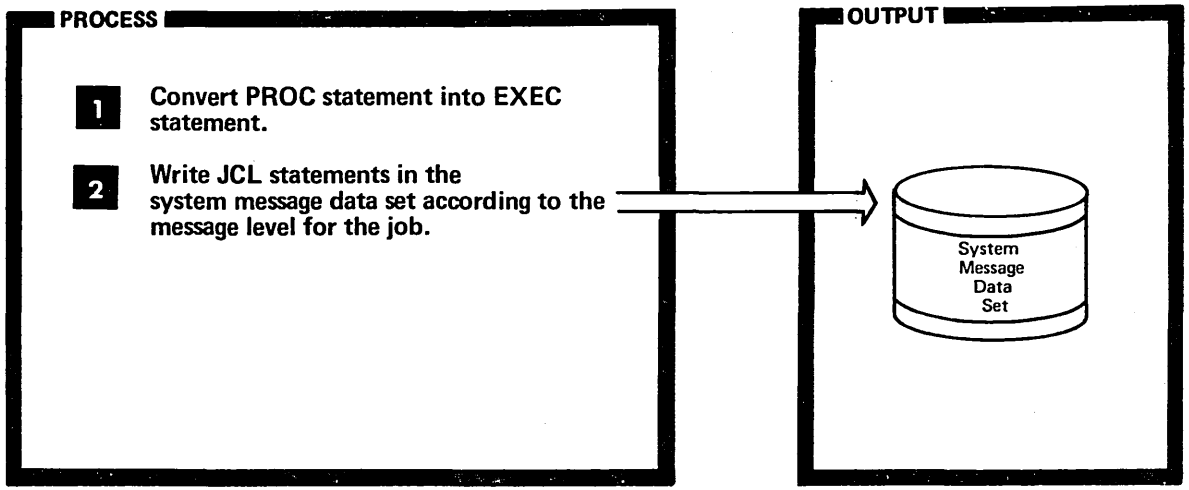
Figure 2-19. Scanning the JCL



Extended Description	Module	Figure
1 The Job Control Language Scan routine scans each JCL statement from left to right.	IEFVFA	
2 If any JCL statements contain symbolic parameters, the Symbolic Parameter routine enters the substituted JCL statement into the system message data set following the original input JCL statement.	IEFVFA	3-17 3-19
3 For each valid keyword, the scan dictionary entry contains the corresponding one-byte binary "key", and lists: • the keys of any mutually exclusive keywords, for example, DDNAME and SPACE. These are for error checking. • the keys of any minor keywords associated with the keyword that the entry represents, for example, SEP is listed as a minor keyword of the UNIT parameter. The minor keys are overridden when the corresponding major key is overridden.	IEFVFA	

Extended Description	Module	Figure
4 The Scan routine tests bits in a duplicate table to determine that a keyword was not encountered previously or a mutually exclusive keyword was not encountered. The duplicate table is a 32-byte table containing a bit for each key. The position of the bit in the table corresponds to the key. When the scan routine makes an entry in the internal list, it turns on the bit in the duplicate table that corresponds to the key it is processing, as well as the bits corresponding to any mutually exclusive keywords. If a bit in the duplicate table is on during a procedure merge, the field being processed was overridden and the Scan routine proceeds to the next field. In all other cases, the routine turns on the job-failed bit in the JCT and passes control to the Post-scan routine (IEFVHF).	IEFVFA IEFVFA	

Figure 2-20. Prescan EXEC and DD Statement Processing



Extended Description		Module	Figure
1	The Prescan Preparation routine converts the PROC statement into an EXEC statement.	IEFVHEB	3-16 3-17
2	If the JCL message level is one, the Message Writing routine (IEFVGM) places the other JCL statements and the cataloged procedure statement for the job into the system message data set.	IEFVGM	

Extended Description	Module	Figure
If the message level is two, only input JCL statements enter into the data set. The routine then passes control to the User JCL Validation routine (IEFUJV), if supplied.	IEFUJV	
If the return code form IEFUJV specifies job cancellation, the routine sets the job-failed bit in the JCT. Then, it passes control to the JCL Scan routine (IEFVFA) (Figure 2-19).	IEFVGM	

Figure 2-21. Processing Parameters (Part 1 of 2)

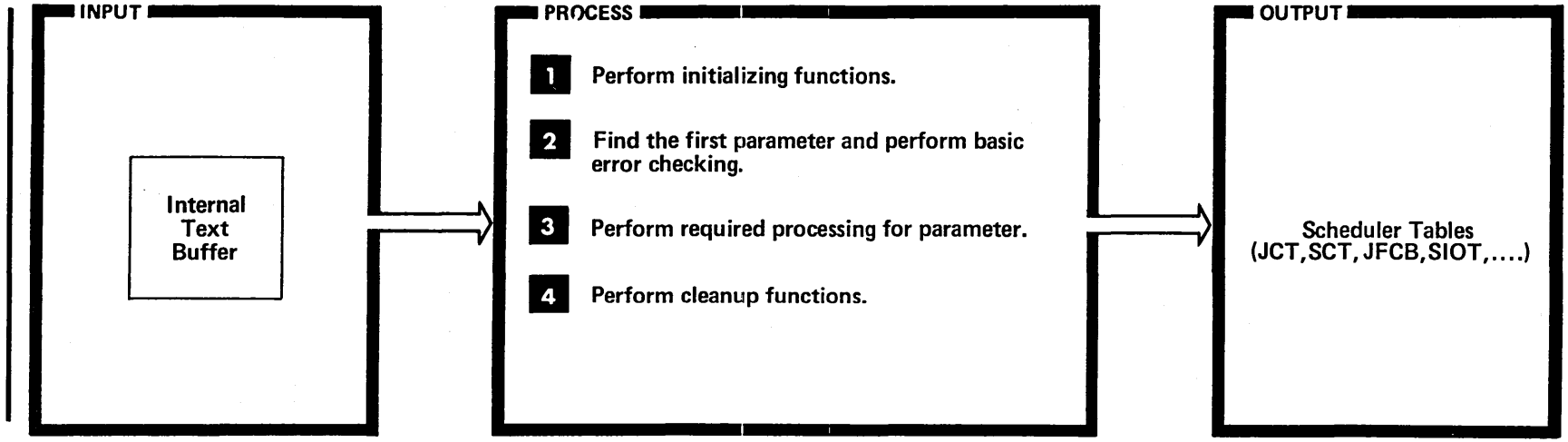


Figure 2-21. (Part 2 of 2).

Extended Description	Module	Figure
1 The Statement Processor routine (IEFVJA - JOB statement processor, IEFVEA - EXEC statement processor, or IEFVDA - DD statement processor) clears the storage area in which it constructs tables. An exception occurs in the case of IEFVEA where some overrides pass from step to step in the table and therefore are not cleared. The Local Work Area (LWA) is initialized and control passes to the Get Parameter routine (IEFVGK).	IEFVJA IEFVEA IEFVDA	3-15
2 The Get Parameter routine locates the next (or first) parameter and returns control to the appropriate keyword subroutine of the processor routine.	IEFVGK	
3 The keyword subroutine passes control to the test-and-store routine which processes the parameter in accordance with the Parameter Description Table (PDT). The PDT describes the processing required for a parameter; there is one entry for every keyword. The entries contain two types of information: length and error checking information and control information describing the processing to be performed. When processing is complete, control passes back to the keyword subroutine which may perform some additional processing and then passes control to the Get Parameter routine to get another parameter. When the last keyword has been processed or when the test-and-store routine or the Get Parameter routine finds an error, control passes to the cleanup routine. The information obtained from the parameters by the processors is stored in the following tables: IEFVJA – Job Control Table (JCT) Account Control Table (ACT) IEFVEA – Step Control Table (SCT) Account Control Table (ACT) IEFVDA – Step I/O Table (SIOT) Job File Control Block (JFCB) Volume Table (VOLT) Data Set Enqueue table (DSENQ) Data Set Name Table (DSNT) IEFVEA and IEFVDA link to IEFVGI and IEFVGS to create and retrieve information from the dictionaries used in resolving JCL backward references.	IEFVJA IEFVEA IEFVDA IEFVGT IEFVGI IEFVGS	
4 The cleanup routine uses the Message routine to write any error messages. IEFVJA's cleanup routine writes out the ACT. IEFVEA's cleanup routine resolves overrides from the JOB and/or EXEC PROC statements. IEFVDA's cleanup routine uses IEFVDBSD to construct the DSENQ table. In the case of a SYSOUT DD statement, IEFVSD13 is linked to in order to create a Data Set Block (DSB) for the data set. An ODS and an ENDS are issued against this data set to checkpoint it. The SIOT and the JFCB are written out to the queue (or SWADS) for every DD statement. When a cleanup routine has completed its processing, it passes control to the Interpreter [Control routine at the Post-scan routine (IEFVHF).	IEFVJA IEFVEA IEFVDA IEFVDBSD IEFVSD13	

Figure 2-22. In-Stream Procedure Processing (Part 1 of 2)

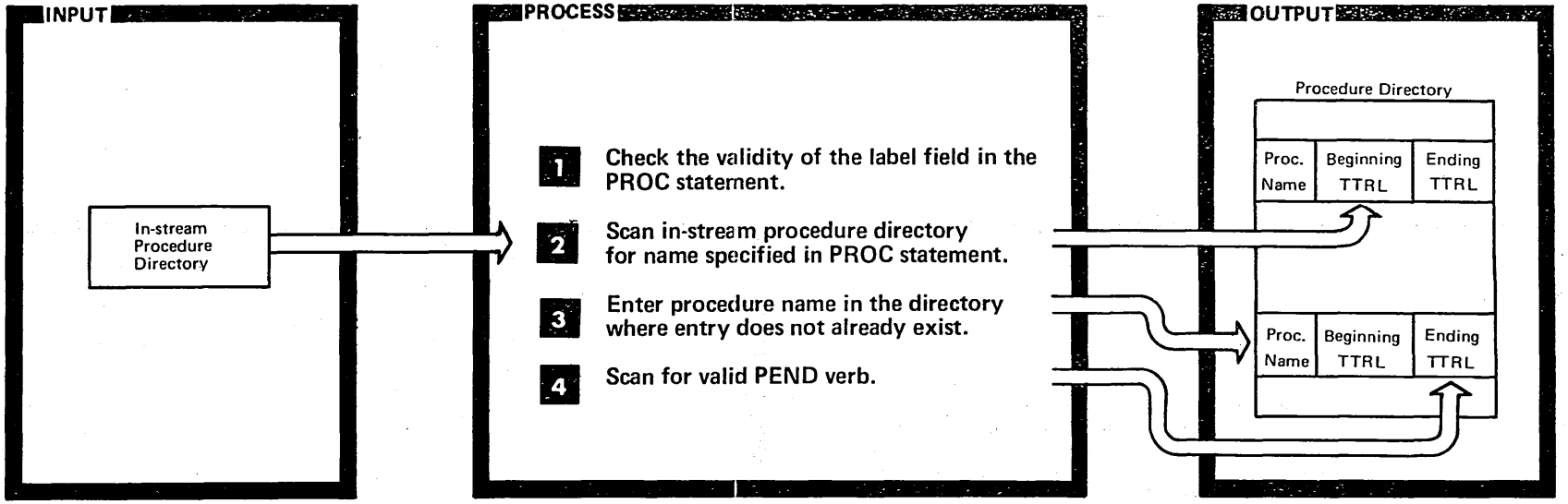
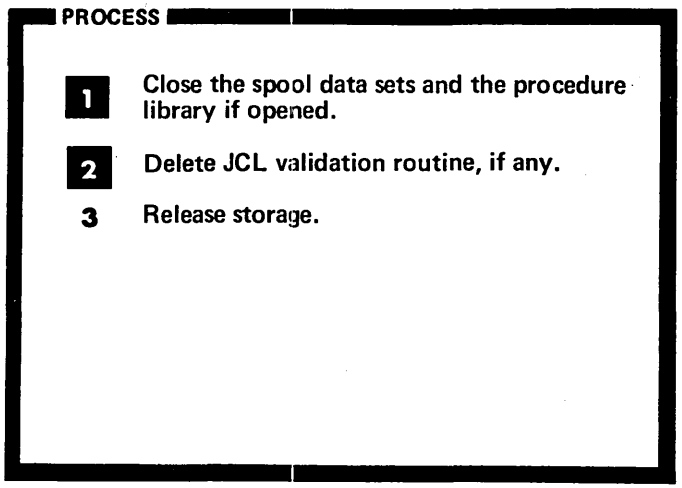


Figure 2-22. (Part 2 of 2)

Extended Description	Module	Figure
1 The In-stream Procedure Syntax Check routine (IEFVINE) receives control from the In-stream Procedure Router (IEFVINA) to perform the syntax check, then passes a return code back to the Router. The Router processing depends upon the code.	IEFVINE	3-20
2 If the procedure name is found, the Directory Search routine (IEFVINB) updates the address in the directory of the first record containing the procedure. If the procedure is not found, the routine sets a return code of zero in the parameter list and passes control to the in-stream router (IEFVINA) which gives control to the In-stream Procedure Directory Build routine (IEFVINC).	IEFVINB	3-20
3 If the directory already contains fifteen entries, control returns to the in-stream procedure router to create an error message and set the job-failed bit in the Job Control Table (JCT). This routine passes control to the Interpreter Get routine to read the next statement.	IEFVINA	3-20
4 The in-stream procedure router scans each statement for a valid PEND verb. When one is encountered, it places the ending TTRL in the directory, frees the parameter list, and passes control to the Interpreter Get routine (IEFVHA).	IEFVINA	3-20

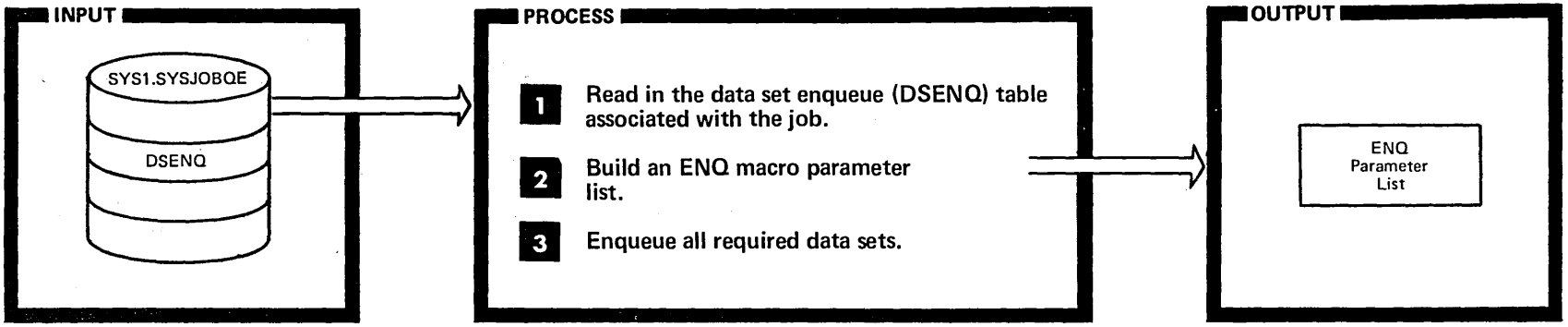
Code	Meaning	Action
0	Procedure name omitted from PROC statement.	Set job-failed bit in job control table and issue the appropriate error message. Pass control to Interpreter Get routine to process next statement.
12	Invalid label.	
1-8	Length of procedure name.	Determine if the procedure being processed is the first in-stream procedure. If it is, control passes to the In-stream Procedure Directory Build routine (IEFVINC). Otherwise, build a parameter list for the procedure and pass control to the In-stream Procedure Directory Search routine (IEFVINB).

Figure 2-23. Interpretation Exit Routine



Extended Description		Module	Figure
1	If the job is being cancelled or interpretation is complete, control passes to the Interpreter Termination routine.	IEFVHN	
2	The Termination routine releases all areas obtained by the interpreter, and passes control to the initiator.	IEFVHN	

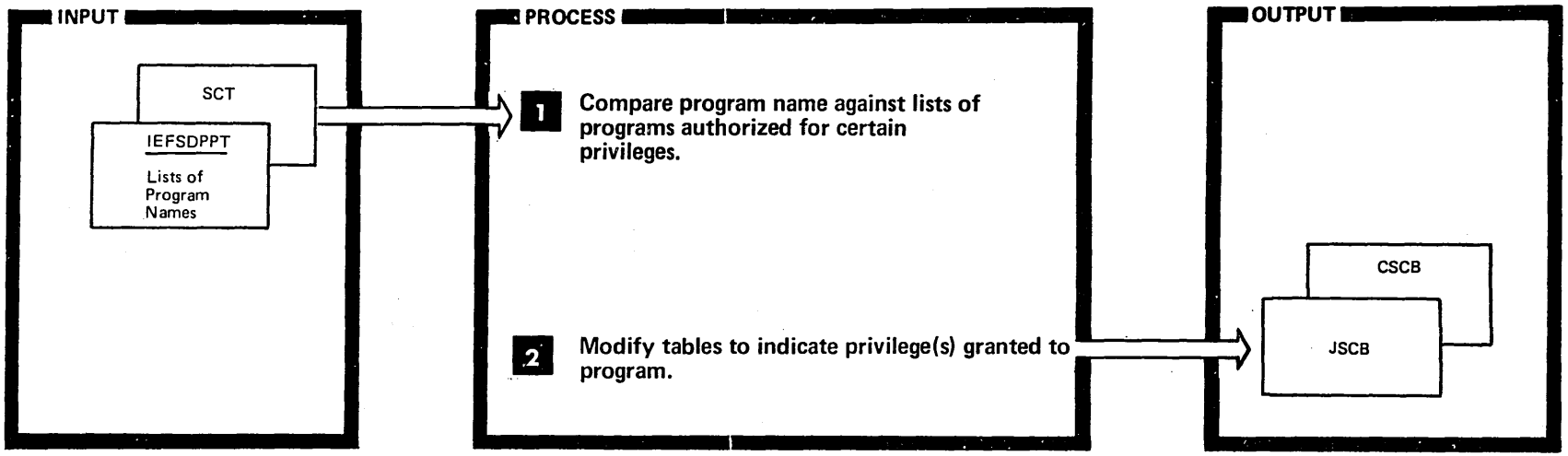
Figure 2-24. Enqueuing on Data Sets



Extended Description		Module	Figure
1	The DSENQ table contains the names of the non-temporary data sets required by the job, and indicates whether the data set can be shared or whether exclusive use of it is required.	IEFSD161	
2	The ENQ macro parameter list contains each name in the DSENQ table, as well as its attribute. Duplicate names in the DSENQ table are eliminated in the ENQ macro parameter list, and, where the attributes of the duplicates differ, the more restrictive attribute is assigned. This parameter list is written to the Scheduler Work Area Data Set (SWADS) for a problem program or the queue data set for a system task.	IEFSD102	

Extended Description		Module	Figure
3	If the data sets are unavailable, a message is issued with a reply of RETRY, HOLD, or CANCEL requested for a problem program, and RETRY or CANCEL for a system task.	IEFSD102	

Figure 2-25. Program Authorization



Extended Description	Module	Figure
1 The Program Authorization Check routine gets the program name from IEFSDPPT of each step as it is scheduled. It compares the name against three lists of programs eligible for certain privileges in execution.	IEFSD101	3-8
2 If a program is found on any or all of the lists, the Authorization Check routine sets bits in the appropriate tables to indicate the authorization.	IEFSD101	3-8

Extended Description	Module	Figure

Privilege	Table-Field	Bit setting
Non-cancel-lable status	CSCB-CHACT	bit 4-off
Long exe-cution time	JSCB-JSCBAUTH	bit 2-on
Issue MODESET, EXCPVR, or FIX/FREE PAGES	JSCB-JSCBAUTH	bit 7-on

Figure 2-26. Initiating a Job Step

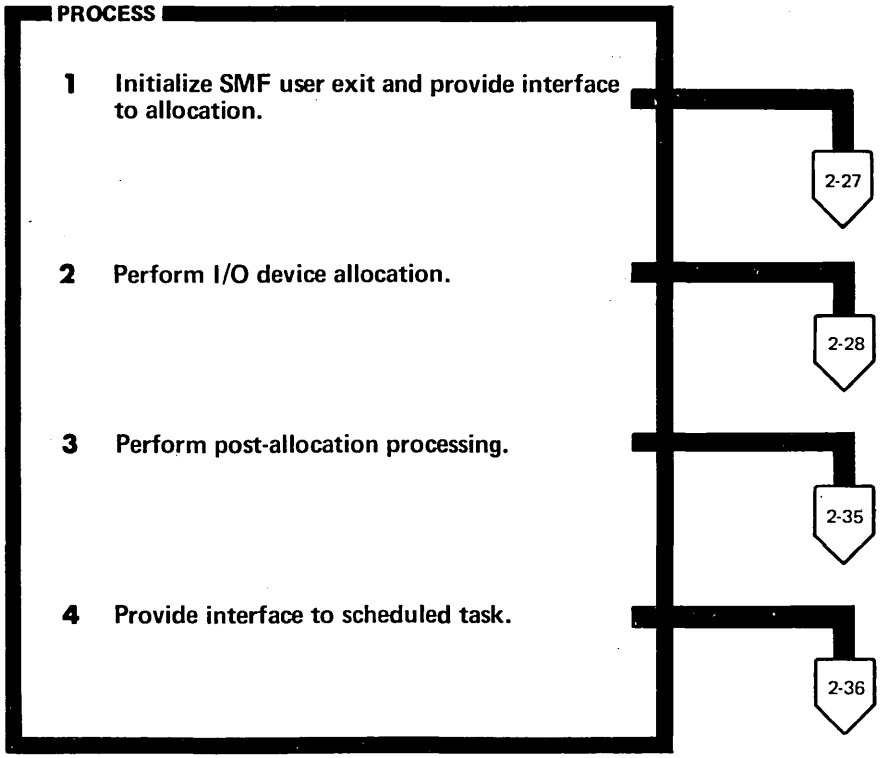
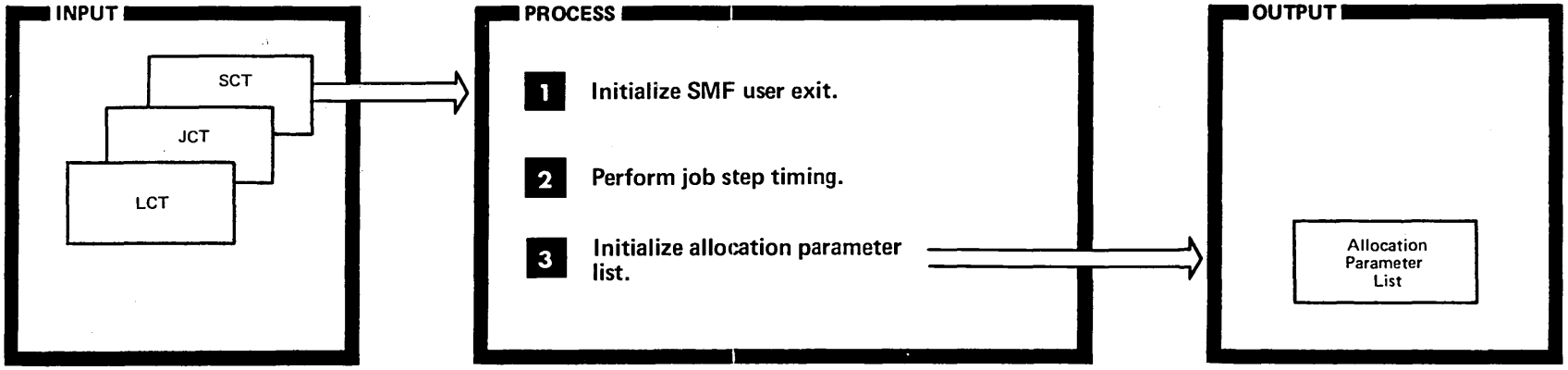


Figure 2-27. Initializing the SMF Exit



Extended Description	Module	Figure
1 If SMF options are specified for the system, the Step Initiation routine (IEFSD162) calls the SMF Initiatization routine (IEFSMFIE) to build the TCT and in-core JMR used by the user's SMF exits.	IEFSD162 IEFSMFIE	3-4
2 The Step Initiation routine performs job-step timing according to TIME parameters on the JOB or EXEC statement.	IEFSD162	

Extended Description	Module	Figure
The TQE pointers of the TCB and PIB are swapped before and after the STIMER.	IEFSD263	
3 The Step Initiation routine builds the parameter list and passes control to the Allocation Entry routine (IEFSD21Q).	IEFSD162	

Figure 2-28. I/O Device Allocation Overview

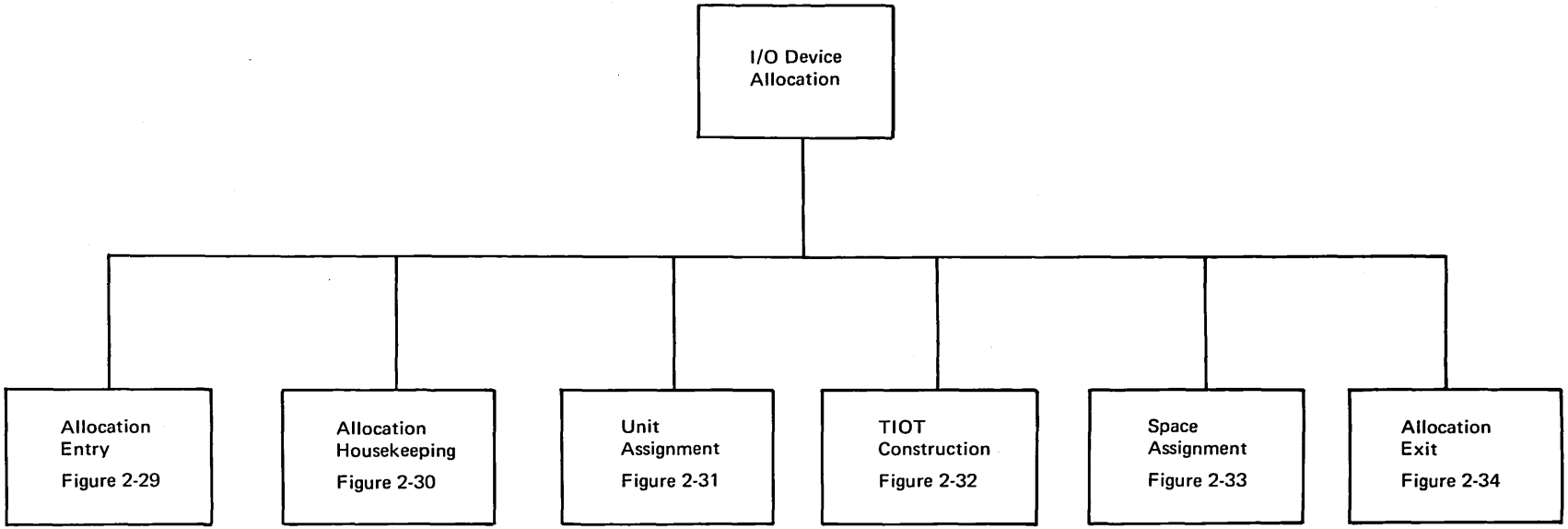


Figure 2-29. Checking the EXEC Statement Condition Codes (Part 1 of 2)

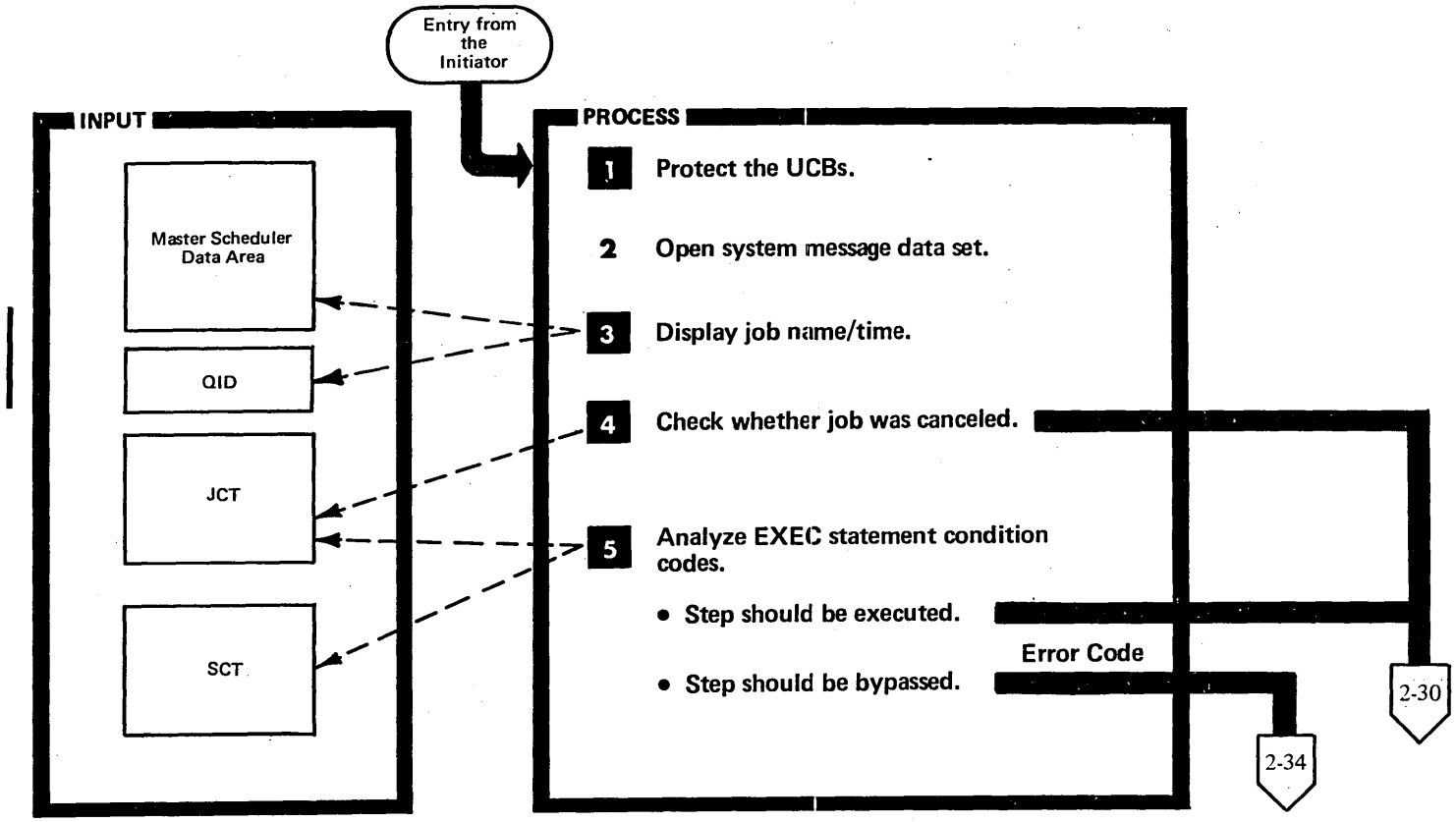


Figure 2-29. (Part 2 of 2).

Extended Description	Module	Figure
1 An ENQ macro instruction is issued to prevent access by other tasks to the UCBs. This ensures that the termination or allocation of another task cannot take place involving unit and volumes required by allocation for this task. The ENQ is in major resource name SYSIEFSD and minor name Q4.	IEFSD21Q	3-23
3 If this is the first step of the job, the job and time notification bits in the master scheduler data area and the QID for RES are tested. If they are on, the job started message is issued.	IEFSD21Q	3-23
4 If the job-failed or job-flush bits are on (JCT), control passes to the JFCB Housekeeping Control routine. The remaining steps are processed, but not executed. The job-failed bit is set off, and the job-flush bit is set on.	IEFSD21Q	3-23

Extended Description	Module	Figure
5 If this is not the first step of a job, the step is bypassed if: • One or more previous steps were abnormally terminated, and the COND field specifies neither EVEN nor ONLY. • No steps were abnormally terminated, and the COND field specifies ONLY. • Comparison of the return code of the step specified in the first COND field to the condition code specified for the current step matches the condition code operator.	IEFVKIMP	3-23

Figure 2-30. Gathering Information about Requests for I/O Devices and Data Sets (Part 1 of 2)

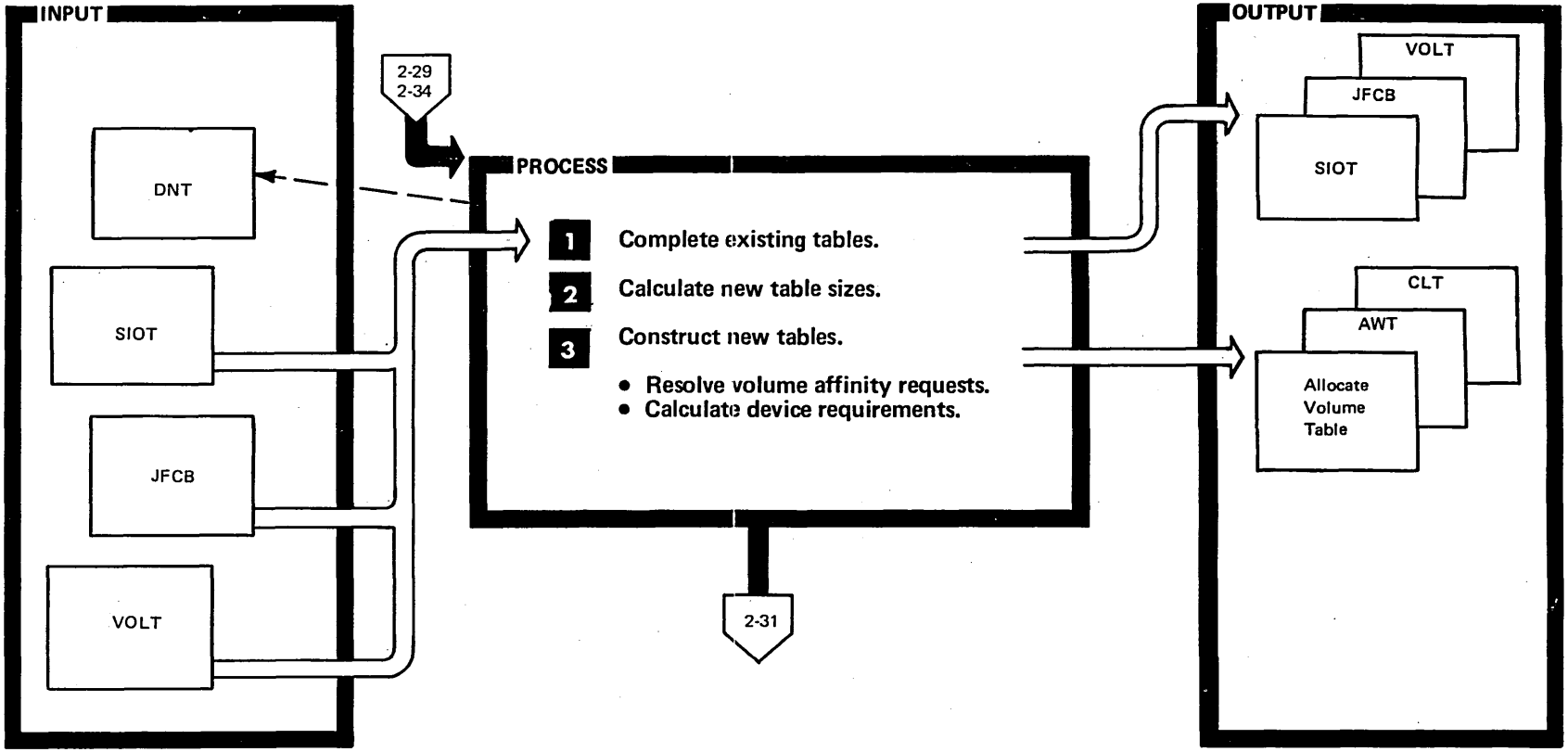


Figure 2-30. (Part 2 of 2).

Extended Description	Module	Figure
1 The JFCB Housekeeping routines complete volume information in the JFCB, SIOT, and VOLT. Note: If the job-flush bit or step-flush bit is on, the step is processed in flush mode. All data sets are processed as dummy data sets, and no volume information is needed. This is called flush-mode processing.	IEFVMLS1	3-23
2 The Allocation Control routine determines the sizes of certain tables constructed by I/O Device Allocation, obtains the necessary virtual storage, and puts the address of the storage reserved for each table into the Allocation Control Block. It also checks for and executes any VARY OFFLINE or UNLOAD commands.	IEFXCSSS	3-23

Extended Description	Module	Figure
3 The Demand Allocation routine builds the Allocate Work Table (AWT), and links data sets with similar device requirements (SPLIT and SUBALLOC), and builds the Allocate Volume Table. It also resolves volume affinity requests, linking entries in the Allocate Volume Table. It calculates the device requirements of each data set, and constructs the Channel Load Table (CLT).	IEFWA000	3-23

Figure 2-31. Allocating Devices to a Job Step or System Task (Part 1 of 2)

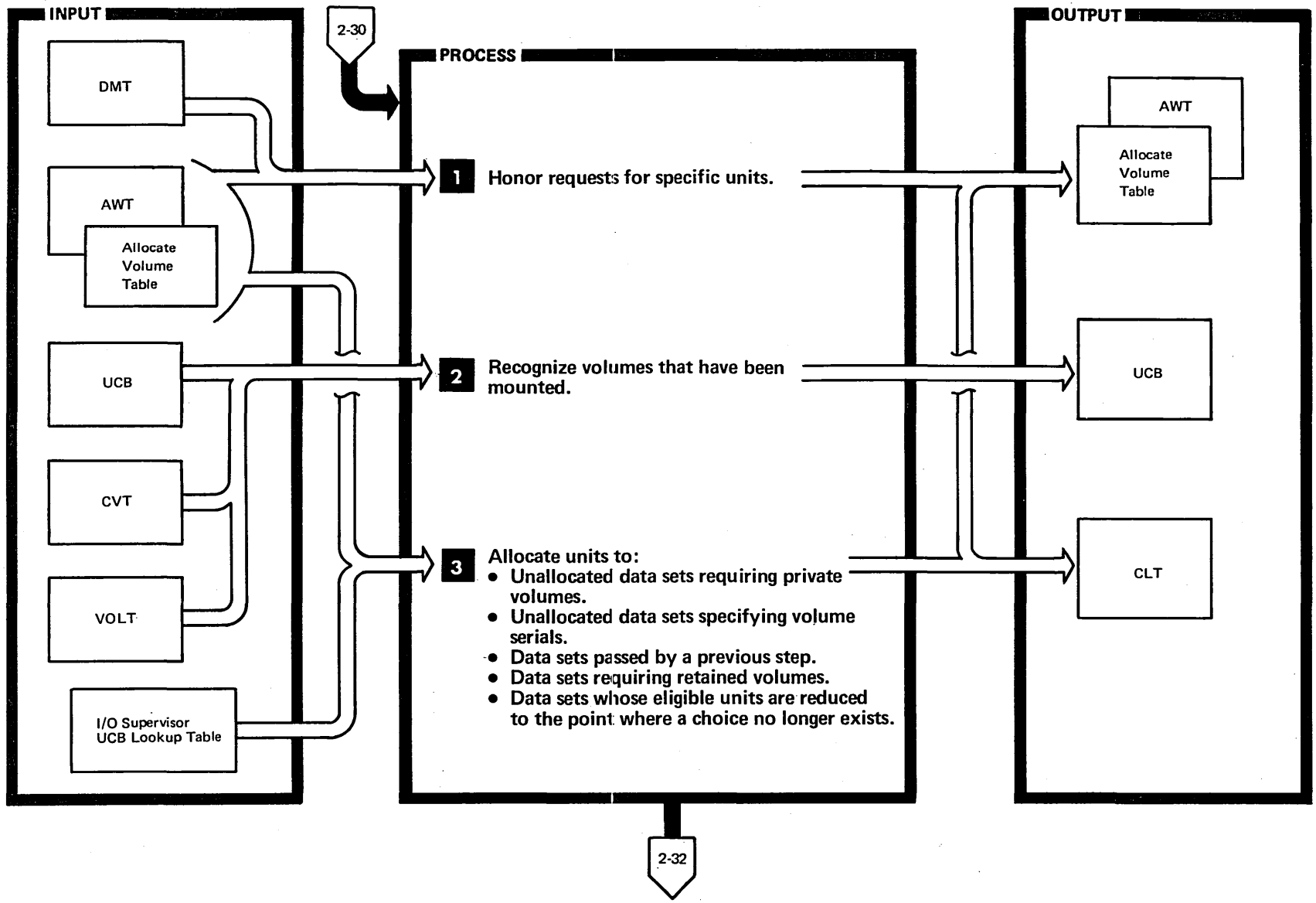


Figure 2-31. (Part 2 of 2).

Extended Description	Module	Figure
1 The Demand Allocation routine honors explicit requests for resident direct access volumes and reserved units; it reduces the range of eligible devices; and it honors explicit and implied requests for specific units.	IEFWA000	3-23
2 If Automatic Volume Recognition (AVR) is specified when the system is generated, the operator can mount volumes required for subsequent job steps as soon as units become available. The AVR routine recognizes volumes that have been mounted for the step being initiated, thus saving the time that the system would otherwise spend waiting for the operator to find and mount them.	IEFXV001	3-23

Extended Description	Module	Figure
3 The Decision Allocation routine processes channel and unit separation requests. The routine then processes all remaining specific volume requests which have not been satisfied by the Demand Allocation routine or the AVR routine.	IEFX5000	3-23

Figure 2-32. Creating the TIOT (Part 1 of 2)

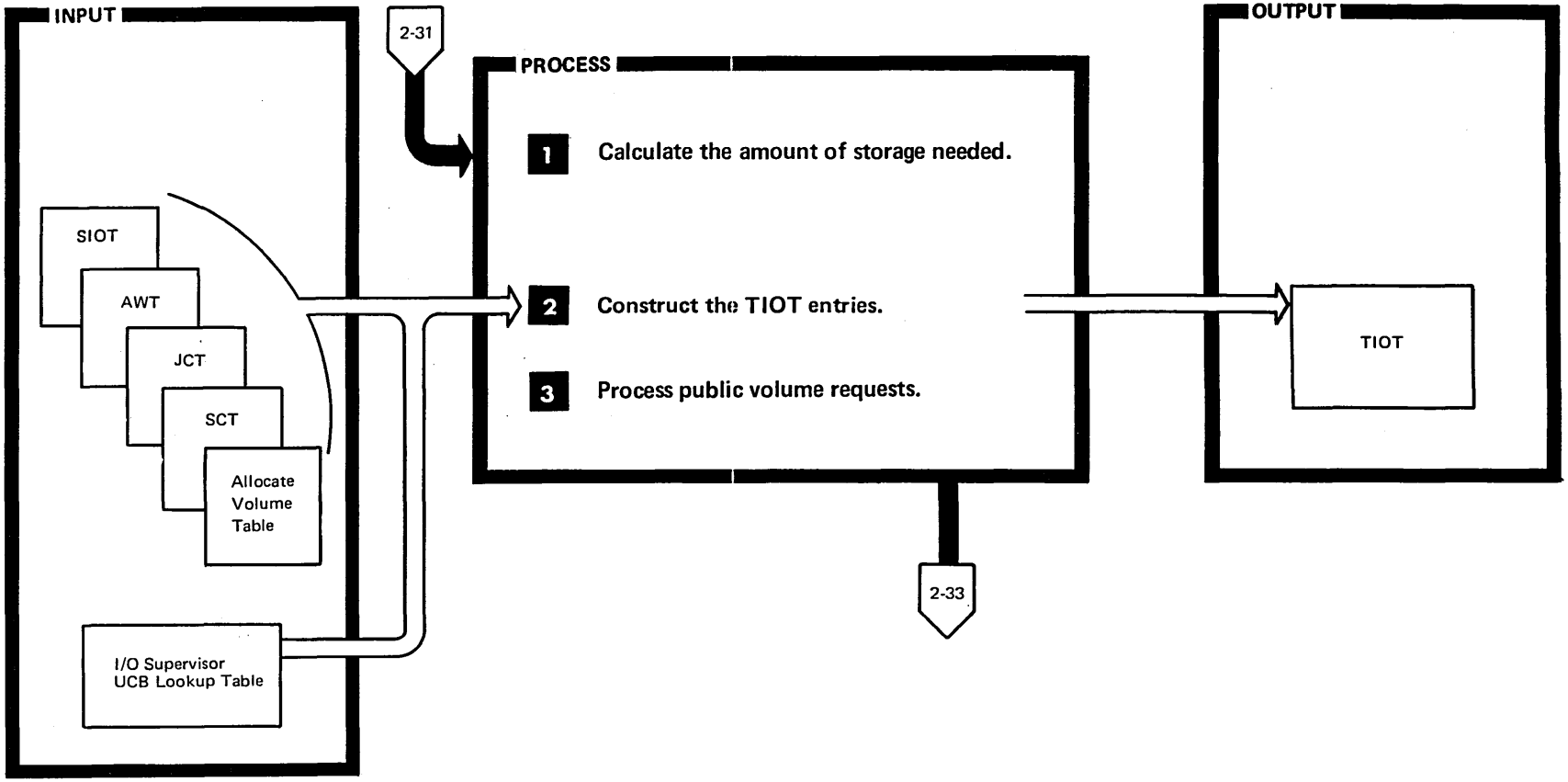


Figure 2-32. (Part 2 of 2)

Extended Description	Module	Figure
1 Before the module builds the TIOT, it must calculate the amount of storage required. The number of bytes needed is the sum of: • 40 - for the TIOT header and TIOT list. • 16 times the number of DD statements in the step. • 4 times the number of devices allocated to the step. • 4 times the product of the number of non-specific requests for space on public volumes times the number of devices available for public volumes.	IEFWCIMP	3-23

Extended Description	Module	Figure
2 The module constructs a TIOT entry for each data set associated with the step. As the entries are built, any logical links between data sets are recorded.	IEFWCIMP	3-23
3 When it creates the TIOT entry for a data set requiring space on a public volume, the TIOT Construction routine builds in the entry, a list of eligible units in descending order of suitability. If I/O Load Balancing is taken as a system generation option, the candidate list is not in descending order.	IEFWCIMP	3-23

Figure 2-33. Issuing Messages for Mounting Volumes and Allocating DASD Space (Part 1 of 2)

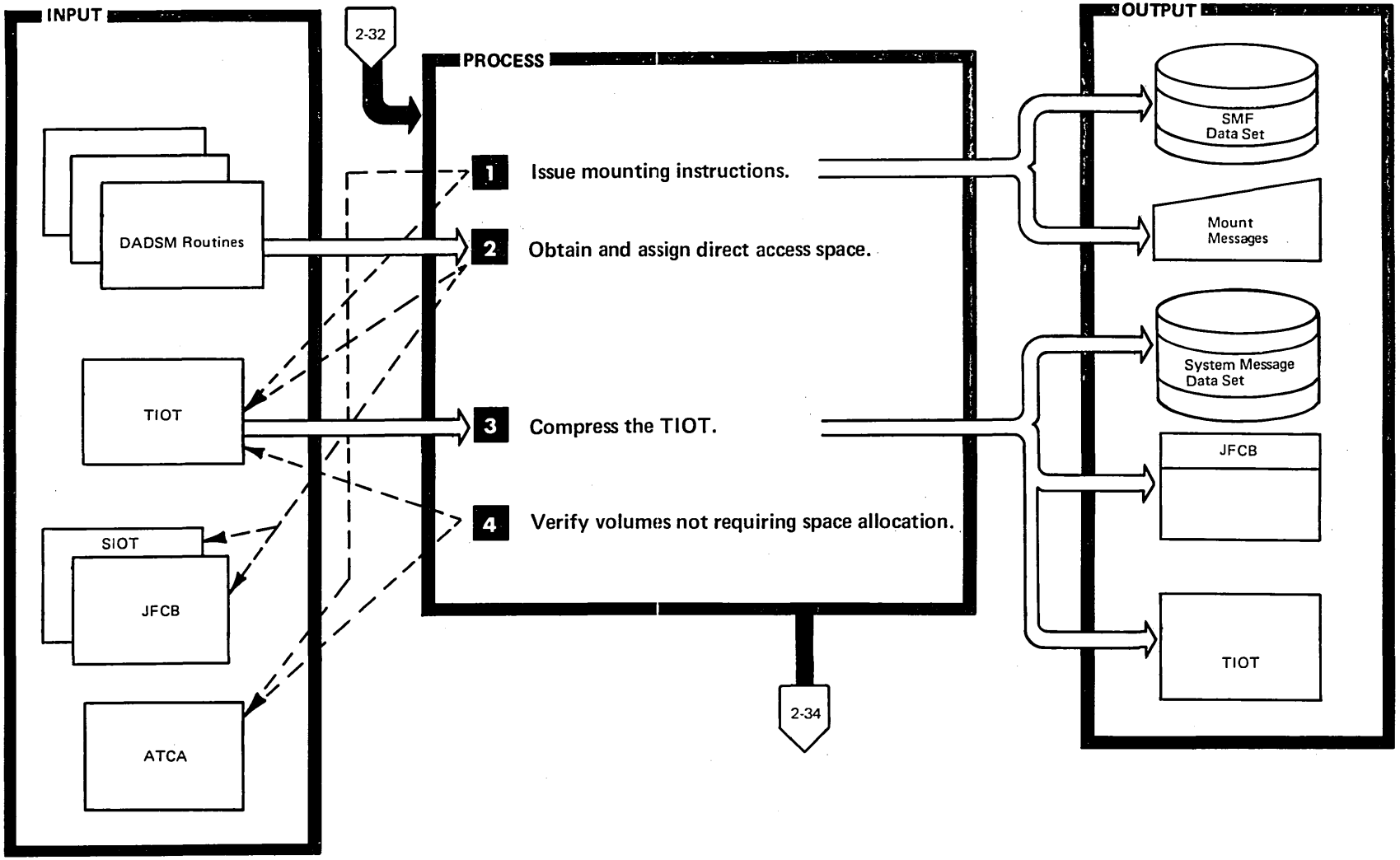
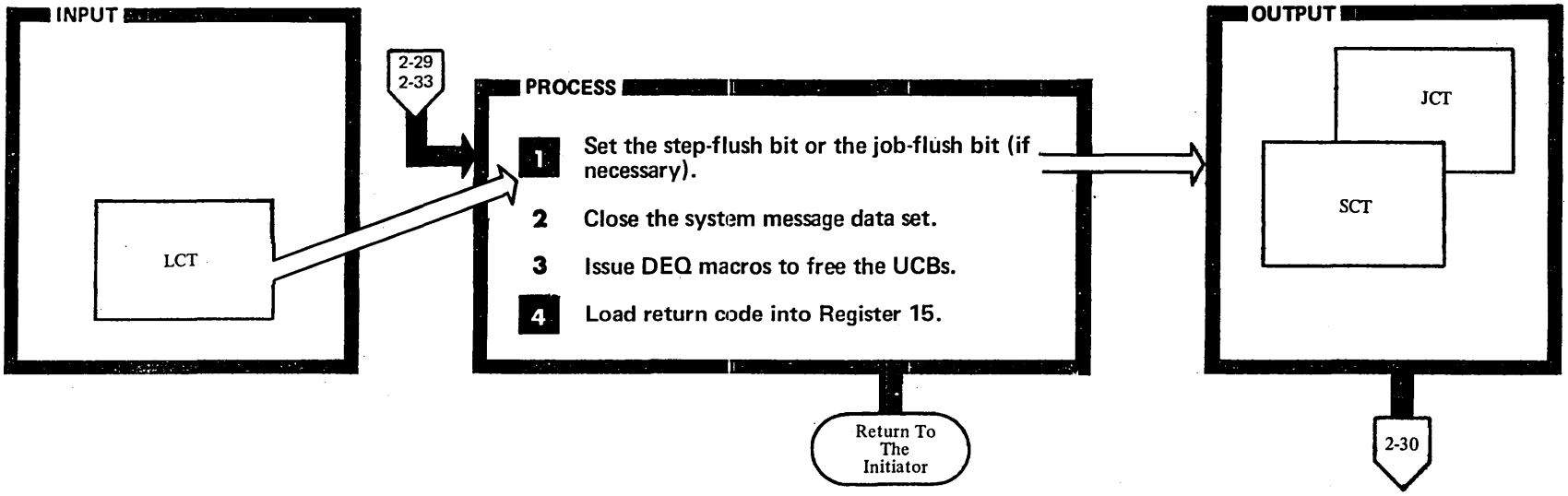


Figure 2-33. (Part 2 of 2)

Extended Description	Module	Figure
1 The External Action routine is entered from the TIOT Construction routine to issue mounting instructions, to verify that volumes requiring space allocation have been mounted, and to unload any devices containing the wrong volumes.	IEFWD000	3-23
2 The Space Request routine requests space from DADSM and completes the allocation of units to data sets requiring public volume space. For public volume requests, if I/O Load Balancing is in the system, the Space Request routine must first choose a unit from the list of eligible units built by TIOT Construction routines. I/O Load Balancing routines are invoked to choose the unit which will add least to device contention.	IEFXT00D IEFAB400 IEFAB401	3-23
3 The TIOT is compressed by eliminating unused units from the list in TIOT entries that requested space on public volumes. The amount of storage required for the compressed TIOT is calculated, and the excess storage is released. This routine also provides an interface with the VARY command when a device is changed from an online status to a console status. The number of bytes required for the compressed TIOT is the sum of: • 40 • 16 times the number of DD statements in the step. • 4 times the number of devices allocated to the step.	IEFXT002	3-23
4 Volumes that do not require space allocation (volumes containing old data sets) are verified.	IEFWEXTA	3-23

Figure 2-34. Allocation Exit



Extended Description	Module	Figure
1. The Allocation Exit routine, when entered because of an error condition, sets the step-flush bit (if the error was a condition code error) or job-flush bit on to indicate flush-mode processing. If the TIOT has not been built, it passes control to the JFCB House-keeping routine.	IEFSD41Q	3-23

Extended Description	Module	Figure
4 The Allocation Exit routine determines whether allocation was successful or unsuccessful.	IEFSD41Q	3-23

Code	Meaning
0	Successful allocation
4	Unsuccessful allocation

Figure 2-35. Writing Job Separators (Part 1 of 2)

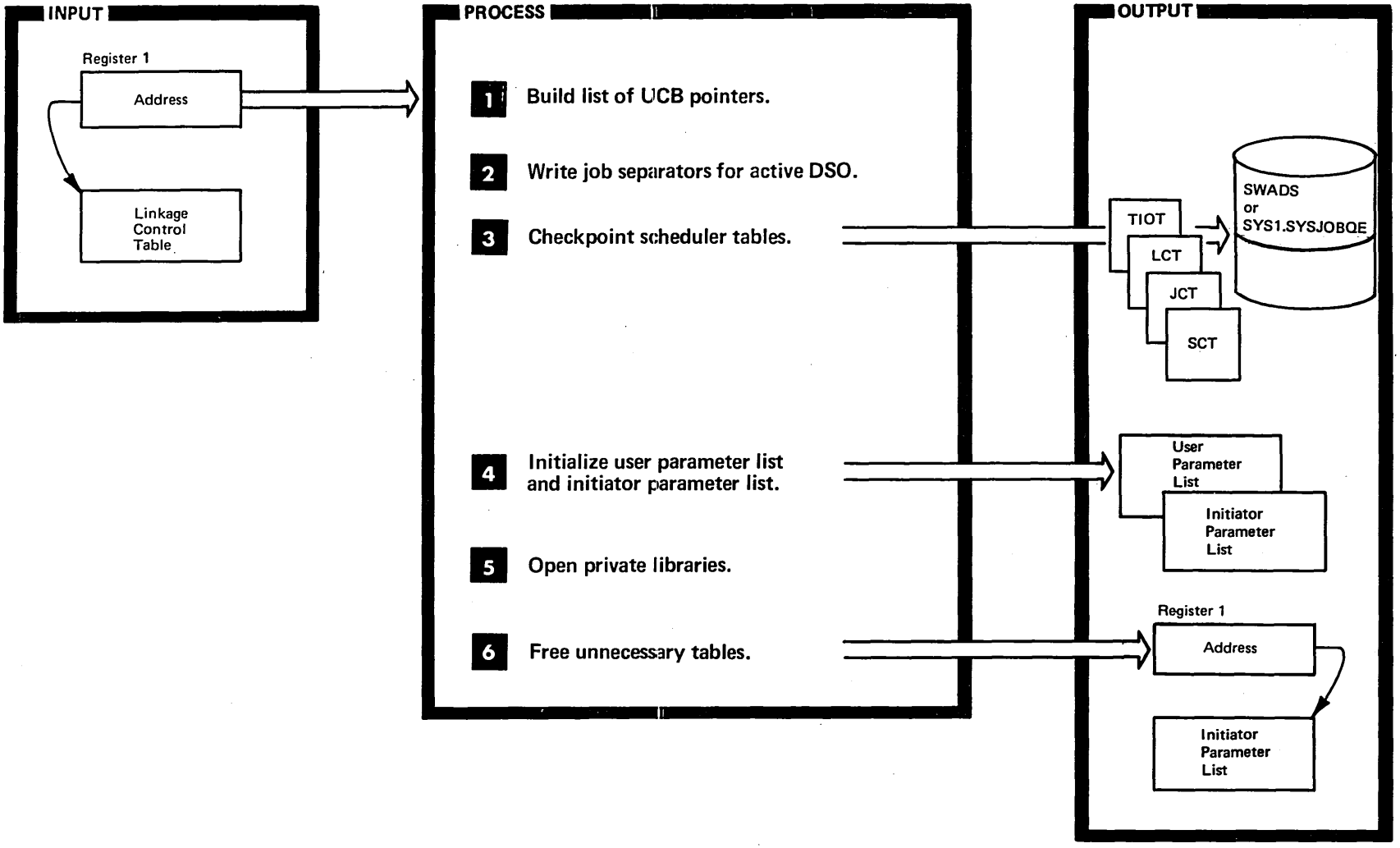


Figure 2-35. (Part 2 of 2).

Extended Description	Module	Figure
1 Allocation returns control to the Step Initiation/Allocation Interface routine. If allocation was successful, a list of UCB pointers is built so that if the scheduler abends, devices allocated can be deallocated by IEFSD515. If no scheduler abend occurs, the list is freed by IEFSD263 for system tasks, and by Termination for dequeued and started problem programs.	IEFSD162	3-12
2 The routine checks for Direct System Output (DSO). If DSO is active, IEFSD162 passes control to the DSO writer to write job separators, for the first step only.	IEFSD162	3-12
3 If allocation was successful, the routine writes the Task Input/Output Table (TIOT), Linkage Control Table (LCT), exit list, Job Control Table (JCT), and Step Control Table (SCT) to the job queue for system tasks and started problem programs. It writes the TIOT, LCT, JCT, and SCT to the Scheduler Work Area Data Set (SWADS) for dequeued problem programs. If allocation could not be done, the routine exits to the Alternate Step Delete routine (IEFSD065) to terminate the step.	IEFSD162	3-12

Extended Description	Module	Figure
4 The routine initializes the user parameter list and the initiator parameter list passed to the Problem Program/System Task Interface routine IEFSD263.	IEFSD162	3-12
5 The routine opens JOBLIB, STEPLIB, or FETCHLIB if required by the problem program. If the OPEN is not successful, a dummy program name (IEFSD0xx) is placed in the list passed to IEFSD263, and the message IEF317I is issued via the WTO macro. The problem program is terminated as a program not found. A STAE is active during OPEN processing so that an abend will cause the problem program, rather than the scheduler, to terminate.	IEFSD162	3-12
6 The routine frees all tables not required by IEFSD263, places a pointer to the initiator parameter list in register 1, and exits to IEFSD263.	IEFSD162	

Figure 2-36. Interfacing for a Task (Part 1 of 2)

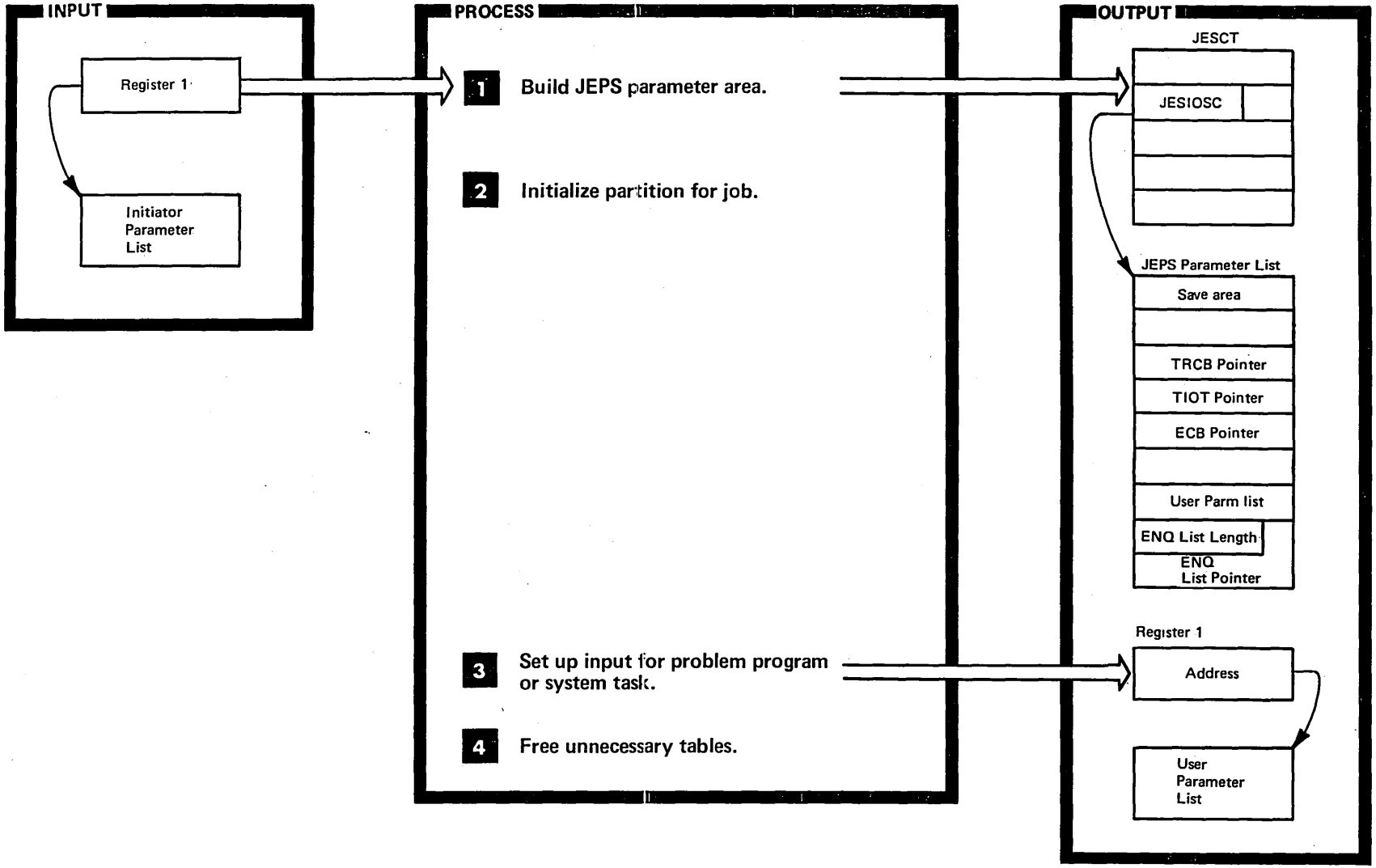


Figure 2-36. (Part 2 of 2).

Extended Description	Module	Figure
1 When a reader or writer is being started, the System Task/Problem Program Interface routine acquires the JEPS parameter area. A parameter list containing pointers and tables moves to the JEPS parameter area, and JEPS is posted for the START. When JEPS completes processing, it posts IEFSD263, which cleans up the tables and passes control to the partition controller.	IEFSD263	3-3
2 The Problem Program Interface routine passes control to the SMF Timing Control Task Input/Output Table (TCTIOT) Build routine (IEFSMFAT). If SMF exists in the system, this routine builds a TCTIOT and passes a pointer to the Timing Control Table (TCT) back to the Interface routine, which stores a job wait limit and a flag indicating whether the time limit is for a job or job step in the TCT. The Problem Program Interface routine then handles a virtual=real request if specified in the JCL. If the GETMAIN is unsuccessful, a message is issued via a WTOR macro specifying RETRY or CANCEL. The routine issues the STIMER macro to initiate job and job step timing if it is not overridden in the JCL.	IEFSMFAT IEFSD263	

Extended Description	Module	Figure
3 The Interface routine builds the user parameter list containing values specified in the PARM field of the user's JOB or EXEC statements. The routine also moves the Task Input/Output Table (TIOT) built by allocation for each job step into the upper end of the partition.	IEFSD263	
4 The Interface routine frees the Life-Of-Task (LOT) block, input/output blocks (except for IEFDSDRP), and, for a started reader or writer, the register save area, TIOT, and JEPS parameter area. The routine passes control via an XCTL macro to the problem program or system task.	IEFSD263	3-3

Figure 2-37. Terminating a Job and Job Steps

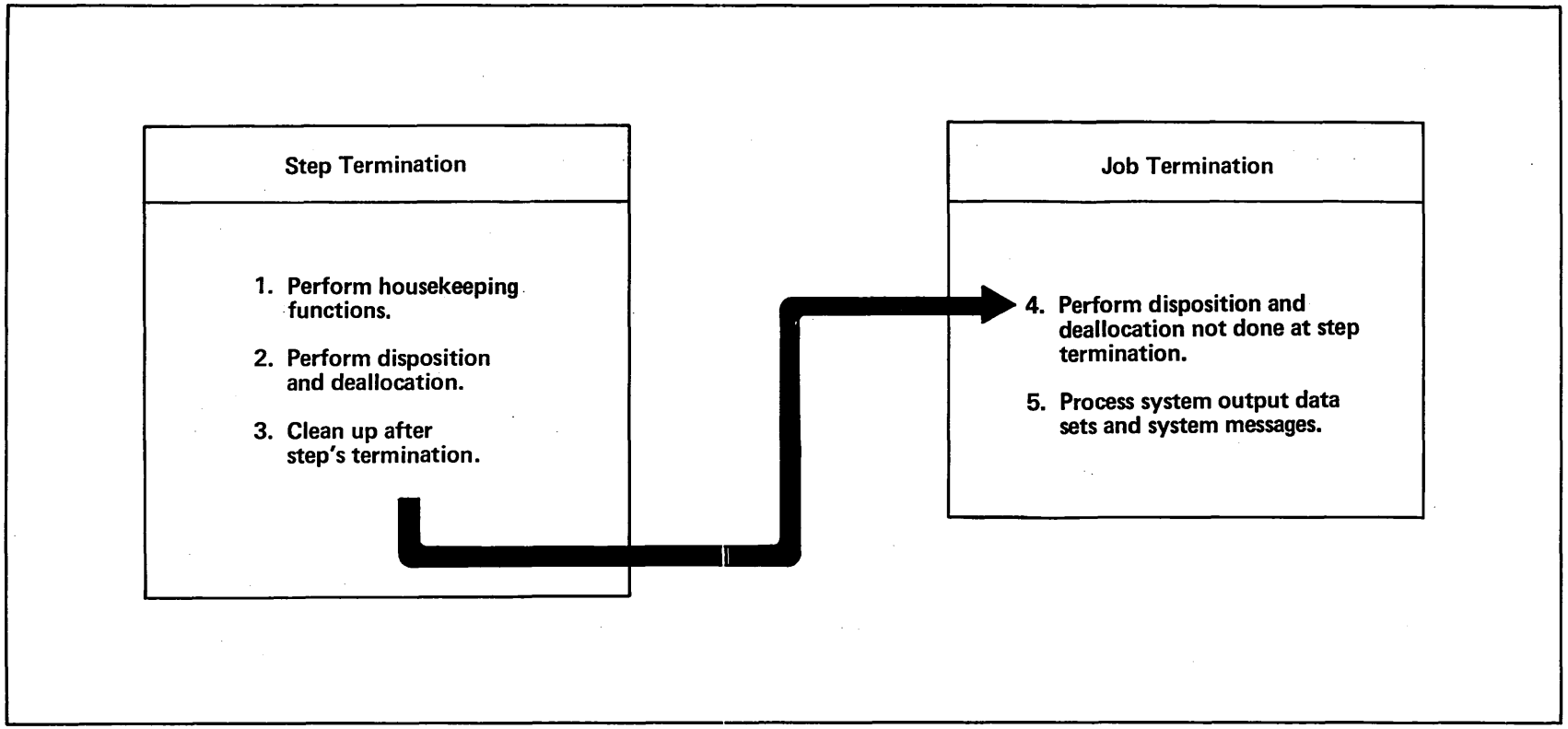


Figure 2-38. Preparing for Job and Job Step Termination (Part 1 of 2)

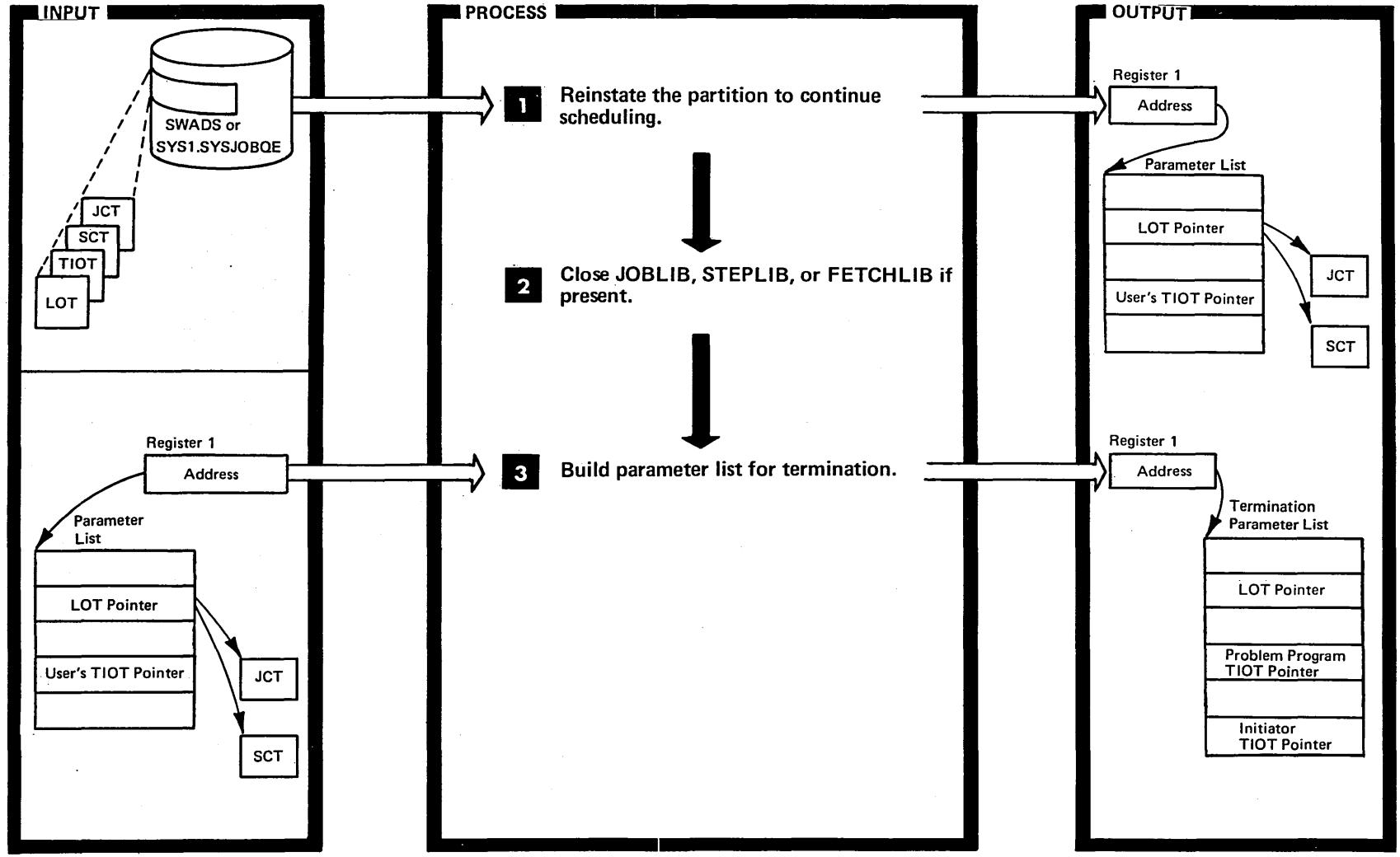


Figure 2-38. (Part 2 of 2).

Extended Description	Module	Figure
1 The Partition Step Deletion routine reads the job-related tables for a problem program into storage from the Scheduler Work Area Data Set (SWADS). It reads the job-related tables for a system task from the job queue. The routine builds a parameter list to pass pointers to these tables to the Termination Interface routine.	IEFSD515	
2 The Termination Interface routine closes the libraries.	IEFSD164	

Extended Description	Module	Figure
3 The Termination Interface routine builds a parameter list for Termination, using the parameter list passed to it from the Step Deletion routine. It builds the list for both job and step termination. It updates the job control table (JCT), writes it back to SWADS, and reads in the SCT for the next step, if there is one.	IEFSD164	

Figure 2-39. Normal Step Termination (Part 1 of 2)

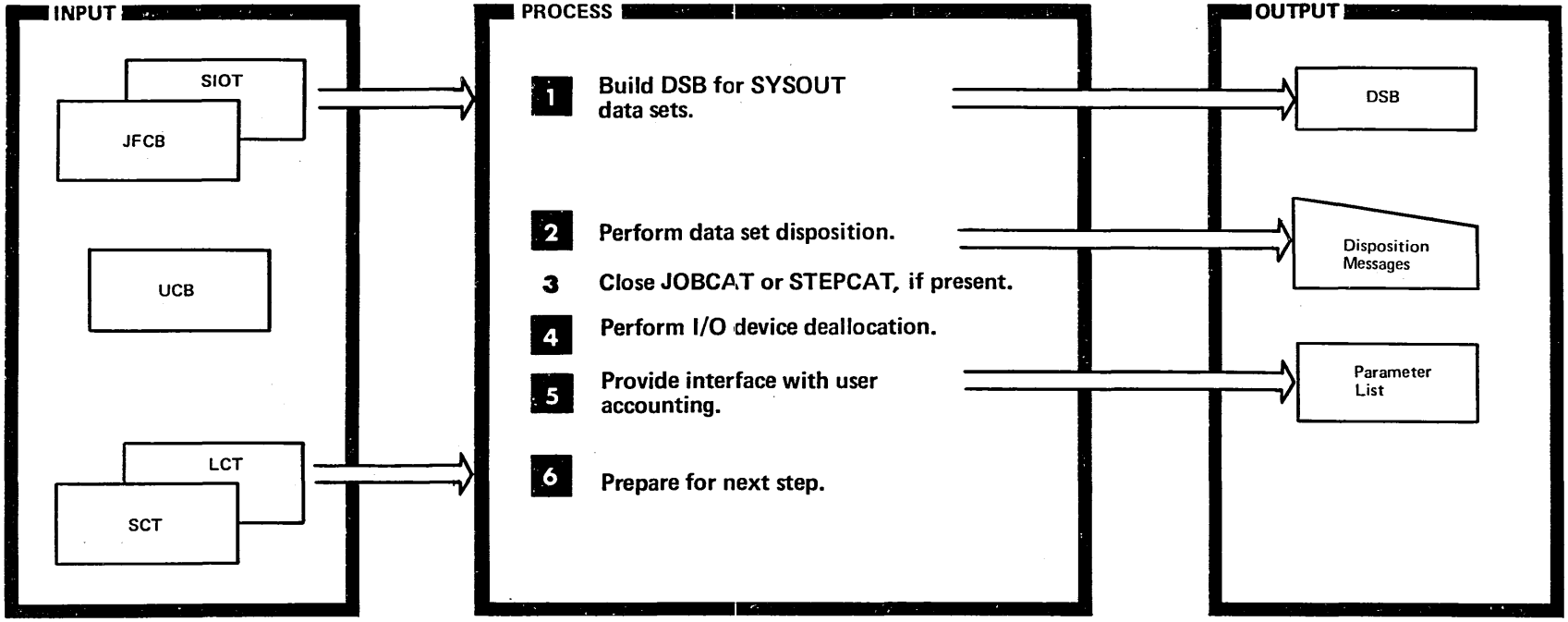
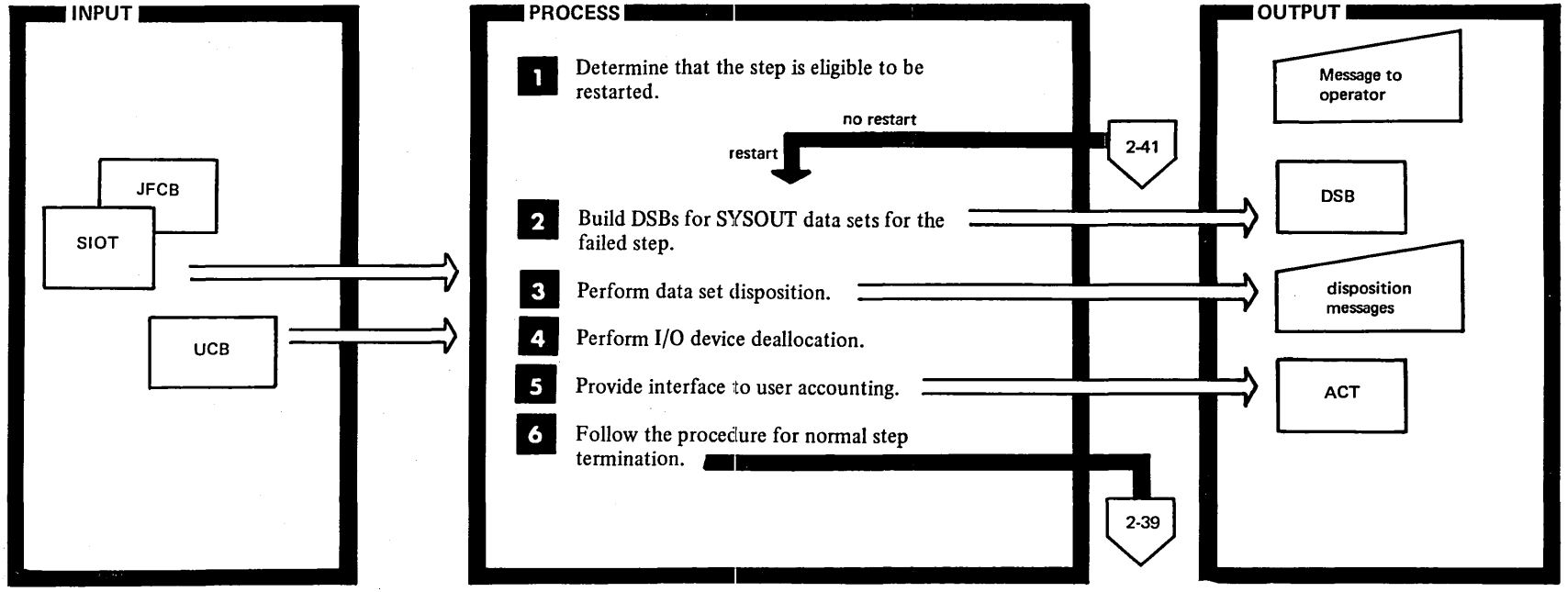


Figure 2-39. (Part 2 of 2)

Extended Description	Module	Figure
1 The Data Set Block (DSB) is used by the writer to print SYSOUT data. If DSO is active, the DSB Processing routine turns off each of the active bits of the step's Direct System Output Control Blocks (DSOCB).	IEFYTVMS	3-32
2 The Step Termination Disposition routine performs the disposition specified in the DISP field of the DD statement associated with the data set.	IEFZGST1	
4 The Step Termination Deallocation routine gets a DD entry from the Task Input/Output Table (TIOT), processes any pending commands (UNLOAD, VARY ONLINE/OFFLINE,MCS) then deallocates all Unit Control Blocks (UCB) for each DD entry. The following kinds of volumes are not demountable: • Public volumes • System residence volumes • Volumes mounted on reserved units • Permanently resident volumes • Volumes whose status is 'RETAIN' • Volumes mounted on system input or system output units • Volumes containing data sets having PASS disposition • Volumes whose user count is greater than zero	IEFZGST2	
If the DISPLAY DSNAME command is in effect, the demount messages include the names of the first eligible non-temporary data sets associated with the TIOT device entries for which demount messages are to be issued.	IEFZHMSG	

Extended Description	Module	Figure
5 The deallocation routine (IEFZGST2) provides a parameter list for the User Accounting Routine Linkage routine (IEFACTLK). The list contains pointers to the Queue Management Parameter Area (QMPA), the Job Management Record (JMR), an identification indicating whether it is job or step termination, and the Logical Record Control Block (LRCB).	IEFZGST2	
6 The Job Statement Condition Code Processor (IEFVJIMP) determines whether or not a job should be terminated when an error condition previously specified in the JOB statement has been detected. If it is to be terminated, the processor issues a WTO and a message to the system message data set. The Step Termination Exit routine (IEFSD22Q) places the address of the Step Control Table (SCT) for the next step in the Linkage Control Table (LCT), checkpoints the data set descriptor (DSD), then posts the Event Control Block (ECB) in the allocate/IEFVPOST communications block.	IEFVJIMP IEFSD22Q	

Figure 2-40. Abnormal Step Termination with Restart



Extended Description	Module	Figure
1 If the step is eligible to be restarted, a message is issued to the operator, asking permission to restart the step. If the step is not to be restarted, the procedure for abnormal step termination without restart is followed. If restart is to take place, all passed data sets for the failed step are taken off the Passed Data Set Queue (PDQ).	IEFRPREP	
2 See Figure 2-39.	IEFYTVMS	
3 If a disposition is not specified, conditional or through the DISP field of the DD statement, the disposition routine issues the SCRATCH macro for NEW data sets created for the failed step, and the KEEP message for OLD ones.	IEFZGST1	

Extended Description	Module	Figure
4 See Figure 2-39.	IEFZGST2	
5 The User Accounting Routine Linkage routine provides the accounting routine with the same information as in normal step termination.	IEFACTLK	
6 The last step of a job also follows the procedure for normal step termination, instead of the procedure for job termination.		3-31

Figure 2-41. Abnormal Step Termination Without Restart (Part 1 of 2)

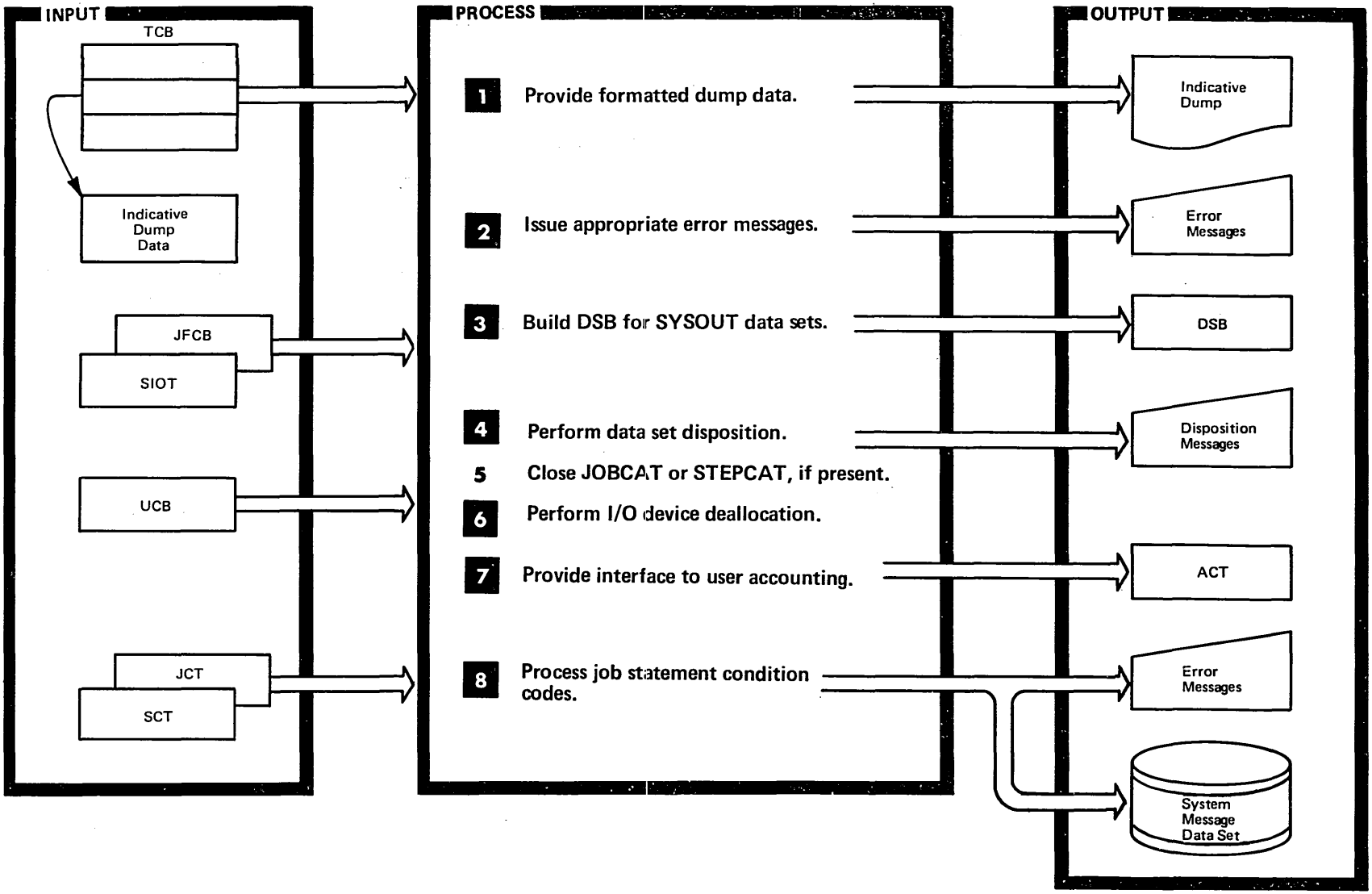


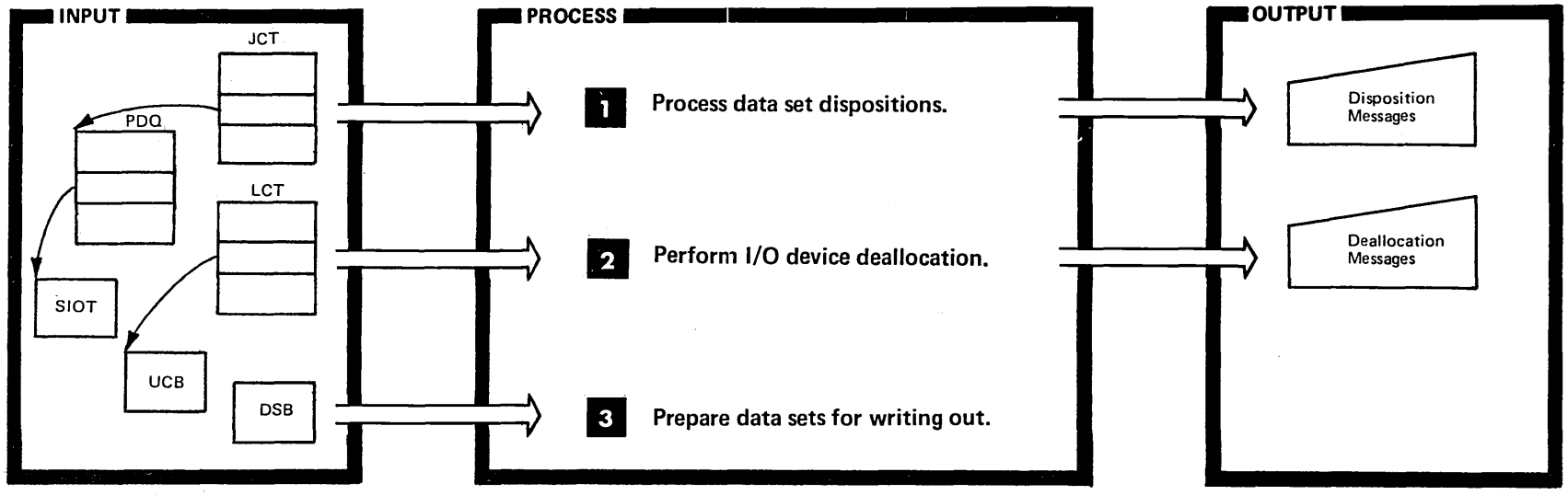
Figure 2-41. (Part 2 of 2)

Extended Description	Module	Figure
1 The Indicative Dump routine provides the user with an indicative dump to aid in debugging.	IEFIDUMP	
2 The Step Termination Control routine determines the cause of abnormal termination of the step, issues the appropriate message, and sets the appropriate bits in the Job Control Table (JCT), if necessary, to indicate the type of failure to other scheduler components.	IEFYNIMP	3-32

Error	Message issued	Bits in JCTJSTAT
CANCEL command	WTO	Job-failed
ABEND		ABEND and job-failed
In initiator processing	to system message data set	Job-failed

Extended Description	Module	Figure
3 See Figure 2-39.	IEFYTVMS	3-32
4 If a disposition is not specified, conditional, or through the DISP field of the DD statement, the disposition routine issues the SCRATCH macro for NEW data sets and the KEEP message for OLD ones.	IEFZGST1	3-32
6 See Figure 2-39.	IEFZGST2	3-32
7 The User Accounting Routine Linkage routine provides the accounting routine with the same information as in normal step termination.	IEFACTLK	3-32
8 The Job Statement Condition Code Processor determines whether or not a job should be terminated when an error condition previously specified in the JOB statement has been detected. It then issues a WTO and a message to the system message data set, and proceeds as in Figure 2-39.	IEFVJIMP	3-32

Figure 2-42. Job Termination



Extended Description		Module	Figure
1	The Job Disposition and Deallocation routine locates any passed data sets that were never received. If termination is normal, it deletes data sets created during the job and keeps data sets created prior to the job. If termination is abnormal, it performs the conditional disposition if specified; otherwise, it handles disposition processing as for normal termination.	IEFZGJB1	3-33
2	The Disposition and Deallocation Message routine writes disposition messages to the system message data set and processes any pending commands.	IEFZHMSG	3-33
	The Job Disposition and Deallocation routine performs deallocation.	IEFZGJB1	3-33

Extended Description		Module	Figure
3	The Job Terminate Exit routine closes the system message data set, enqueues it and handles any writing that was done since the job was interpreted, issues a job-ended or job-failed message for abnormal termination, deletes the SYSIN data sets, and returns control to the initiator. The Exit routine links to the DSO writer (IEFDSOWR) if DSO is active to write out system messages.	IEFSD31Q	3-33

Figure 2-43. Writing Job Output (Part 1 of 2)

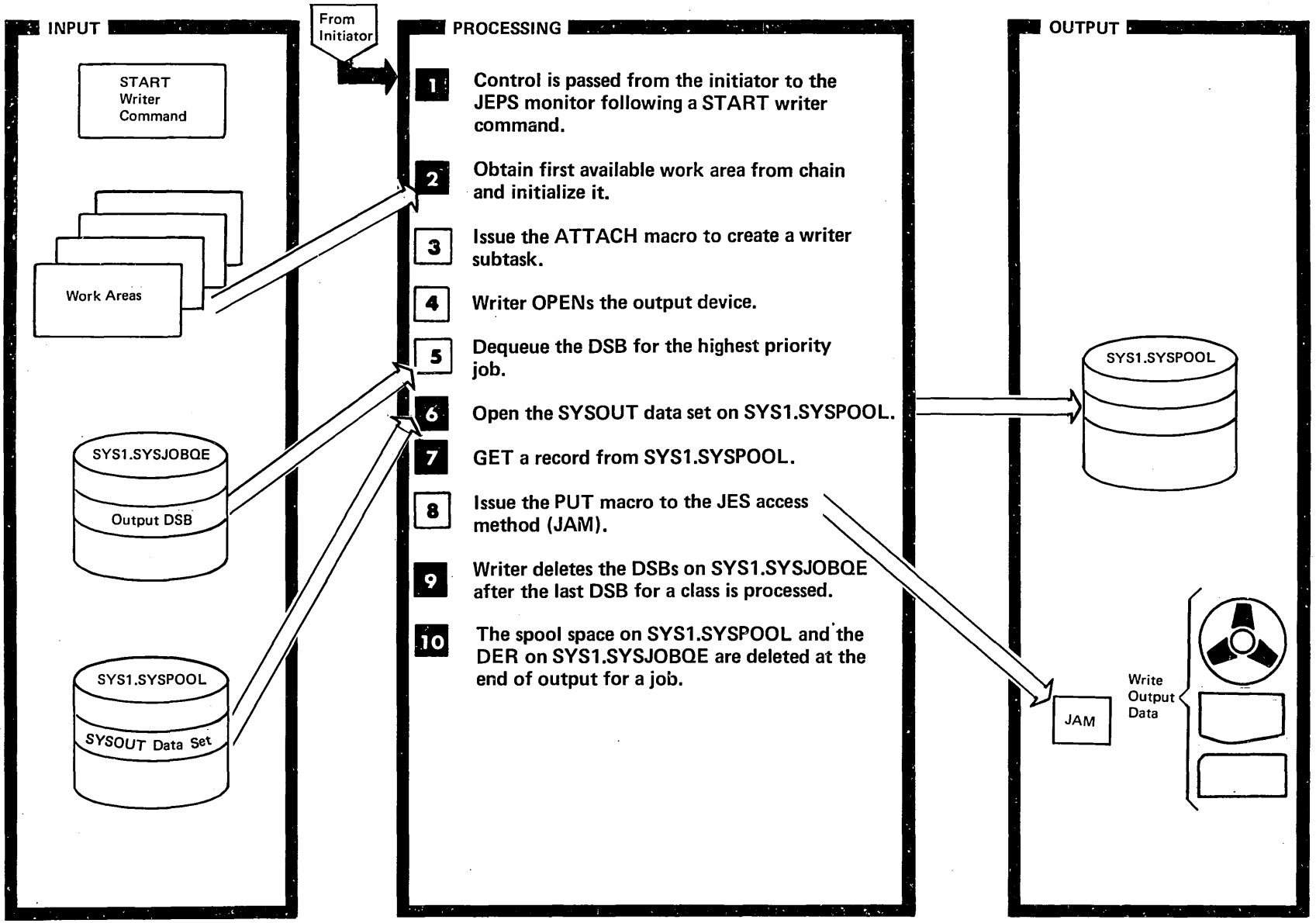


Figure 2-43. (Part 2 of 2)

Extended Description	Module	Figure
1 The JEPS monitor obtains a work area and initializes it with information in the initiator parameter area. Validity check of the user parameters in the writer procedure is made. Printer, punch, or tape output is determined.	IEFVMA	3-39
	IEFVMD	3-39
2 The JEPS monitor initializes the:	IEFVMD	3-39
• JECS parameter lists for the necessary parameter lists required for the writer.		
• ACB, RPL, and EXLST, which are required by data management to pass data to the output device.		
• CSCB, which is posted for the CANCEL and writer commands.		
• TIOT, which is used by data management to describe the output device.	IEFVMA	3-39

Extended Description	Module	Figure
6 OPEN is one of six JECS macros issued by the spool manager to access, retrieve, open, and close the data sets on SYS1.SYSPOOL.	IEFSMIFC	3-40
7 The spool manager obtains the buffers to hold the output data sets and then deblocks the data. The work area manager performs the necessary I/O operations.	IEFSMGET	3-49
9 The writer invokes the queue manager to delete all DSBs for the job class being processed after the last DSB for a job class has been processed.	IEFJESQM	3-54
10 The SYS1.SYSPOOL space associated with the job is freed after the last data set for the job has been processed.	IEFSMCLD	3-51

Figure 2-44. System Restart (Part 1 of 2)

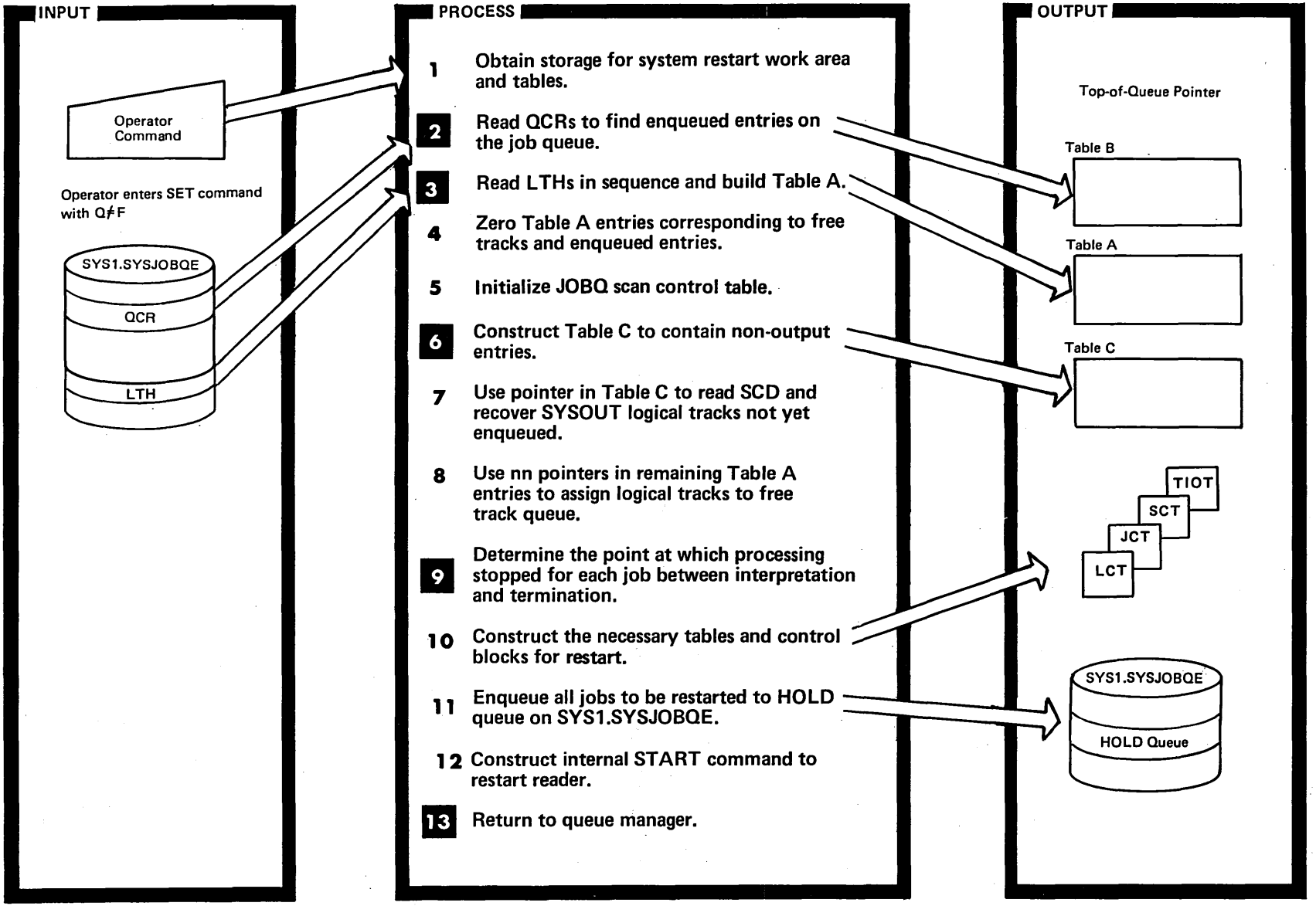


Figure 2-44. (Part 2 of 2)

Extended Description		Module	Figure
2	The top-of-queue pointer is extracted from each QCR to build Table B.	IEFSD300	3-35
3	Convert each two-byte pointer (in nn format) in the LTH to a relative address and place it in Table A.	IEFSD300	3-35
6	Scan Table A to determine the status of entries on the queues. A Table C entry is created for all non-output queue entries. This includes incomplete input, dequeued input, enqueued input, and all HOLD queue entries.	IEFSD300	3-35

Extended Description		Module	Figure
9	Link to termination to determine the point at which processing stopped. The alias name used by system restart is IEFW42SD.	IEFSD42Q	3-32
13	Issue the MGCR macro to schedule execution of the restart reader if the job is eligible for restart; otherwise, enqueue SYSOUT for the job if it is not restarted.	IEFVSDRA	3-32

Figure 2-45. Scheduler Automatic Step Restart (Part 1 of 2)

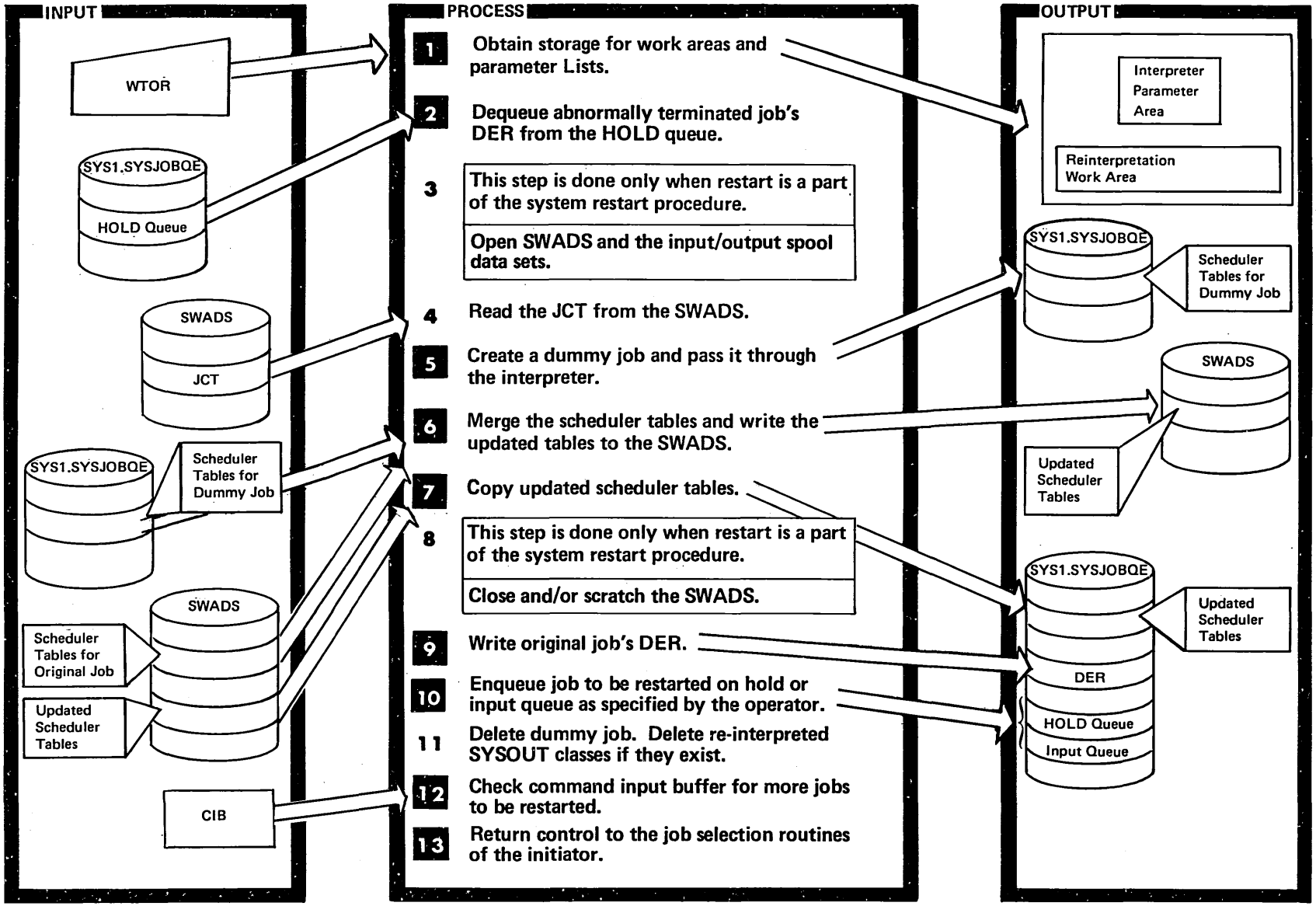
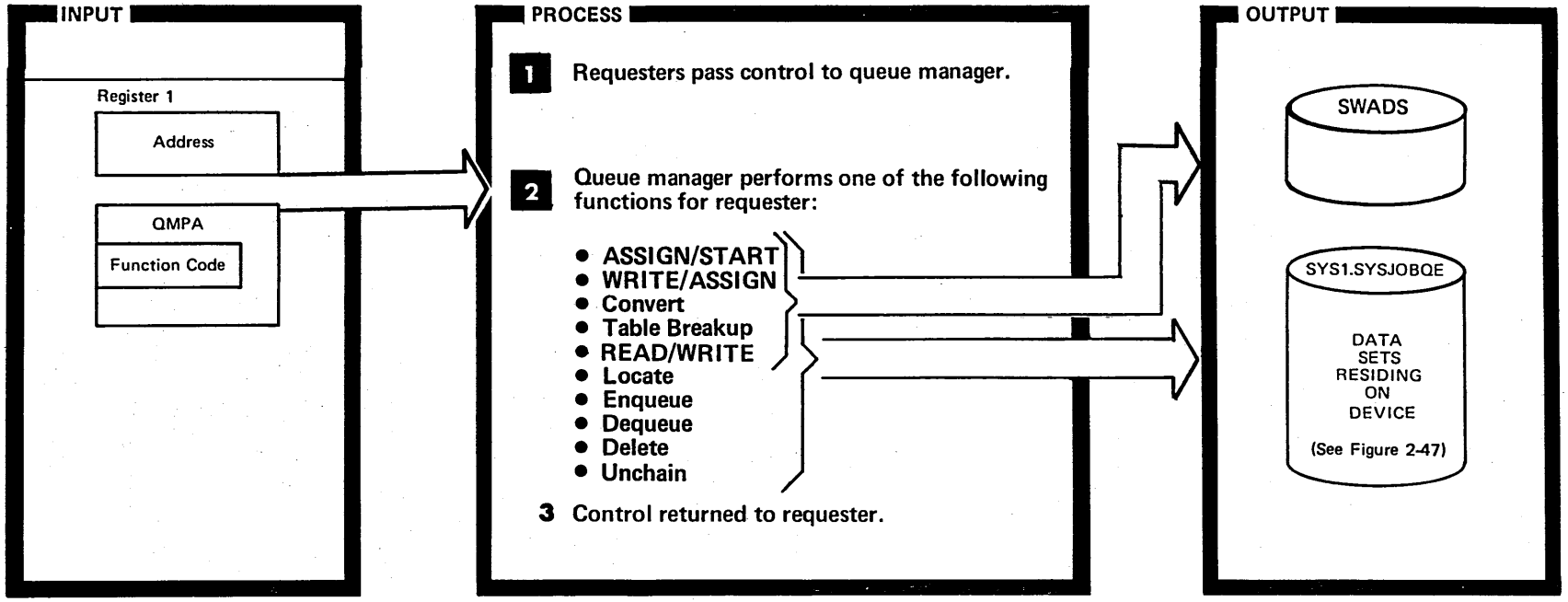


Figure 2-45. (Part 2 of 2)

Extended Description	Module	Figure
1 The job name of the job to be restarted (in the CIB) is placed in the reinterpretation work area.	IEFVRRC	3-37
2 Convert nn to MBBCCHHR for the DER and create the QMPA for the SWADS.	IEFVRR1	3-37
5 The user parameters passed to the interpreter are processed and entered in the DER. The interpreter parameter list is then completed by supplying pointers to the QMPA and to the DER, and placing the entry point for the interpreter to XCTL to the restart reader routines. A DER is created for a dummy job and the interpreter constructs the scheduler tables for the dummy job.	IEFVRRC	3-37

Extended Description	Module	Figure
6 Table merge routines.	IEFVRR2	3-37
7 Copy tables		
9 10 Write and enqueue the DER.	IEFVRR3	3-37
12 If there are more jobs to be restarted, return to step 2.	IEFVRR3	3-37
13 This step is executed only when there are no more jobs to be restarted.	IEFVRRC	3-37

Figure 2-46. Queue Manager



Extended Description		Module	Figure
1	Module IEFQMJOI determines the function to be performed by the queue manager. Requests for queue manager services come from: Allocation, Data Management, Initiator, Interpreter, Queue Command, Reader, Restart, System Log, System Management, Termination, Warm Start, and the Writer.	IEFQMJ01	3-54
2	ASSIGN/START, ASSIGN	IEFQMJ01	3-54
	WRITE/ASSIGN, READ/WRITE	IEFQMJ01	3-54
	DEQUEUE, DELETE	IEFQMJ02	3-54
	CONVERT, UNCHAIN, TABLE BREAK-UP, READ/WRITE, LOCATE	IEFQMJ03	3-54

Figure 2-47. Format of SYS1.SYSJOBQE (Part 1 of 2)

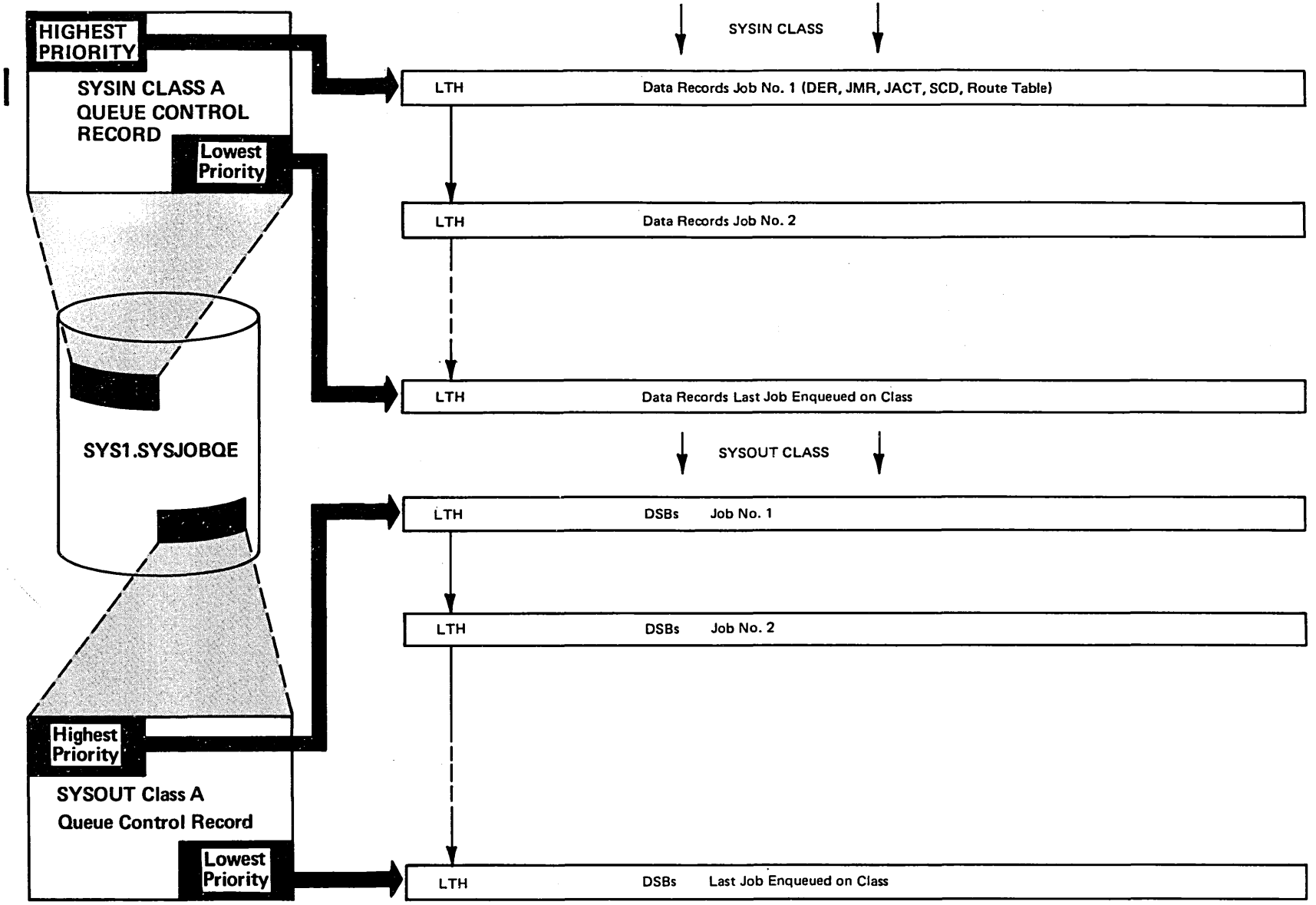


Figure 2-47. (Part 2 of 2)

SYS1.SYSJOBQE contains records of the following types:

Queue control records	36 bytes
Logical track header records	176 bytes
Disk entry records	176 bytes
Job management records	176 bytes
Data set blocks	176 bytes

SYS1.SYSJOBQE contains 76 work queues plus additional queues if RTAM is generated into the system.

- 2 HOLD queues (one input, one output)
- 36 output class queues
- 15 input class queues
- 22 reserved queues
- 1 free space queue
- 1 SYS1.SYSPOOL configuration control queue

For RTAM:

- 1 output HOLD queue
- 36 output class queues

} for each QID

The entries on SYS1.SYSJOBQE fall into two types: input and output. There are 15 input classes and 36 output classes. Each input job or class of output data sets can be assigned a different or the same priority.

For each class of information residing on this device there is a Queue Control Record (QCR). The QCR points to the last LTH of the highest priority entry on the single queue it controls. The QCR also contains pointers to the last LTH of the latest entry on the queue for each priority from 14 to 0.

The QCR for the free space queue (called the Master QCR) contains a pointer to the first available logical track and the number of available tracks remaining. In addition, it contains control information for the entire job queue data set.

The spool configuration QCR contains the volume serial number of the first volume of the spool data set. This volume is the Spool Control Volume and contains information for the entire spool data set.

All logical tracks for a job's input queue entry are chained together; however, most jobs have only one logical track. The same is true for logical tracks for a job output queue entry. All jobs of the same input job class or SYSOUT class are chained together.

The disk entry record for input jobs contains a pointer to the job management record for that job.

Every data set block for an output data set for a job contains a pointer to the disk entry record for that job.

Figure 2-48. JECS Overview (Part 1 of 2)

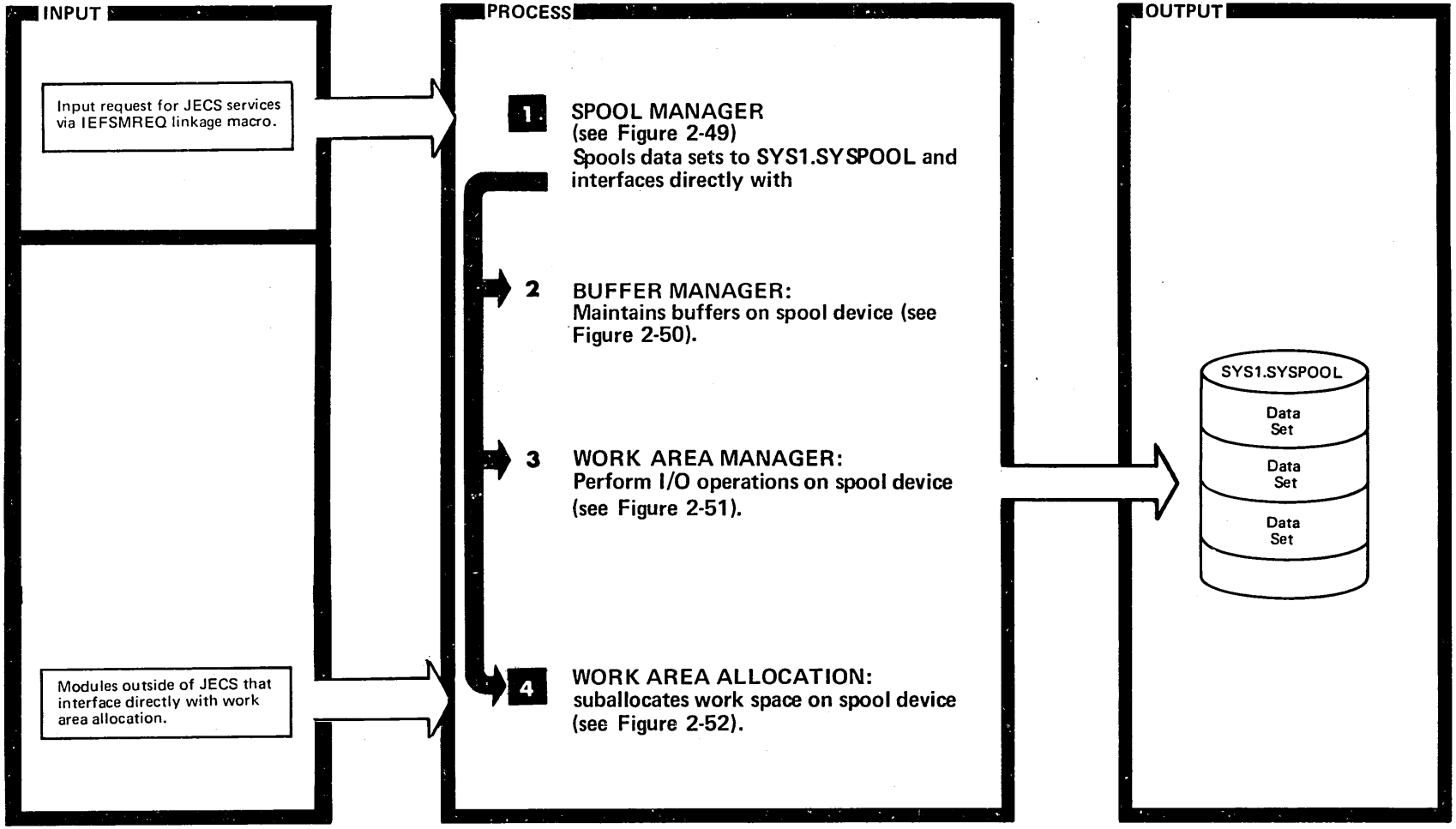
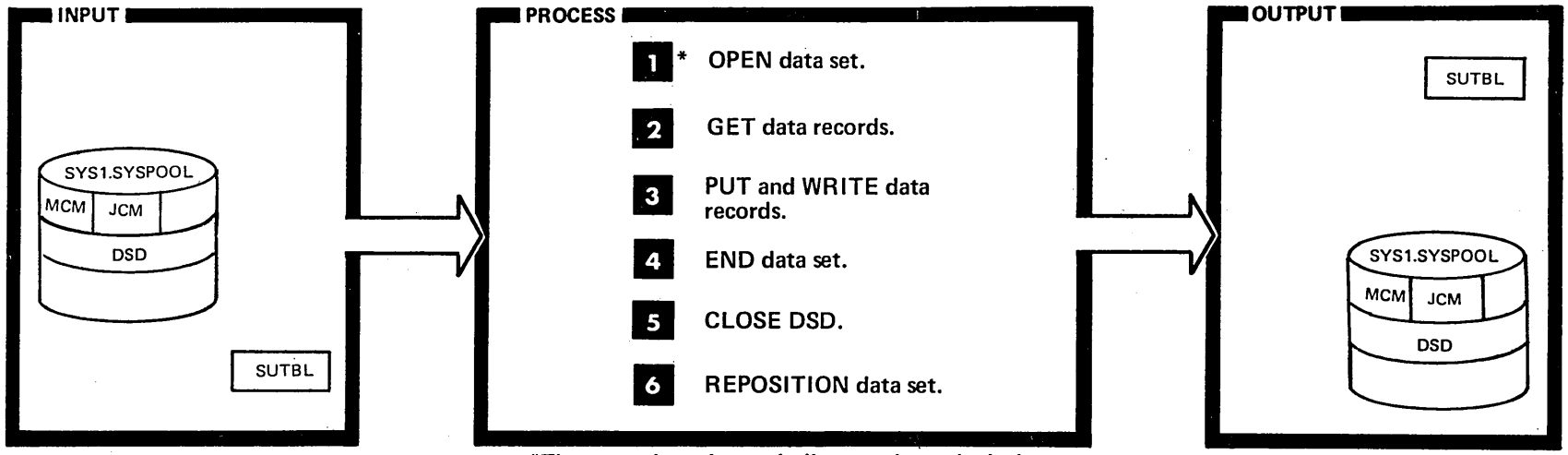


Figure 2-48. (Part 2 of 2)

Extended Description		Module	Figure
1 The spool manager handles sequential data on a logical record basis. The data sets handled include job control language text, in-core input data, procedures (from private libraries), system messages, and spooled data sets. The spool manager consists of six macros to perform the following functions:			
OPEN	Search the Spool Usage Table for job ID.	IEFSMODS	3-48
GET	Assign buffers for data, retrieve records from SYS1.SYSPOOL and move a logical record into user's buffer.	IEFSMGET	3-49
PUT	Assign buffers for data, move user data to buffer, and write buffer to SYS1.SYSPOOL.	IEFSMPUT	3-50
CLOSE OPEN DICTIONARY	Checkpoint and free the job-cylinder map and data set dictionary.	IEFSMCLD	3-51
END DATA SET	Update data set dictionary entry, write out any partial buffers to SYS1.SYSPOOL, return buffers to pool, and record SMF statistics.	IEFSMEND	3-52
REPOSITION DATA SET	Assign buffer for request, read buffer, and reset tables.	IEFSMREP	3-53
4 Work area allocation also maintains both the master and job cylinder maps for SYS1.SYSPOOL. The following VS1 components interface directly with work area allocation to maintain these maps: termination, initiator, interpreter, JES readers and writers, direct system output, express cancel, and MSI restart.			

Figure 2-49. Spool Manager



*These numbers do not indicate a chronological order.

Extended Description	Module	Figure
1 Search the SUTBL for the job's identification. Read the DSD from SYS1.SYSPOOL, if a TTR was passed in the RPL; otherwise, get storage for the DSD and the CWA and initialize the DSD and the LRCB.	IEFSMODS	3-48
2 Obtain a buffer to store data retrieved from SYS1.SYSPOOL. Unblock records and pass them to requester.	IEFSMGET	3-49
3 Obtain a buffer to store data passed by the requester and fill buffers requesting allocation of a logical cylinder on DASD. Write buffers to the allocated cylinder.	IEFSMPUT	3-50

Extended Description	Module	Figure
4 Update the DSD entry and return any buffers the user may have to buffer management. Free the CWA and record SMF statistics.	IEFSMEND	3-52
5 Checkpoint the internal spool tables reflecting the user's job and then release these tables. Checkpoint the MCM, JCM, and the DSD. Issue a FREEMAIN for the JCM and DSD.	IEFSMCLD	3-51
6 Obtain a buffer and restart a checkpointed job by positioning the data set to a point where the checkpoint was taken.	IEFSMREP	3-53

Figure 2-50. Buffer Manager (Part 1 of 2)

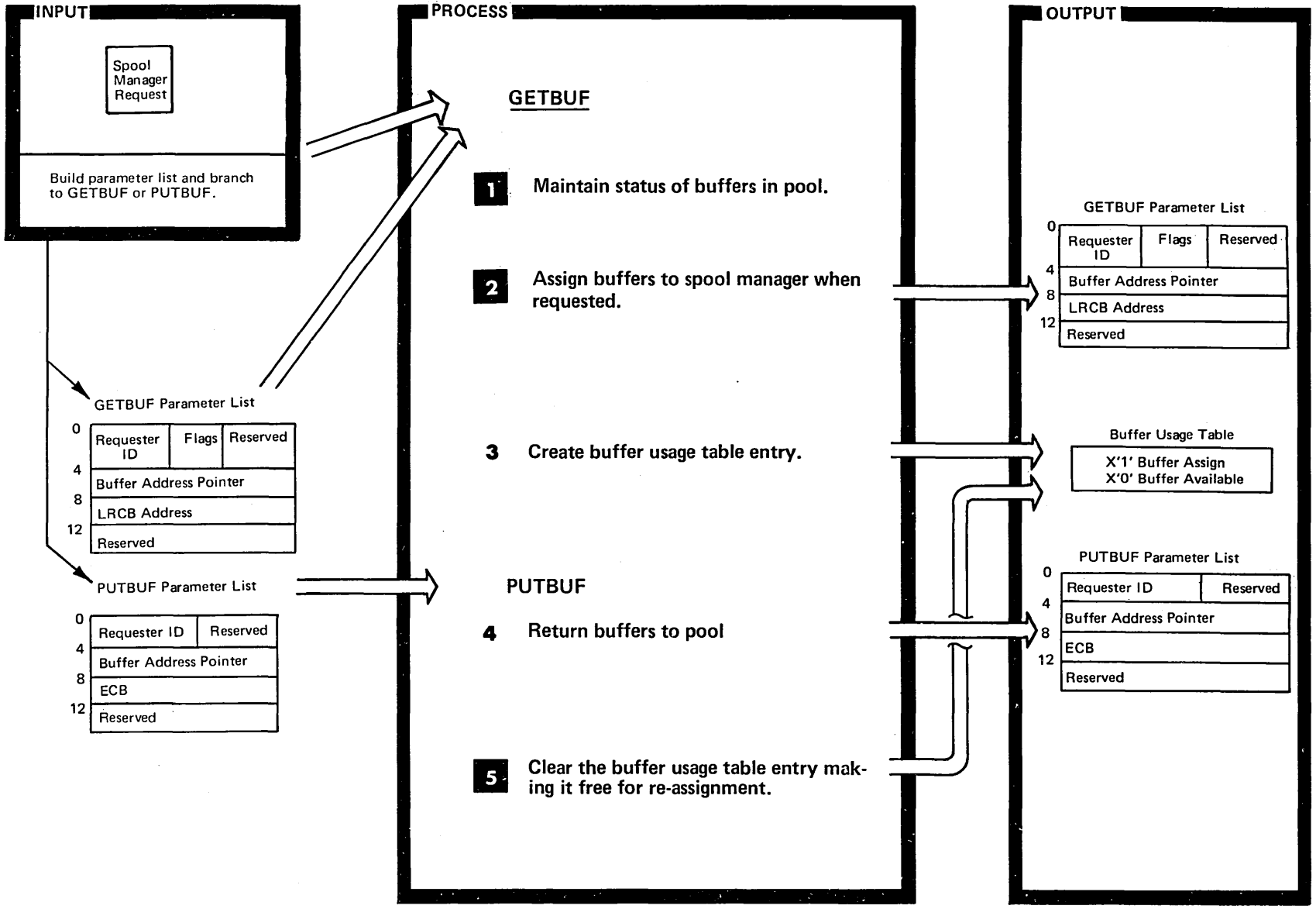


Figure 2-50. (Part 2 of 2)

Extended Description	Module	Figure
1 The buffer pool is located in the JES control program area. The user can specify the number of buffers as a SYSGEN option or by specifying a NUMBUFS parameter for SYS1.PARMLIB. The number of buffers can be changed at IPL time via the SET command. The size of the buffers is specified at SYSGEN time. The minimum size is 436 bytes.		
2 **5** When the spool manager requires a buffer, the buffer address pointer in the GETBUF parameter list specifies the address of an available buffer. When the buffer is no longer required, the address pointer in the PUTBUF parameter list specifies the address of the buffer to be returned to the pool.	IEFBMGET IEFBMPUT	3-49 3-50 3-53

Figure 2-51. Work Area Manager (Part 1 of 4)

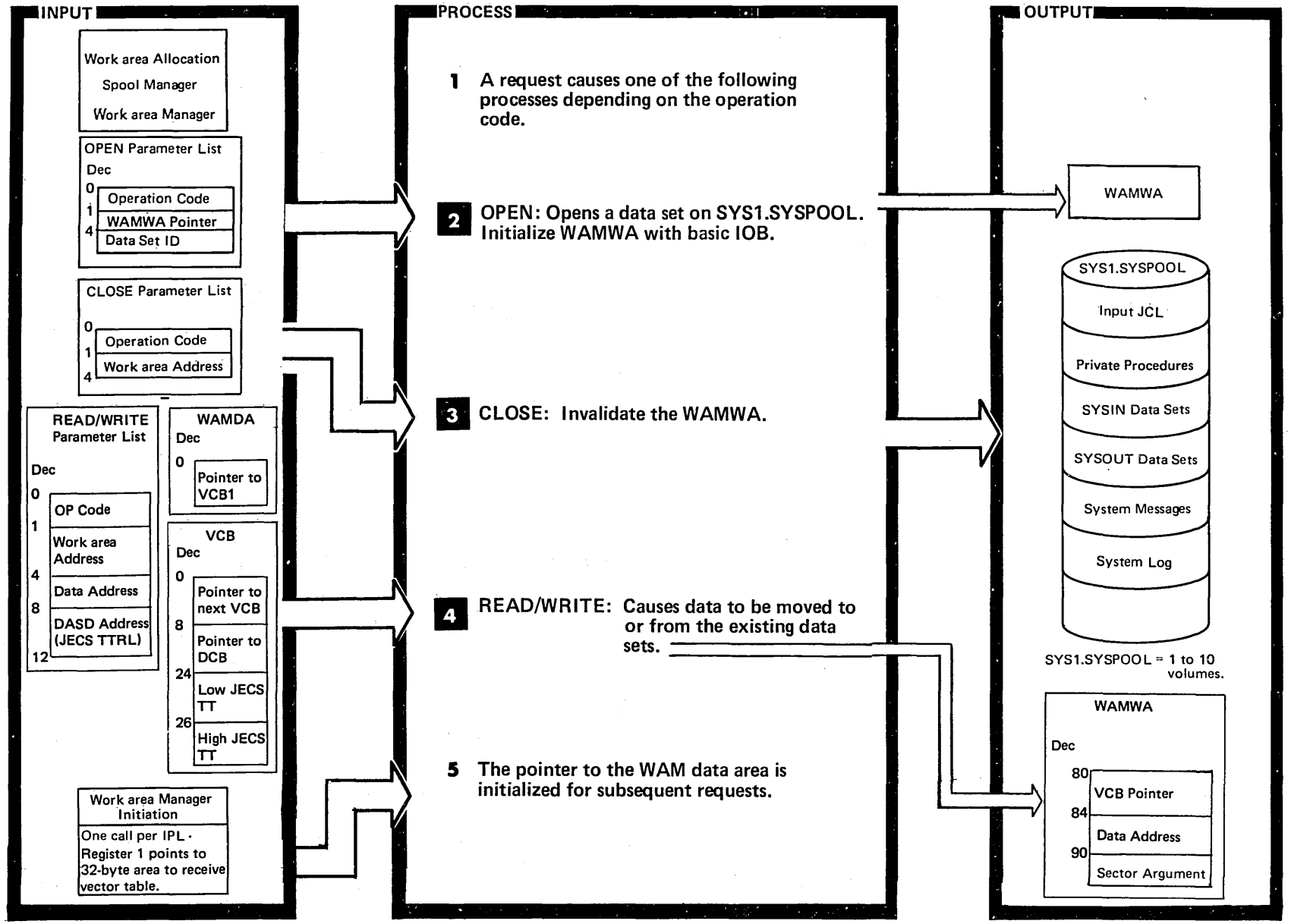


Figure 2-51. (Part 2 of 4)

Extended Description	Module	Figure
2 If end of work area is above "virtual equals real" line, ensure that CCWs within the work area do not overflow a page. For overflow conditions, a remote work area is provided in SQS space. WAMWA is initialized with basic IOB.	IEFWAMGR	3-48
3 If there is a remote work area, the storage used is freed. The pointers to all existing work areas are reset so that subsequent read or write requests will not be executed.	IEFWAMGR	
4 READ/WRITE Functions: • The spool configuration is made up of from one to ten volumes each of which contains a SYS1.SYSPOOL data set. Associated with each volume is a volume control block (VCB). JECS TTs represent a track displacement into the spool configuration. Each VCB contains the JECS TT range for its respective volume (IEFVCBLT - low JECS TT on the volume; IEFVCBHT - high JECS TT on the volume). Using the input DASD address (JECS TT), the correct VCB is found and left in WAMWA. The pointer to the DCB, which is found in the VCB, is placed into the IOB. • Set the IOB start pointer (points to start of CCW chain) to either the set sector CCW (for RPS devices) or the search CCW.		

Extended Description	Module	Figure
• The JECS TTRL has the following format: TT - two byte JECS TT, the track displacement into the spool configuration. R - one byte record number L - one byte logical record number (used by spool management to access data within a buffer). Convert the JECS TTRL to a DASD TTRL as follows: TT = JECS TT - IEFVCBTT R = R (no change) L = 0 Convert the DASD TTRL to absolute disk address by using the relative address convert routine (CVTPCNVT). For RPS devices, the R is an index into the sector table pointed at by the VCB (IEFVCBSC). The sector argument associated with R is moved into WAMWA. • If READ/WRITE is for less than buffer size, flag work area for channel end appendage. • Set data length in READ/WRITE CCW and data pointer in work area for page fix routine. • Issue SVC 114 via EXCPVR macro. • Return to requester.		

Figure 2-51. (Part 3 of 4)

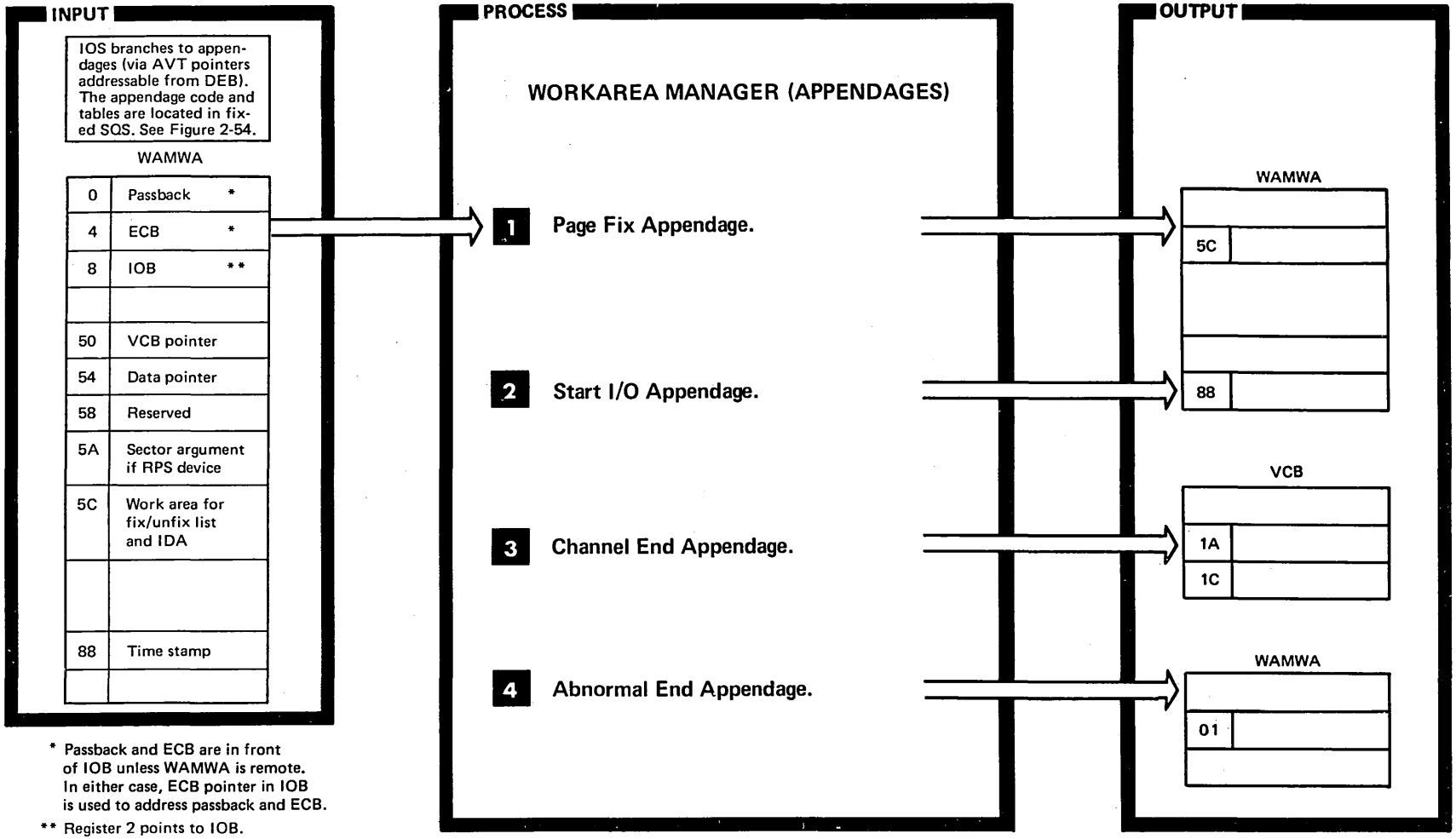


Figure 2-51. (Part 4 of 4)

Extended Description	Module	Figure
1 FIX/UNFIX pages that contain the work area manager work area (WAMWA) and I/O buffer. The FIX/UNFIX list is built in WAMWA. The page fix appendage takes the time stamp and saves it in the WAMWA.	IEFWAMAP	
2 Translate CCWs and build indirect address list (IDA) in WAMWA.	IEFWAMAP	
3 Calculate the time increment from the first entry to page fix to channel end and add it to the accumulator in VCB (see Figure 2-52). Increment EXCPVR counter in VCB.	IEFWAMAP	

Extended Description	Module	Figure
Detect any short READ/WRITE request so that abnormal end appendage is not entered.		
4 For a permanent I/O error, the sense bits in the IOB are tested for data check, data check in count area, or track overrun, and a bit in passback is set so that the caller will interface with work area allocation to delete the logical cylinder causing the error.	IEFWAMGR	

Figure 2-52. Work Area Allocation (Part 1 of 3)

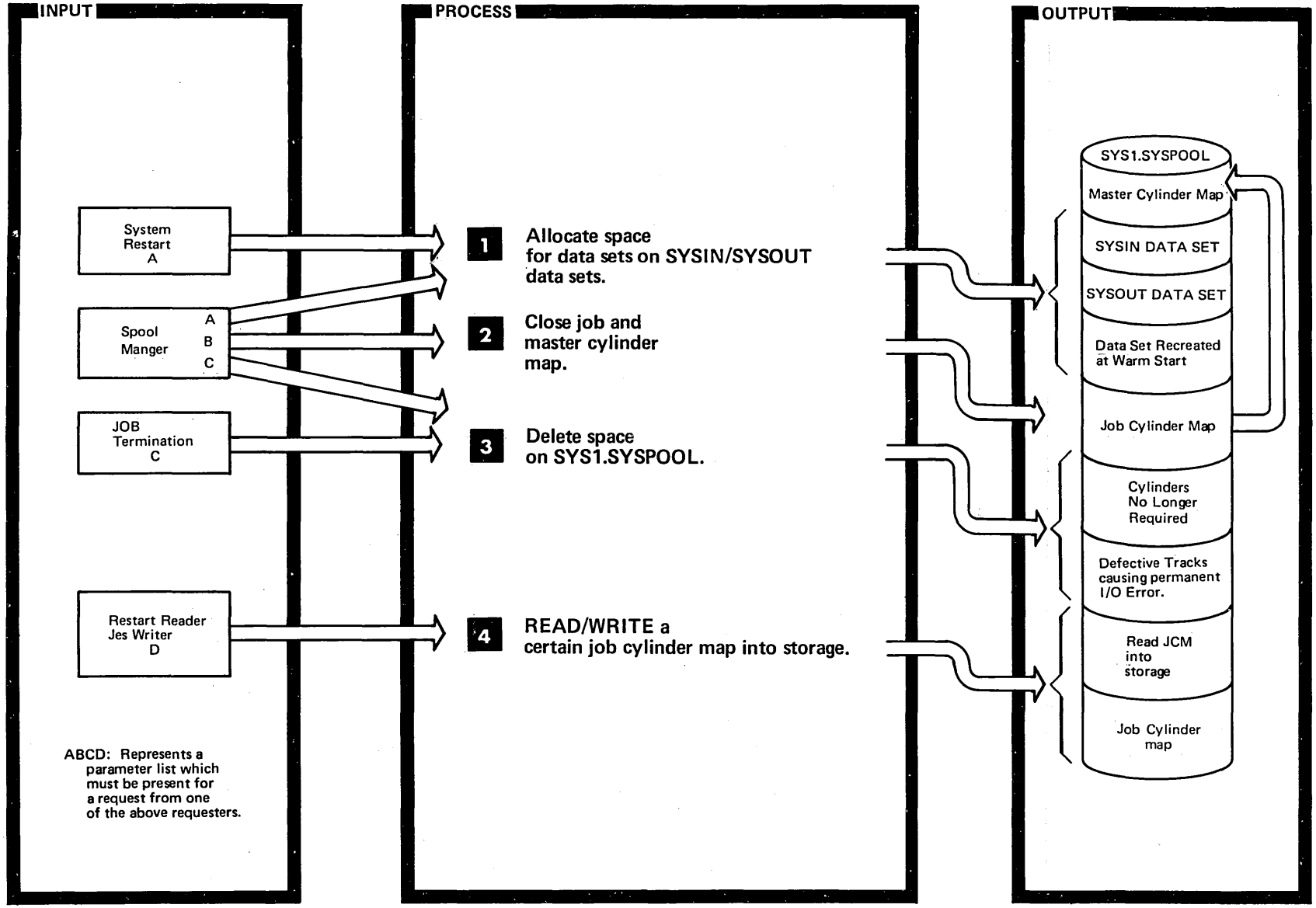


Figure 2-52. (Part 2 of 3)

Extended Description		Module	Figure
1	An allocate function suballocates DASD work space to the logical data sets created by the spool manager. A specific allocation function allocates specific space requested if the space is available.	IEFWAA01 IEFWAA03	
2	A CLOSE function checkpoints the JCM/MCM and, if indicated, releases the storage occupied by the JCM.	IEFWAA02	
3	A DELETE function makes used cylinders available to the system again for allocation. A specific DELETE function deletes a specific logical cylinder from SYS1.SYSPOOL.	IEFWAA02 IEFWAA03	
4	A READ function reads the job cylinder map into storage. A WRITE function writes the job cylinder map onto DASD.	IFFWAA03	

Figure 2-52. (3 of 3)

A Parameter List for Allocate and Specific Allocate

Operation code X'01'
Flags: X'00' Allocate X'01' Specific Allocate X'10' Obtains core in SQS for JCM X'20' Spool critical-conditionally allocates a logical cylinder X'40' Required to checkpoint JCM, MCM, and allocate passback
Request ID passed to Spool Manager at time Open Data Set. (See Figure 3-48)
Passback residual number of tracks available in allocated logical cylinders
Job Cylinder Map TTRL
Current/Specific TTRL Allocate input-last TTRL used Allocate passback next TTRL Allocate specific - specific TTRL
Passback track capacity (number of records)
Error passback (see list below)
Reserved

B Parameter List for Close (Checkpoint)

Operation code X'02'
Flags: X'10' Release core in SQS for JCM X'20' JCM has increased beyond buffer size after system restart X'40' Required to checkpoint JCM, MCM, and allocate passback X'80' To free JCM space on a destructive close (See Figure 3-51)
Reserved
Job Cylinder Map TTRL
Reserved
Error Passback (see list below)
Reserved

C Parameter List for Delete/Specific Delete

Operation code X'04'
Flags: X'00' Delete X'01' Specific delete
Reserved
Job Cylinder Map TTRL
Reserved/specific TTRL to delete
Reserved
Error passback (see list below)
Reserved

D Parameter List for READ/WRITE

Operation code X'08'
Flags: X'80' Read JCM X'40' Write JCM (See Figure 3-49 and 3-50)
Reserved
Job Cylinder Map TTRL
I/O area pointer for area into which to read and write
Reserved
Error passback (see list below)
Reserved

Error Passback Codes and Meanings

X'0000'	Function successfully performed
X'8000'	Invalid TTRL reference
X'4000'	GETMAIN not satisfied
X'2000'	I/O error in writing JCM
X'1000'	I/O error in reading JCM
X'0800'	Specific allocate cylinders not available
X'0200'	Invalid Operation code
X'0100'	Invalid Spool volume concontrol block ID. Internal tables are destroyed.
X'0080'	JCM dictionary overflow
X'0040'	No spool space available
X'0020'	Spool space critical

Figure 2-53. Work Area Manager Initialization (Part 1 of 4)

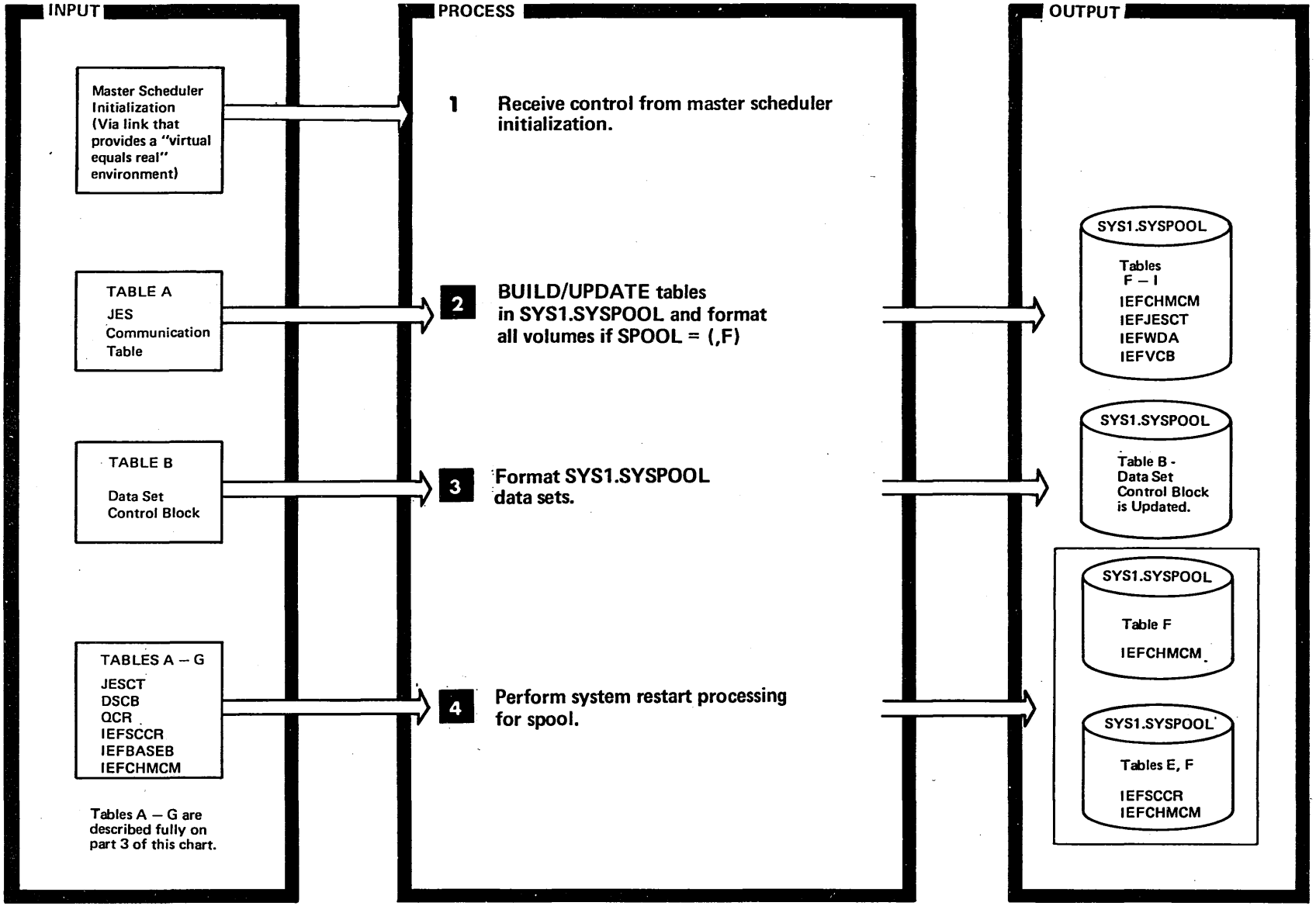


Figure 2-53. (Part 2 of 4)

Extended Description	Module	Figure
2 Moves WAM tables and I/O appendage code to fixed SQS.	IEFWAMAP	2-54
3 The SYS1.SYSPOOL data sets are formatted based on the following tests: a. If SPOOL = (,F), format all volumes, b. Otherwise, read DSCB for each volume and test for a need to format.	IEFWAMIN	2-52
4 The following steps are necessary to perform a system restart: a. Tables A - E are required to verify that the necessary volumes are mounted and correctly formatted.		

Extended Description	Module	Figure
b. Tables D and E are required to verify VOLID of SYS1.SYSJOBQE. c. Table F (IEFCHMCM) will update the MCMs in storage with the checkpointed MCM. The output is Table F. d. Checkpoint work area manager tables required for system restart.	IEFWAMIN	2-52

Figure 2-53. (Part 3 of 4)

A

TABLE 1 (Input Only)

IEFJESCT:	JES Communication Table (In storage via CVT)
Displacement	
X'30'	**JESUNT1,** high-order bit ON implies spool = (,f) was specified. Low order 3 bytes point to JESUNITS.
X'3C'	**JESBFSIZ,** sysgen/ JESPARMS buffer size.
X'60'	**JESSMNBR,** number of JCM pointer entries to be provided in IEFWDA.
X'62'	**JESUNITS'** up to 10 ordered pairs of (VOLID, UCBPTR) such that if UCBPTR=0, volume is not mounted.
X'B2'	JESUNTDL Table delimiter

B

TABLE 2 (Input Only)

DSCB:	Data Set Control Block (read from VTOC for each VCL in SC).
Displacement	
X'56'	DS1BLK1, contains buffer size if data is formatted.
X'62'	DS1LSTAR, first two bytes indicate if data set is formatted. Third byte gives buffers per track.
X'69'	DS1EXT1, extent information used to read records from SYS1.SYSJOBQE.

C

TABLE 3 (Input Only)

QCR:	Queue Control Record (nn = 56 read from SYS1.SYSJOBQE).
Displacement	
X'08'	Volume ID of SCCV

D

TABLE 4 (Input Only)

IEFBASEB (in storage via CVT)	
Displacement	
X'10'	BAQ, UCB pointer to volume containing SYS1.SYSJOBQE.
X'18'	BACV, high bit indicates warm start.

E

TABLE 5 (Input/Output)

IEFSCCR:	Spool Control Configuration Record (read from record 1).
Displacement	
X'00'	SVOLIDS, space for 10 fields of 7 bytes each. Each field contains an ordered pair (VOLID, BUFFER/TRACK).
X'46'	QVOLID, volume ID of SYS1.SYSJOBQE.
X'4C'	TTRMCM, TTR pointer to checkpoint MCM.
X'50'	TTRLAST, TTR pointer to next available record on SCCV.
X'54'	BUFFERSZ, size of buffer.
X'58'	LCYLCNT, number of logical cylinders.
X'5C'	MCMLENGT, size of checkpoint MCM (including header).

F

TABLE 6 (Input/Output)

IEFCHMCM:	Checkpointed MCM (read from SCCV).
Displacement	
X'00'	CHMCMID, identifier for Master Cylinder Map
X'04'	Disk address of checkpointed MCM
X'08'	CHMCMTHR, percent of allocated logical cylinders for next warning message.
X'0A'	CHMCMCTR number of allocated logical cylinders.
X'0C'	Reserved
X'0E'	MCM bits

G

TABLE 7 (Input/Output)

IEFJESCT:	JES Control Table
Displacement	
X'24'	**JESWAMDA,** pointer to IEFWCA.
X'3A'	JESJOBCM, MCM length (including header).
X'3C'	JESBFSIZ, reset if spool = (,f) was specified and input size exceeds track capacity for a volume.
X'62'	JESUNITS, reset with (VOLID/UCBPTR) pairs for volumes added at warm start.

Figure 2-53. (Part 4 of 4)

H

TABLE 8 (Output Only)

IEFWDA:	
DISP	
X'00'	IEFWDAVB, pointer to first VCB.
X'04'	IFFWDARK, device rank chain header set to 0.
X'08'	IEFWDACN, allocation chain request header set to 0.
X'0C'	IEFWDAWM, pointer to work area manager interface area.
X'10'	IEFWDAMC, checkpointed MCM storage address.
X'14'	IEFWDAMD, checkpointed MCM TTB.
X'18'	IFFWDAMR, resident MCM storage address.
X'1C'	IFFWDAIO, pointer to I/O work buffer.
X'20'	IEFWDACC, total number of logical cylinders.
X'24'	IFFWDATT, highest TT for SC.
X'28'	Reserved.
X'30'	JESSMNBR, variable number of JCM pointers, storage address and disk address for each.

I

TABLE 9 (Output Only)

IEFVCB (one for each spool volume).	
DISP	
X'00'	IEFVCBCH, pointer to next VCB and 0 for last VCB.
X'04'	IEFVCBRC, pointer to next VCB in device rank chain.
X'08'	IEFVCBIO, pointer to DCB for the volume.
X'0C'	IEFVCBUB, pointer to UCB for the volume.
X'10'	IEFVCBSC, for RPS device, pointer to associated sector table, 0 for one RPS device.
X'14'	IEFVCBTM, average EXCPVR time set to 0.
X'18'	IEFVCBTS, time accumulator set to zero.
X'1C'	IEFVCEEX, EXCPVR counter.
X'20'	IEFVCBLD, lower bit displacement into MCM.
X'22'	IEFVCBHD, upper bit displacement into MCM.
X'24'	IEFVCBLT, lower JES TT.

(Continued)

TABLE 9 Cont'd (Output Only)

X'26'	IEFVCBHT, upper JES TT.
X'28'	IEFVCBTC, tracks per logical cylinder.
X'29'	IEFVCBPC, tracks per physical cylinder.
X'2A'	IEFVCBBT, buffers per track.

Figure 2-54. Work Area Manager Appendage Initialization Processing

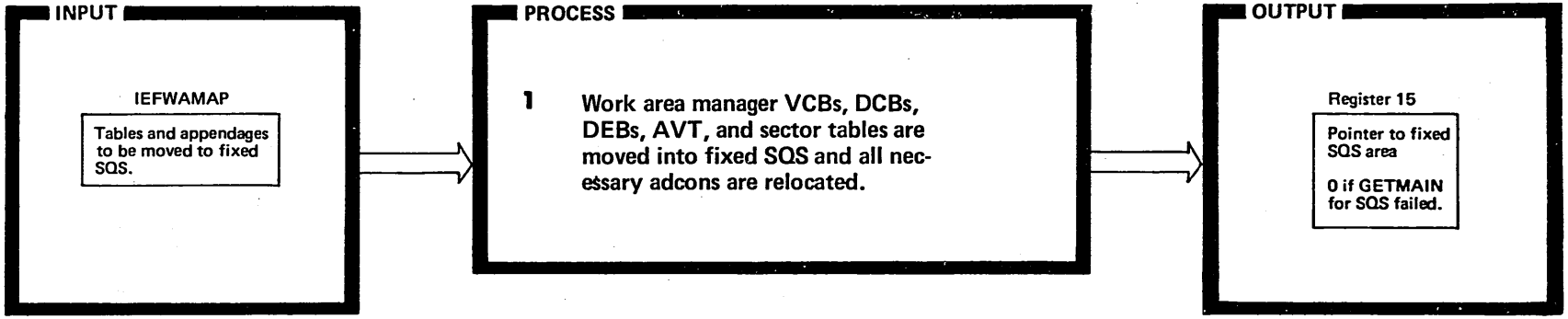


Figure 2-55. Format of SYS1.SYSPOOL

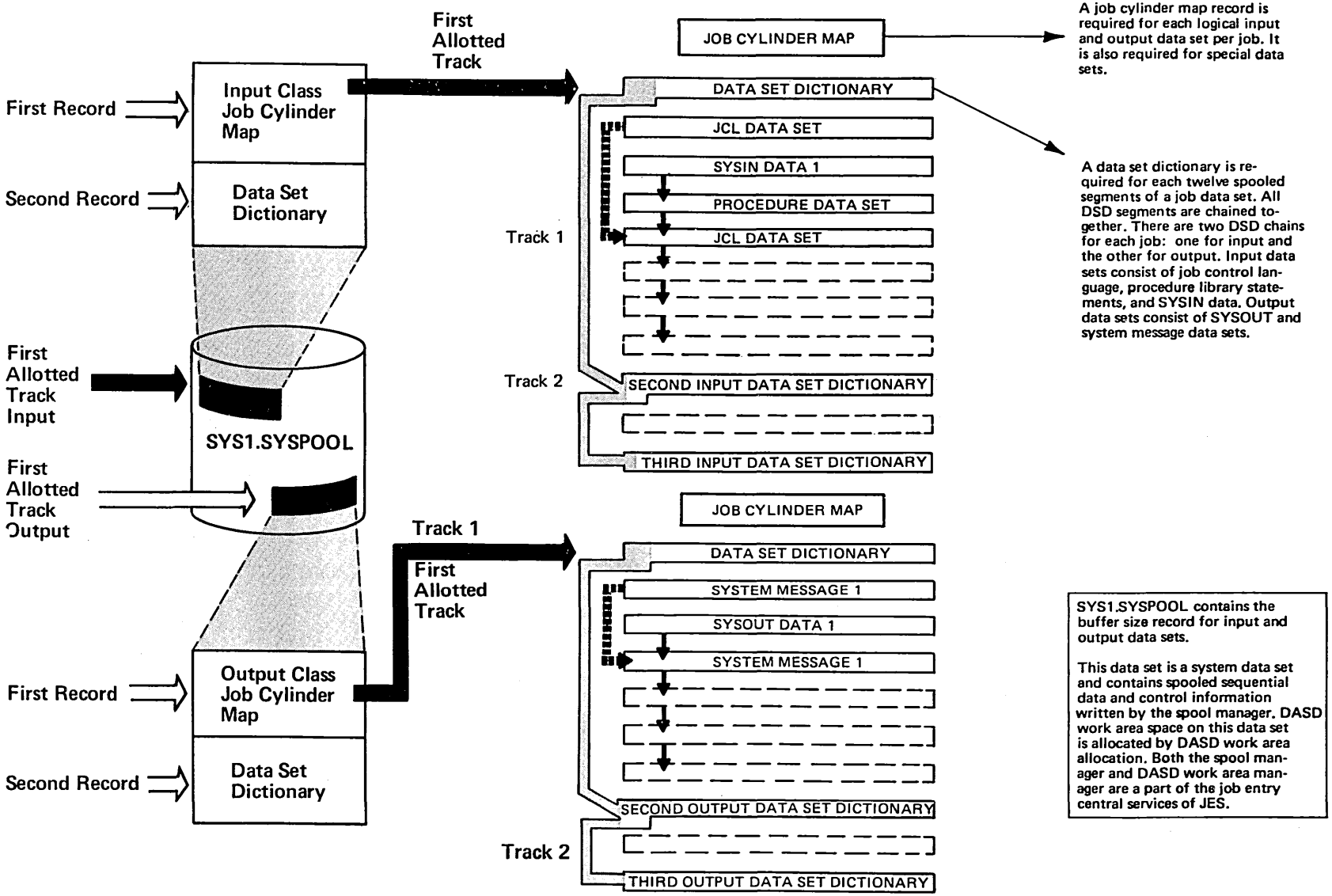


Figure 2-56. Communications Overview

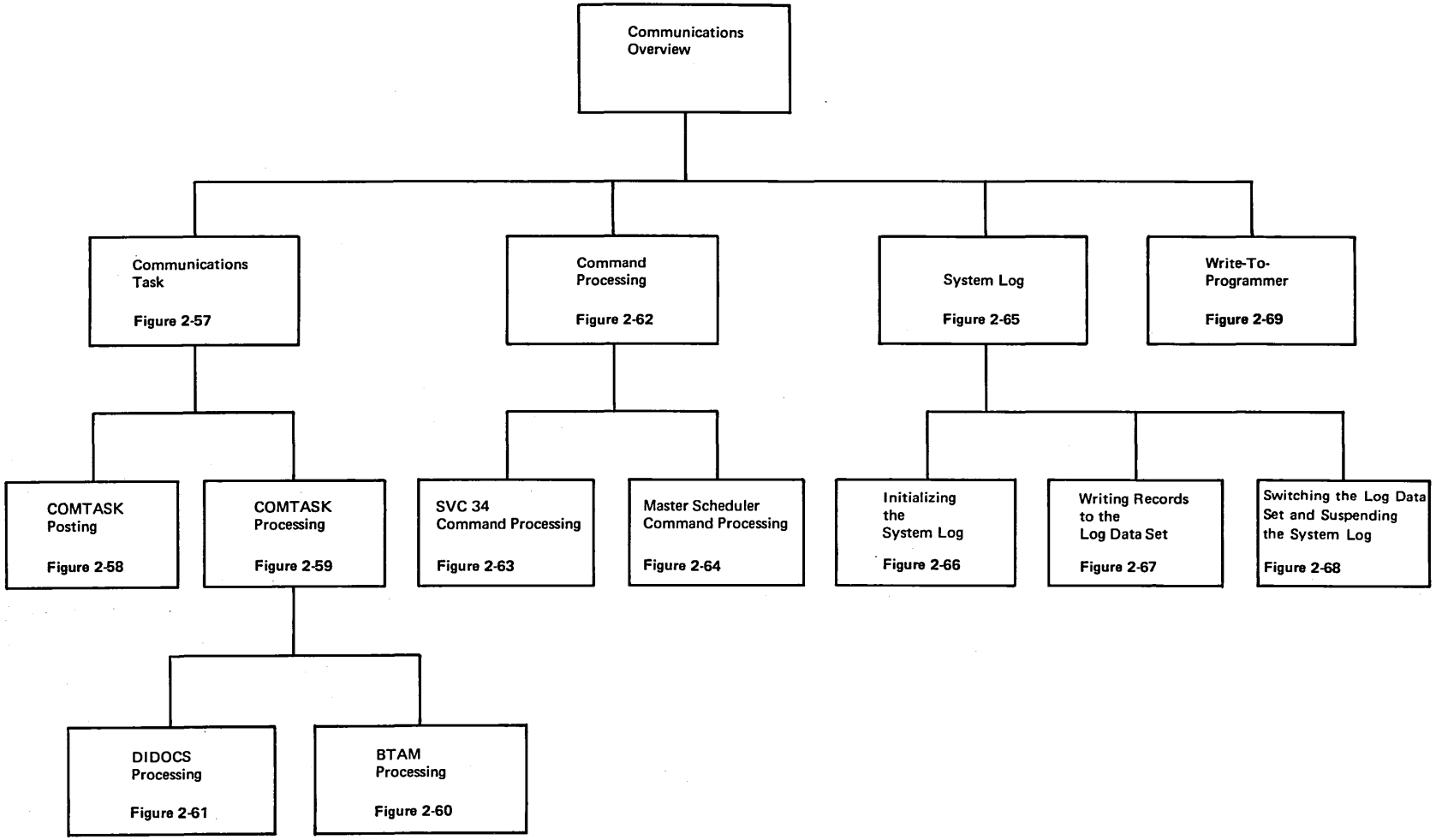


Figure 2-57. COMTASK Overview (Part 1 of 2)

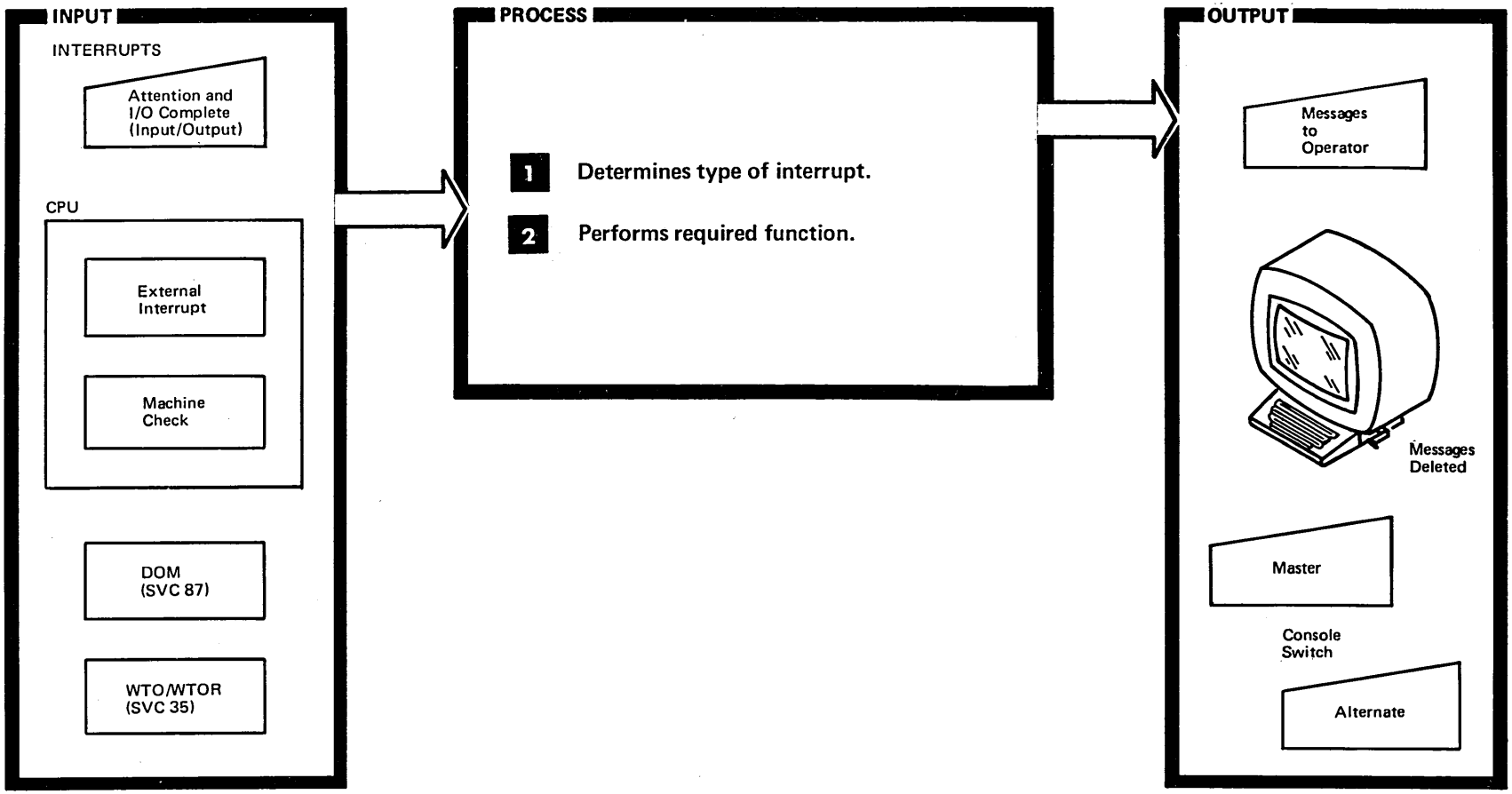


Figure 2-57. (Part 2 of 2)

Extended Description	Module	Figure
1 The Communications Task (COMTASK) handles communications between the operator(s) and the system. The types of communication that COMTASK handles are: • Input to the system in the form of operator commands issued from a console. • Output to the operator caused by the Write-To-Operator (WTO), Write-To-Operator-With-Reply (WTOR), and the Delete-Operator-Message (DOM) macro instructions. • External interrupts, which are caused by the operator pressing the Interrupt key on the operator control panel. This causes a console switch from a primary console to its alternate. • Automatic console switching from a primary console to its alternate when an unrecoverable I/O error occurs on the primary console, or as a result of a VARY command.		

Extended Description	Module	Figure
2 The COMTASK is an interrupt-driven system task. It has its own TCB, created at system generation time. Multiple Console Support (MCS) is an option that supports up to 32 consoles. With MCS, messages can be routed to up to 16 different functional areas, according to the type of information in the message. Device Independent Display Operator Console Support (DIDOCS) is also an option of the control program. It provides uniform operator console services for the: • 2250 Display Unit, Models 1 or 3 • 2260 Display Station, Model 1 with 2848 Display Control or Model 3 • 3158 Display Console in 3215 mode and ANR (3277) mode • 3277 Display Station, Model 2		

Figure 2-58. COMTASK Posting (Part 1 of 2)

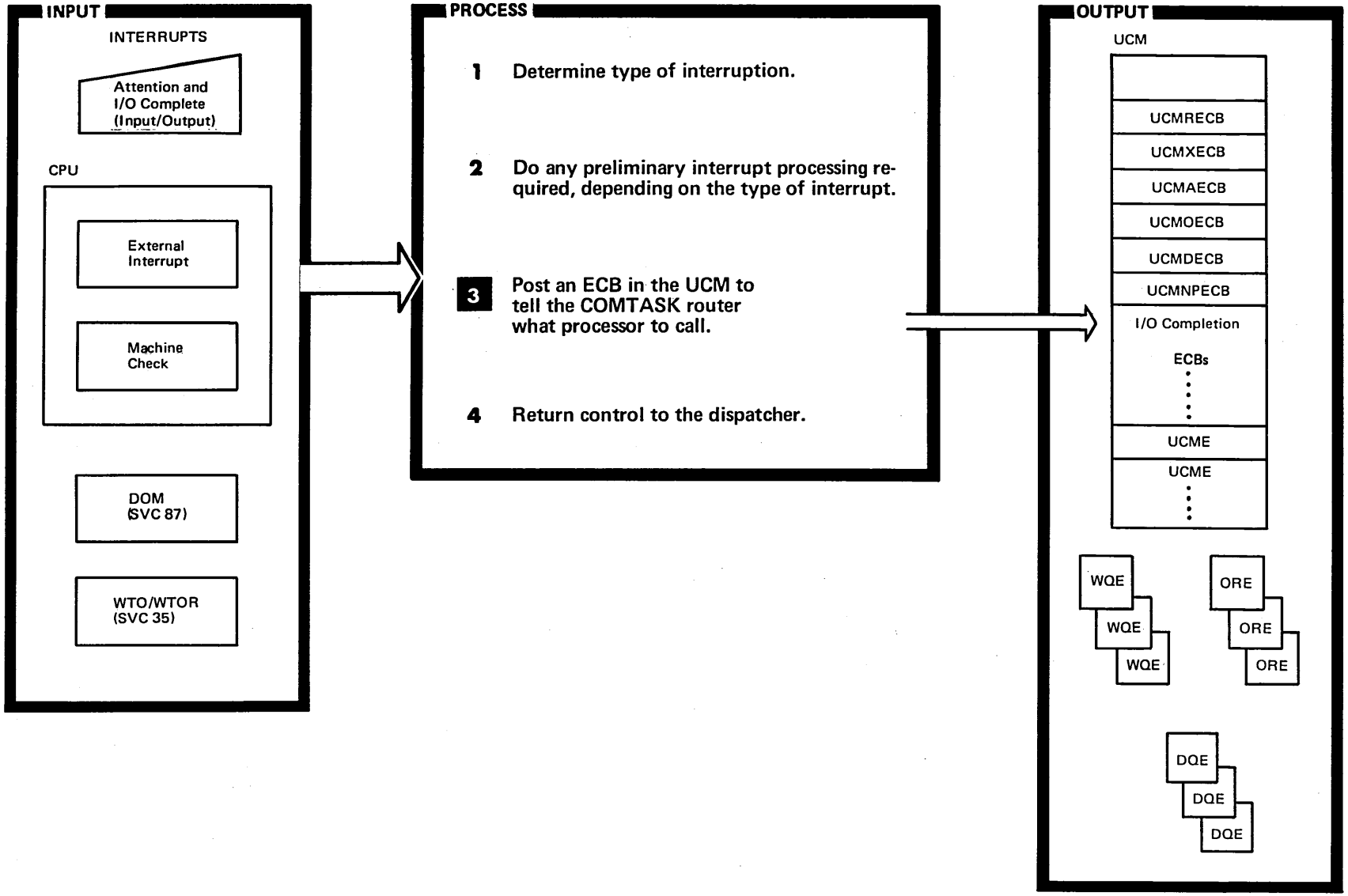


Figure 2-58. (Part 2 of 2)

Extended Description	Module	Figure
3 When an operator presses the correct key (s) on his console, he informs the system that he wants to enter some input from his console. This function is called Attention. It causes an I/O interrupt that informs COMTASK, which console had the Attention. COMTASK, in a module appended to IOS, sets the attention indicator in the UCB for the console and posts UCMAECB. When the input and output operations are complete, an I/O interrupt occurs, causing IOS to post the console's corresponding I/O completion ECB. When the CPU has a detectable error, a machine check interrupt occurs. This causes UCMRECB to be posted. See the RMS logic manual listed in Preface for details.	IEECVCRA	3-113
When the operator presses the Interrupt key on the CPU, it causes an external interrupt. IEECVCRX, a COMTASK module resident in real storage, posts UCMXECB.	IEECVCRX	
WTO and WTOR macros issue SVC 35, causing an SVC interrupt which calls COMTASK transient modules. Output messages are transferred to a Write Queue Element (WQE) and UCMOECB is posted For WTOR, an Operator Reply Element (ORE) is associated with the WQE for the reply. For RES messages, a buffer parameter list (BPL) is built for communication with RTAM. Messages with routing code 11 are passed to the Write-To-Programmer module. (See Figure 2-68.)	IEEMFWTO (MLWTOs) IEECVML3 IEECVML5 IEECVML6 IEECVML7 IEFWTP00	
A DOM macro issues an SVC 87, which builds or adds to a list of display operator messages to be deleted and posts UCMDECB.	IEECXDOM	

Figure 2-59. COMTASK Processing (Part 1 of 2)

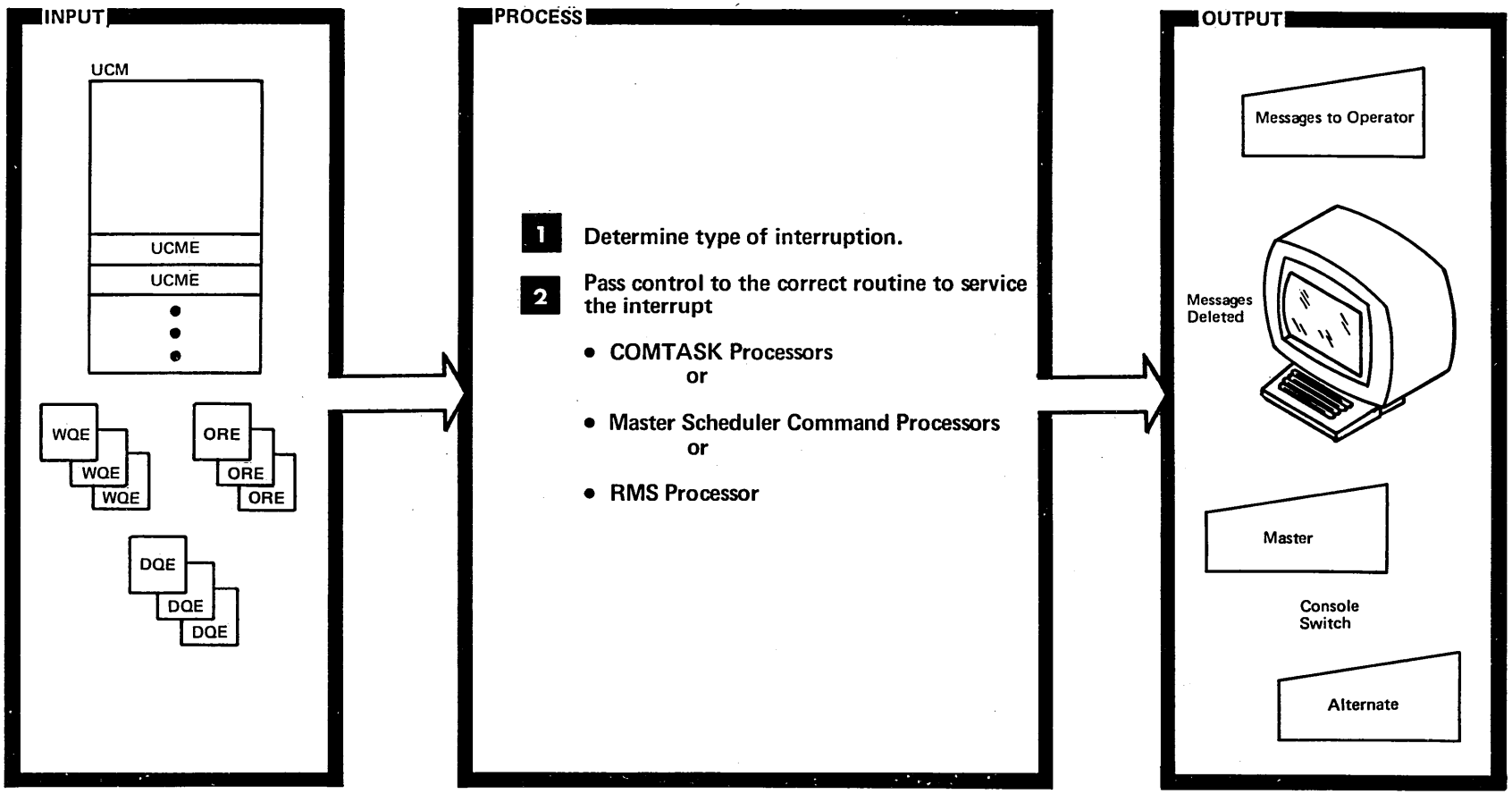


Figure 2-59. (Part 2 of 2)

Extended Description	Module	Figure
1 When an ECB is posted, COMTASK becomes a ready task. It receives control from the dispatcher when it becomes the highest priority task in the system.		3-110
The router module uses the ECBs pointed to by the Event Indication List (EIL), in the Unit Control Module (UCM), to determine the cause of entry. It passes control to a service module. The ECBs are examined by the router in the order: RMS, external, attention, I/O, WTO, and DOM. NIP hardcopy messages are written only once after NIP, if the NIP Hardcopy ECB is posted. Initialization occurs at master scheduler initialization time.	IEECMAWR IEECVCTI	3-112
2 COMTASK uses the following routines to service interruptions.		
The nonexecutable, pageable UCM contains storage areas, pointers, and the event control blocks.	IEEUCMC	
The pageable router receives control when one of the ECBs in the UCM is posted. It issues a WAIT macro instruction specifying a list of addresses (EIL) of the ECBs. It passes control to other COMTASK modules to service the interruptions and to manage communication task queues.	IEECMAWR	
DOM service indicates that specified operator messages can be deleted.	IEECMDOM	3-130
Device service consolidates system queues or passes control through the Mini-Router to appropriate device processor modules. These modules perform read, write, open, and close functions, and any special services required by a particular device (such as a light-pen attention on a 2250). The processor modules are summarized in this chart:	IEECMDSV	

DEVICE	PROCESSOR MODULE	OPEN/CLOSE MODULE
Printer/ Keyboard	IEECMPMX	IEECVOCX
Printer	IEECMPMP	IEECMOCP
Card Reader	IEECMPMC	IEECVOCC
2740 Console	IEEC2740	IEEC2740
DIDOCS devices	IGC5107B	IGC5G07B

Extended Description	Module	Figure
WTO service queues WQEs to console output queues.	IEECMWSV	3-116
The 2740 processor performs OPEN and CLOSE, READ and WRITE, using BTAM modules.	IEEC2740	
The Mini-Router receives control as a result of an SVC 72 issued by IEECMAWR, IEECMDOM, IEECMDSV, or IEEC2740. This SVC 72 causes an SVRB to be built for the execution of the device processors and other transient COMTASK modules.	IEECMCTR	
Console-switch routines switch consoles as a result of an external interruption, an unrecoverable I/O error, or a VARY command.	IEECLCTX	3-114
The NIP message buffer writer writes messages from the buffer created by the Nucleus Initialization Program.	IEECMWTL	

Figure 2-60. BTAM 2740 Console Processing (Part 1 of 2)

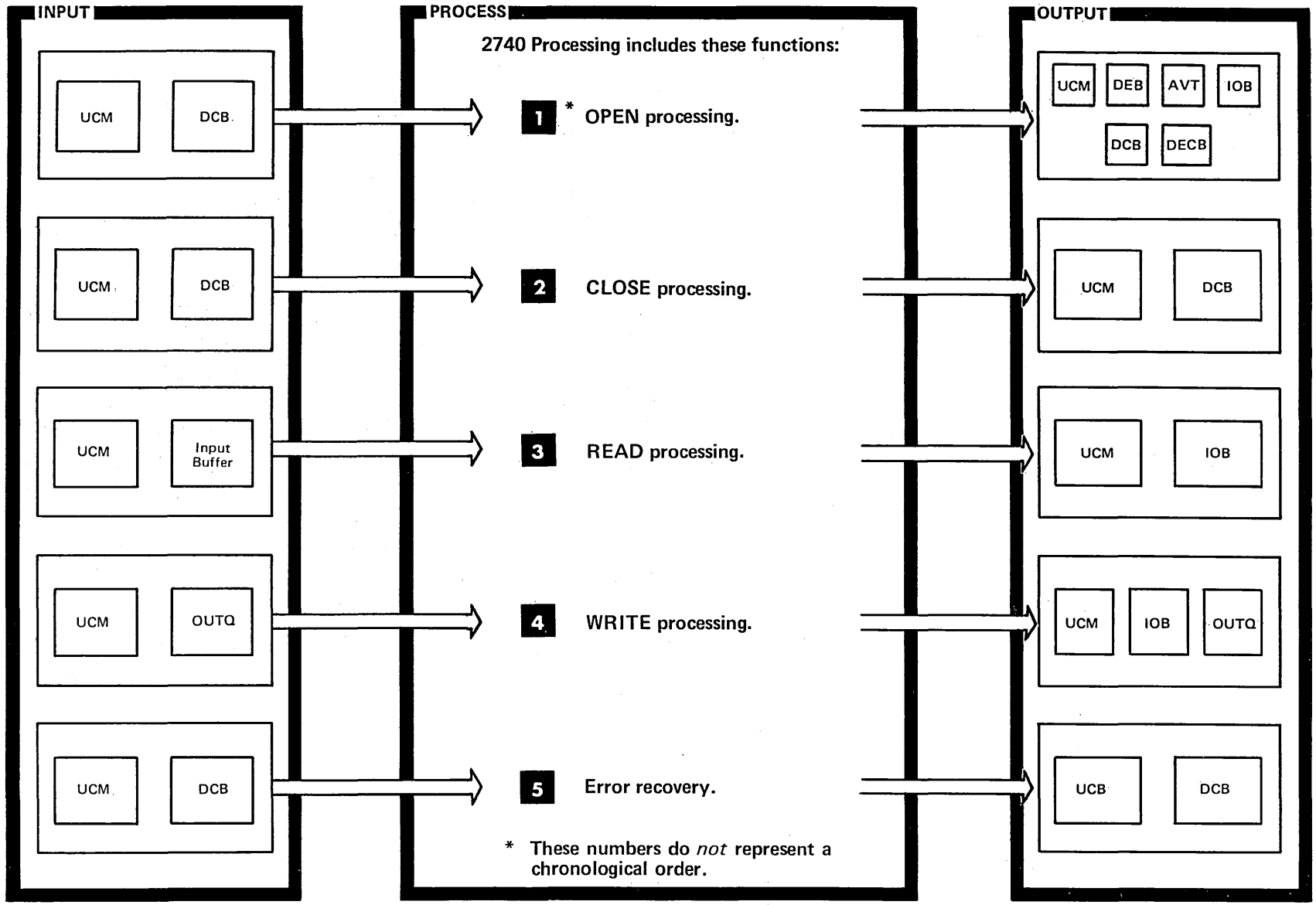


Figure 2-60. (Part 2 of 2)

Extended Description	Module	Figure
1 OPEN is performed when the 2740 Device Support Processor gains control with OPEN indicated in the console's entry in the UCM as a result of a VARY CONSOLE command. At this time IEEC2740 places the console's ECB in the console ECB list, chains the DEB to the COMTASK TCB, obtains buffers, initializes the IRB and IQE for OLT, initializes the AVT, and finds the addresses of the BTAM modules via LOAD macro instructions. The line to the 2740 is initialized, and, if initialization is successful, the UCM entry is set to indicate that the 2740 console is open and active. When the channel program completes, a WRITE operation starts if an output message is available. Otherwise, a READ operation starts.	IEEC2740	
2 CLOSE is performed when the UCM entry indicates CLOSE pending, as a result of a VARY OFFLINE command or a console switch, and when activity at the 2740 console has quiesced. IEEC2740 removes the ECB from the ECB list, and the DEB from the DEB chain of the TCB; it deallocates the UCB, and sets the UCM entry to indicate that the console is closed and inactive. The BTAM modules are deleted.	IEEC2740	

Extended Description	Module	Figure
3 A READ operation is initiated when a previous I/O operation completes and there is nothing on the output message queue to be written. A READ operation can be forced, even with active output queue entries present, if the previous I/O operation was a WRITE during which the 2740 console operator pressed the BID key. When a READ completes successfully, the message received is sent to the master scheduler command processors via SVC 34.	IGG019MA	
4 A WRITE operation is initiated when a previous I/O operation completes, and an active entry exists on the output message queue.	IGG019MA	
5 If an I/O operation cannot be started, or if an unrecoverable line error occurs, the console switch routine (SVC 72) assigns the failing console's functions (and unwritten messages) to its alternate. After SVC 72 returns control, the failing console is closed.		

Figure 2-61. DIDOCS Processing (Part 1 of 2)

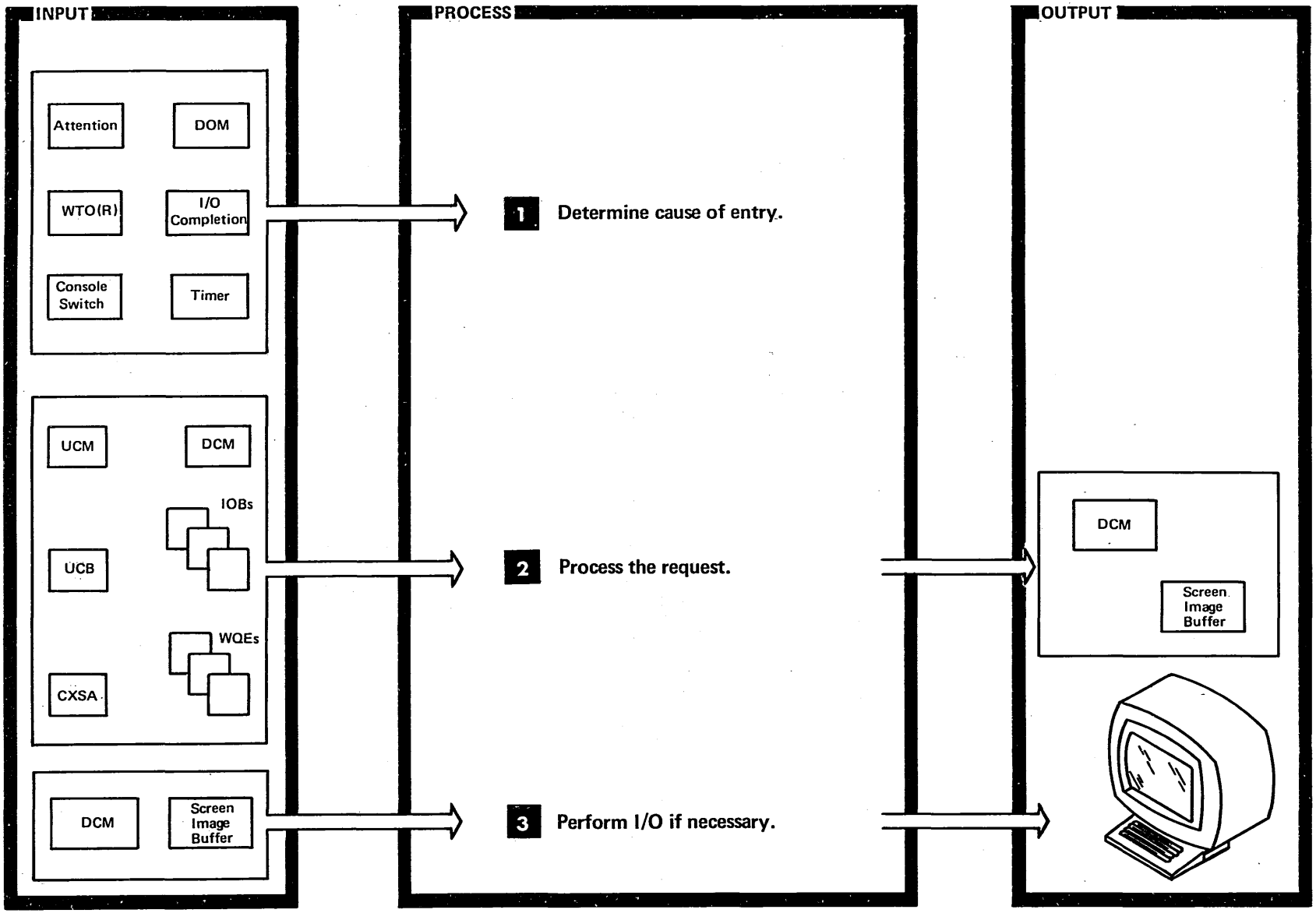
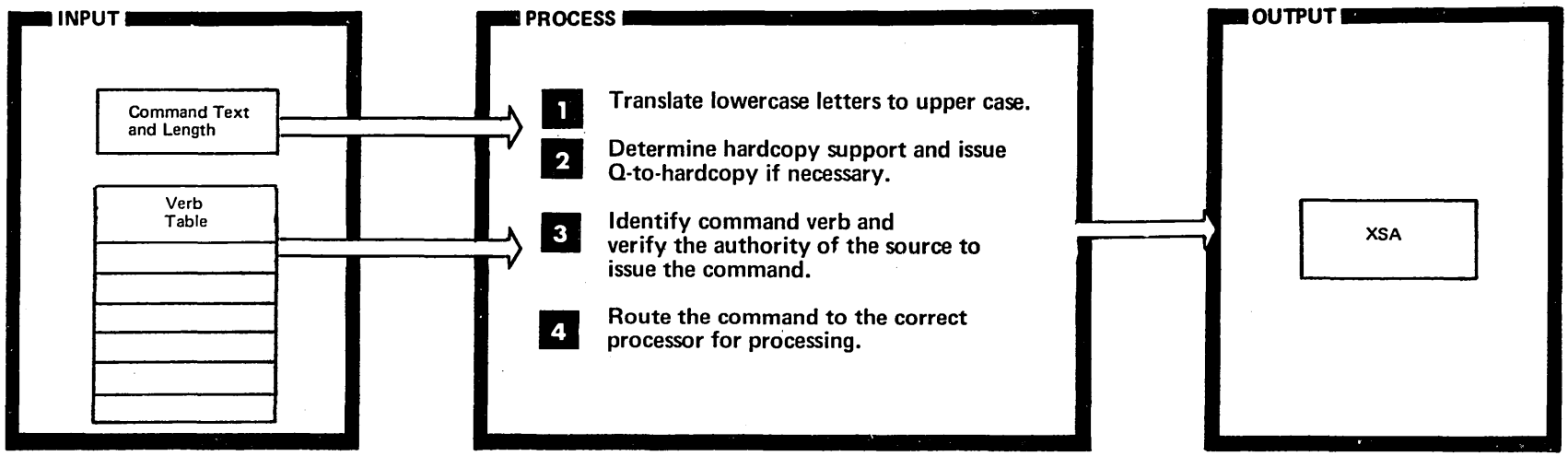


Figure 2-61. (Part 2 of 2)

Extended Description	Module	Figure
1 Device Independent Display Operator Console Support (DIDOCS) takes advantage of device-dependent features of each display device, such as the use of the light pen on the 2250 for message deletion. The DIDOCS processor receives control from MCS when one of these events occurs for a display console:	IGC5107B	
• An attention caused by the operator.		3-125
• Console switching		
• I/O completion		
• A WTO, WTOR, or command		
• A Delete Operator Messages (DOM) request		3-130
• Messages to be displayed		
• A timer interruption		
• Status display on the queue		
2 The DIDOCS support routines provide these functions:		
• Interface between MCS and support routines	IGC5107B	
• Open/Close for display console devices	IGC5G07B	3-122
• Device I/O processing	IGC5P07B IGC5Q07B IGC5R07B	

Extended Description	Module	Figure
• Asynchronous error processing	IGC5C07B	3-124
• Message display	IGC5207B IGC5307B	3-126
• Message deletion	IGC5A07B IGC5607B IGC5707B IGC5807B IGC5907B	
• Roll mode	IGC5J07B	3-123
• Command processing	IGC5407B	
• Light Pen/Cursor support	IGC5F07B	
• CONTROL command display	IGC5N07B IGC5O07B IGC5S07B	3-128
• Timer	IGC5K07B	3-123
3 Device I/O routines handle input/output operations for the:		
• 2250 Display Unit, Models 1 and 3	IGC5P07B IGC5Q07B	3-131
• 2260 Display Station, Model 1 with 2848 Display Control, or Model 3		
These routines build and execute the CCWs needed for reading manual input or input from the entry area, writing and erasing the screen, inserting the cursor, and sounding the alarm.		

Figure 2-62. Command Processing Overview (Part 1 of 2)



Extended Description	Module	Figure
1 Control passes to the command scheduling routines via SVC 34. When control is passed, register 1 points to the command input buffer (CIB) and register 0 identifies the source of the command. Lowercase letters enclosed in apostrophes are not translated.	IEE0403D	
2 In systems with MCS, if the command is added to the hardcopy log, the router issues a WTO with the MCSFLAG=HRDCPY parameter and updates the Extended Save Area (XSA) with the contents of register 0, as follows.	IEE0403D	

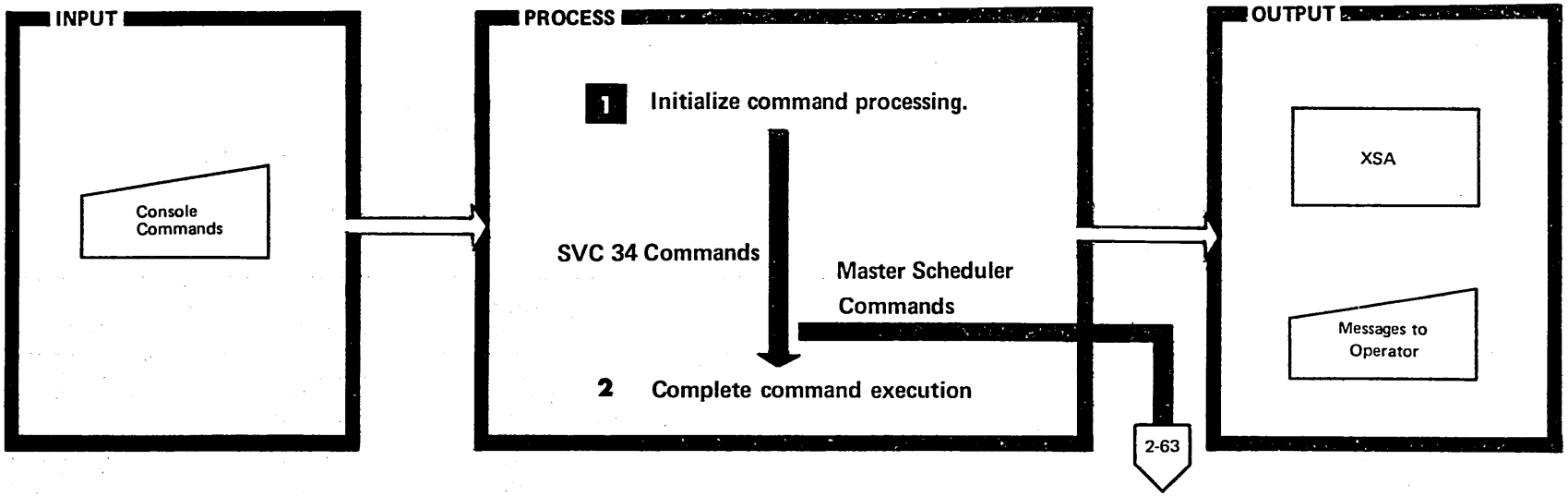
Register 0	Value Stored	XSA Field
positive	contents of register 0	UCMI
negative	X'80' and contents of register 0	UCMI MIWA

Extended Description	Module	Figure
3 The router routine uses the command verb to find the corresponding verb table entry. If there is no entry, an error message is issued. The verb table contains a list of all the command verbs, command abbreviations, their codes, and their command authority. If a command is issued from an unauthorized source, an error message is issued.	IEE0403D	
4 If the command is valid, the SVC 34 router then passes control to the appropriate processor to perform the function (see Part 2 of this figure).	IEE0403D	

Figure 2-62. (Part 2 of 2)

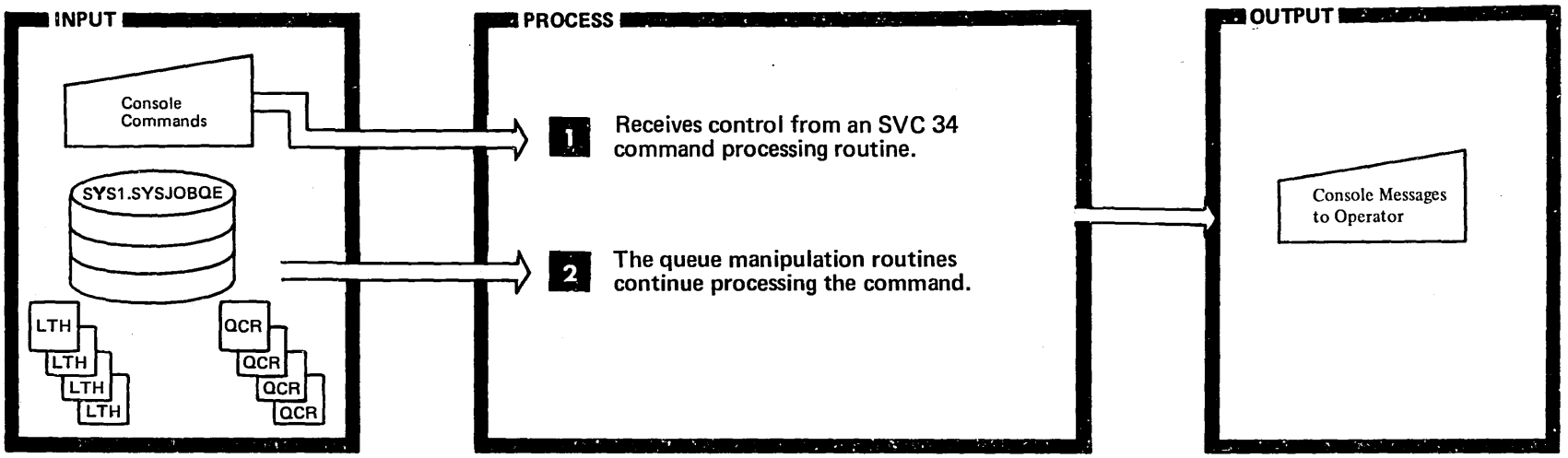
Command	SVC 34 Command Processor Module	Functional Flow Figure Number
CANCEL	IEE3703D	3-58, 3-59
CONTROL	IEE7503D	3-60
DEFINE	IEESD571	3-61
DISPLAY	IEE3503D	3-62
DISPLAY RT	IEE8803D	3-63
DISPLAY U	IEE20110	3-64
DUMP	IEE60110	3-67
HALT	IEE1403D	3-68
HOLD	IEE3505D	3-69
LISTBC	IEELIST	3-70
LOG	IEE1603D	3-71
LOGOFF	IEE8703D	3-72
LOGON	IEELGON	3-73
MODE	IGF2603D	3-74
MODIFY	IEE4503D	3-75
MONITOR	IEE7103D	3-76
MOUNT	IEE1903D	3-77
MSGRT	IEE6303D	3-78
RELEASE	IEE3503D	3-79
REPLY	IEE1A03D	3-80
RESET	IEE3503D	3-81
ROUTE	IEERTE	3-82
SEND	IEEVSEND	3-83, 3-84, 3-85, 3-86, 3-87
SET	IEE0603D	3-88
START	IEE1903D	3-89
STOP	IEE4503D	3-90
STOPMN	IEE4503D	3-91
SWAP	IGF2503D	3-92, 3-93
SWITCH	IEE1403D	3-94
UNLOAD	IEE1103D	3-95
VARY	IEE3203D	3-96, 3-97, 3-98, 3-99, 3-100, 3-101
WRITELOG	IEE1603D	3-102, 3-103
WRITER	IEE9903D	3-104
Message Module	IEE0503D	3-105
Queue Alter Routines		3-106

Figure 2-63. SVC 34 Command Processing



Extended Description	Module	Figure
1 The SVC 34 processor first determines why its services were requested by examining the contents of Registers 0 and 1. If register 1 is greater than zero, it was invoked to process a command; if zero, it was entered for CIB or CSCB manipulation. Each command is passed to the correct processor. If it is a command to be processed by the master scheduler, the processor builds and chains a Command Scheduling Control Block (CSCB) and passes control to the master-scheduler routines. A CSCB is also created for the MOUNT and START commands.		

Figure 2-64. Master Scheduler Command Processing



Extended Description	Module	Figure
1 The operator commands processed by the master scheduler are:		
CANCEL (a job on a queue)	IEE3703D	3-57
DEFINE	IEESD571	3-61
DISPLAY	IEE3503D	3-62
DISPLAY ACTIVE	IEESD566	3-62
HALT	IEE1403D	3-68
HOLD	IEE3503D	3-69
MONITOR ACTIVE	IEESD566	3-76
RELEASE	IEE3503D	3-79
RESET	IEE3503D	3-81
SWITCH	IEE1403D	3-94
The command processing routine builds and chains a CSCB containing the command image.		

Extended Description	Module	Figure
2 The queue alter routines complete the command processing as follows:		3-106
The queue command processor receives control from the Master Scheduler Resident Wait/Router module (IEECIR51).	IEESD561	
The command syntax is checked.	IEESD562	
The validity of the unit is checked, if necessary.	IGCU103D	
The queues are searched for the job by reading the QCRs and the LTHs.	IEESD564 IEESD563 IEESD565	
The job is dequeued and enqueued as necessary, or canceled.	IEESD575 IEESD576	
For DEFINE commands, IEESD571 sets a bit in the MSRDA and passes control to the DEFINE command processors.	IEESD571	3-61

Figure 2-65. System Log Overview

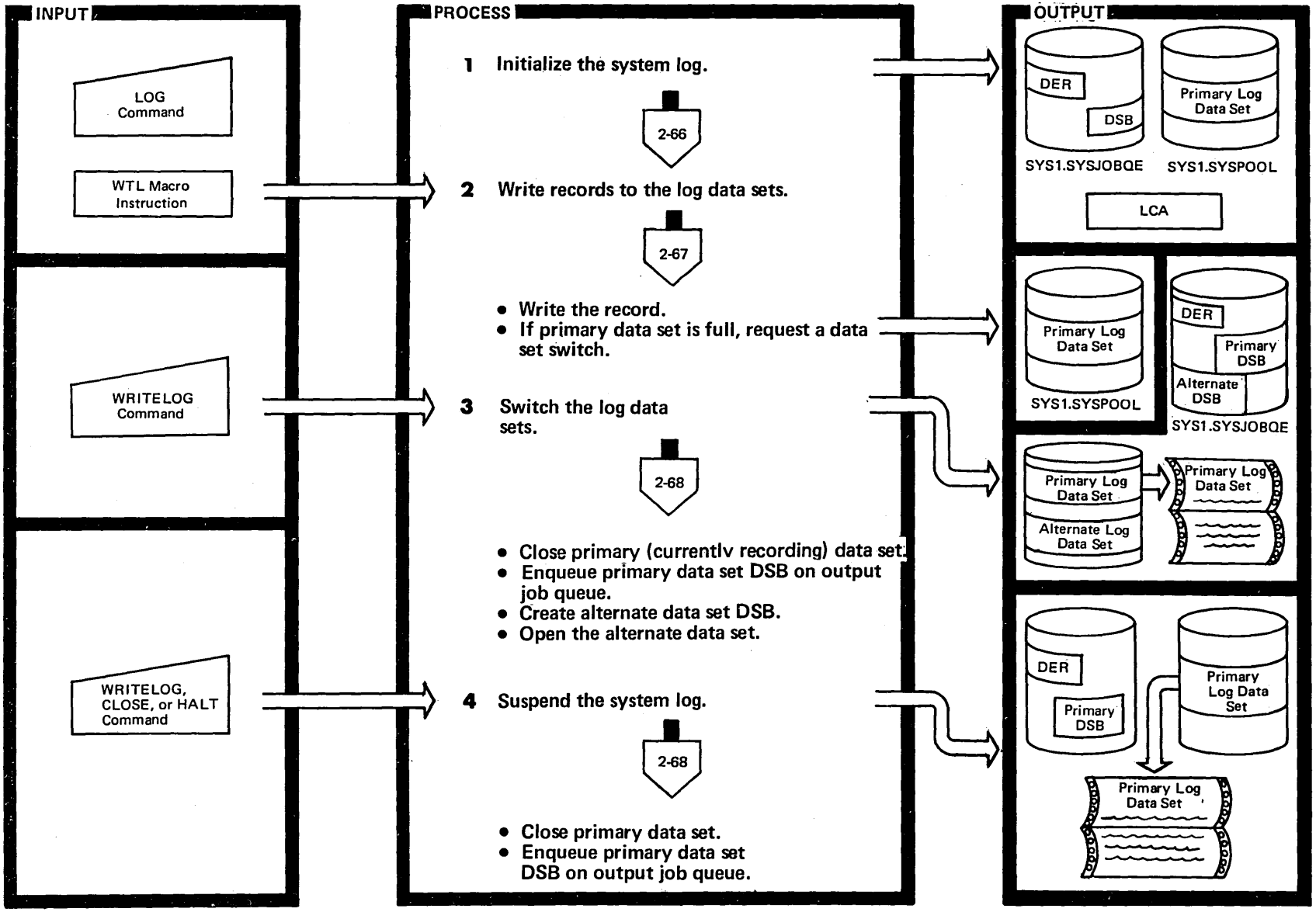


Figure 2-66. Initializing the System Log (Part 1 of 3)

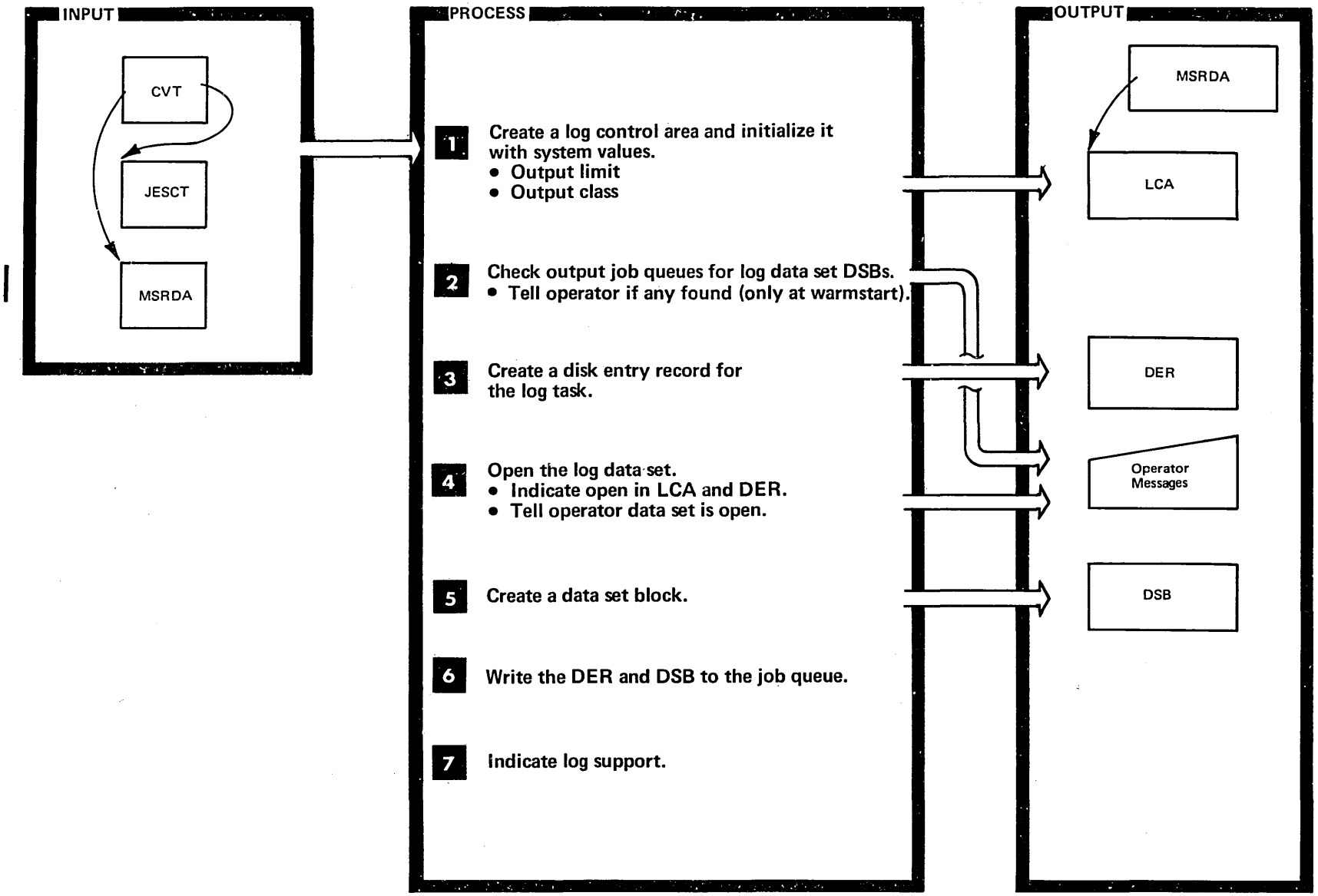


Figure 2-66. (Part 2 of 3)

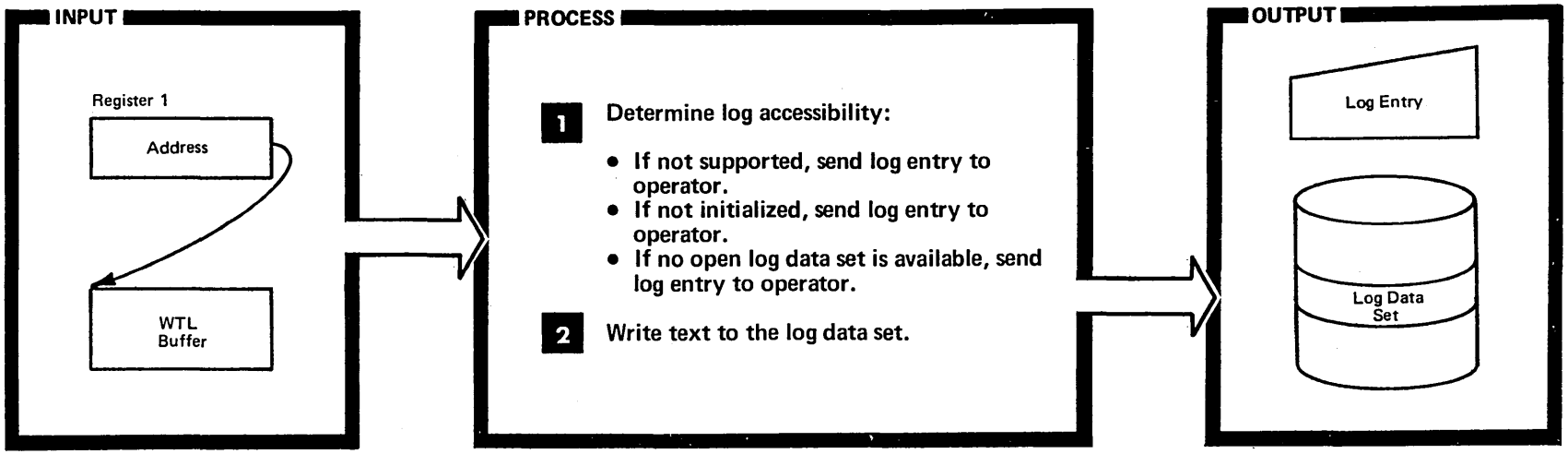
Extended Description	Module	Figure
1 The log initialization routine copies the SYSGEN default value for the system output class and the output limit values for the log data sets into the Log Control Area (LCA).	IEEVLIN	3-107
2 At warmstart IPL, the routine initializes the LOCDSECT parameter area and issues the queue management macro instruction to request a search of the output job queues for one or both of the log data sets DSBs. The return information indicates the number and identity of any data set DSB that is enqueued on an output job queue. The routine then tells the operator the SYSOUT class of any enqueued DSB, so that he may start a writer to the appropriate class to retrieve the contents of the log data set(s). The routine notifies the operator in the first three of the following conditions: • Both data set DSBs are on the same output job queue. • Each data set DSB is on a different output job queue. • Only one data set DSB is on an output job queue. • Neither data set DSB is on an output job queue.		
3 When the system log is being initialized, the Disk Entry Record (DER) that represents the system log task is created. The routine first initializes a Queue Manager Parameter Area (QMPA) and issues the queue management macro instruction to request space on the job queue for a 176-byte record for the DER. The TTR of the DER is returned in the external parameter area (EXTPARM) and the routine copies it into fields of the LCA and the DSB (when it is created). The routine then initializes fields in the DER and sets the DERLOGID bit to 1 to indicate that this DER represents the system log task.	IEE0403D	

Extended Description	Module	Figure
4 When the system log is being initialized, obtain system queue space for the open request parameter list (IEFODRPL) and the Logical Record Control Block (LRCB) before the open data set request can be made to Job Entry Central Services (JECS). The routine saves the address of these areas in the LCA so that they are available for subsequent open requests. The routine initializes the open RPL and issues the JECS macro instruction to request the opening of a log data set. Information is returned in the open RPL. It consists of error data, the TTR of the Data Set Dictionary (DSD) and the Job Control Map (JCM). The routine copies these addresses into the DER and the DSB (when it is created). The routine also initializes an RPL for the JECS close open dictionary function and issues the JECS macro instruction requesting that the DSD and the JCM be written to the SYS1.SYSPOOL data set. The routine then checks the error data fields of the RPL to determine if the OPEN was successful. If not, it reinitializes the OPEN RPL and requests the opening of the alternate data set. If this OPEN request also fails, the routine informs the operator that the log function is inactive and returns control to the issuer of the SVC 36 via an EXIT SVC. The routine indicates that the log data set has been successfully opened by setting bits for the opened data set (either data set x or data set y) in the LCA and the DER and informing the operator that a log data set is ready to receive information.		
5 The routine creates a Data Set Block (DSB) for the data set just opened by first requesting space for a record on the job queue via the queue manager macro instruction. When the TTR of the DSB is returned in the EXTPARM area, the routine copies it into the fields of the DER and the DSB. It then initializes fields in the DSB and gets the first DSBLGOUT bit to 1 to indicate that the DSB represents a system log task data set.		

Figure 2-66. (Part 3 of 3)

Extended Description	Module	Figure
6 The routine then requests that both the DSB and the DER be written on the job queue. It initializes a QMPA and issues the queue management macro instruction to request the writing of the DSB. It then does the same for the DER.		
7 The routine indicates that the system log is initialized by storing the address of the LCA in the Master Scheduler Resident Data Area (MSRDA).	IEEVLIN	

Figure 2-67. Writing Records to the Log Data Set



Extended Description	Module	Figure
1 When the LOG and WRITELOG command handler routine determines that the LOG command is valid, it requests the writing of the log entry to the log data set. Three unfavorable conditions may exist when control reaches this routine: • A bit in the master scheduler resident data area set by IEE0403F may indicate no system support for the log function. • IEEVLIN zeroes the LCA pointer in the master scheduler resident data area if log initialization has not occurred. • IEE0403F manipulates the LCA open bits for LOG data sets as it uses them; if no open data set is available for the WTL, the LCA open bits indicate that fact.	IEE1603D IEE0303F	3-71 3-102 3-103 3-108

Extended Description	Module	Figure
In each of these cases, the routine writes the log entry to the console with an appropriate message ID. **2** If the LOG data set is operative and available, the routine schedules the proper I/O via JECS using the IEFSMREQ macro. The routine then tests the information returned in the RPL and posts the log ECB requesting a data set switch if either of the following conditions exists: • An uncorrectable I/O error has occurred. • The output limit for the data set has been reached (that is, the data set is full).		

Figure 2-68. Switching the Log Data Set and Suspending the System Log (Part 1 of 2)

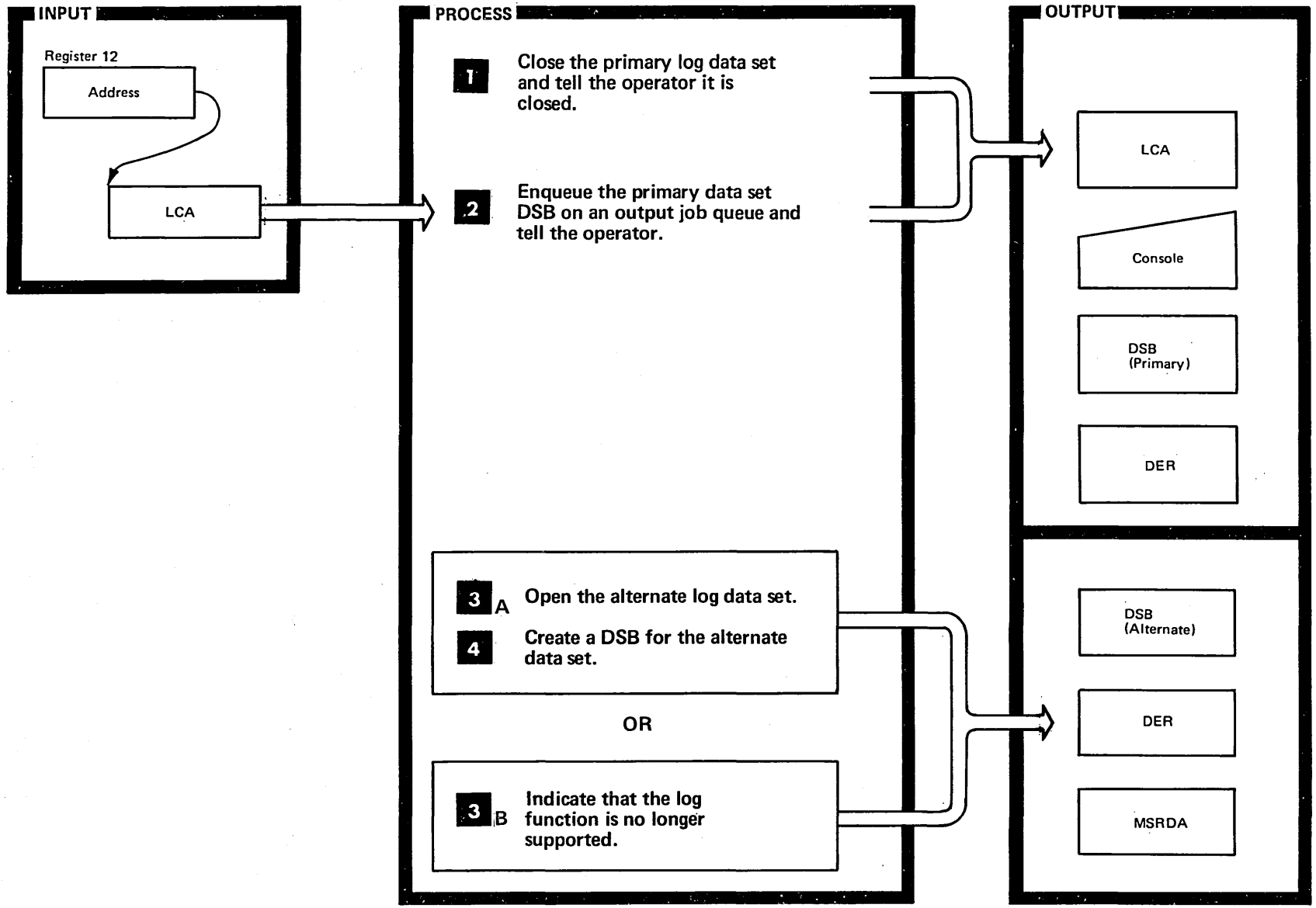
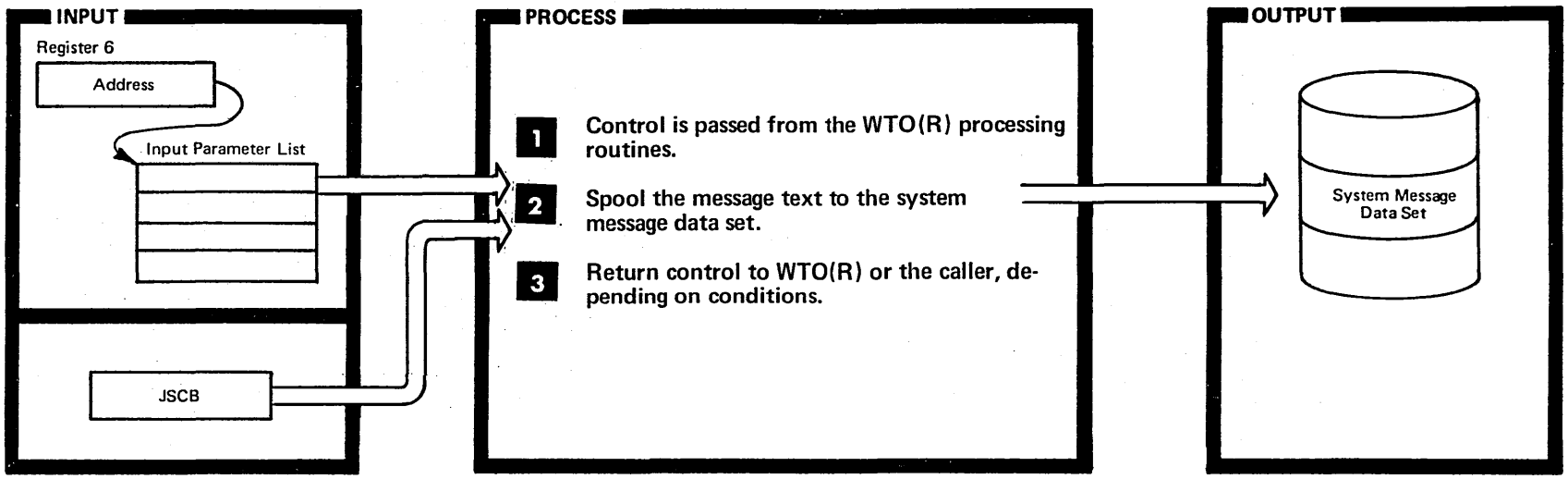


Figure 2-68. (Part 2 of 2)

Extended Description	Module	Figure
1 The routine initializes a QMPA and issues the queue management macro instruction to request the reading of the log Disk Entry Record (DER) into storage. It tests the bits in the DERLOG field to determine the status of the log data sets. If the DSBs for both log data sets are enqueued on an output job queue, the routine tells the operator that the log function is temporarily inactive, and returns control to the issuer of the SVC 36. If a log data set is open, it is the currently recording (primary) data set and must be closed. The routine therefore successively initializes the RPLs for the JECS end data set and close open dictionary functions and issues JECS macro instructions requesting that the primary data set be closed. The routine then indicates that the primary log data set has been successfully closed by setting bits in the DER and the LCA and informing the operator. (The primary data set could be either data set x or data set y.)	IEE0403F	3-103
2 The routine initializes a QMPA and issues the queue management macro instruction to request the reading of the Data Set Block (DSB) for the primary data set into storage. It then copies either the output class specified by the operator on the WRITELOG command or the SYSGEN default for the output class from the LCA to the DSB. It then initializes a QMPA and issues the queue management macro instruction to request that the DSB be written and enqueued to an output job queue. It indicates that the primary data set DSB has been enqueued by setting bits in the LCA and DER and informing the operator.		
3A The routine then opens the alternate log data set in the same manner as that described for opening a log data set in step 4 of Figure 2-66. The only difference is that if the open request fails, the routine returns control to the issuer of the SVC 36.		
4 The routine then creates a DSB for the alternate log data set in the same manner as that described in step 5 of Figure 2-66.		
3B If the command was WRITELOG CLOSE or HALT EOD, the routine indicates that the system log has been eliminated by setting the "log not supported" bit in the MSRDA and zeroing out the address of the LCA. In either case (3A or 3B), the DER is rewritten to SYS1.SYSJOBQE.		

Figure 2-69. Write-To-Programmer



Extended Description	Module	Figure
1 The Write-To-Programmer (WTP) routine receives control whenever a WTO or WTOR command has a routing code of 11. **2** The WTP routine checks for a previous I/O error and for the user limit. In the case of an I/O error, a message to the operator is issued during processing, once per job step. Any other WTP requests in the job step are ignored. If the user limit for the number of messages in the system message data set has been reached, an error message is written to the operator and to the system message data set. All additional WTP requests in that job step are ignored. An additional 15 system WTP messages can still be written to the data set.	IEFWTP00	3-109

Extended Description	Module	Figure
The JSCBWTP field of the JSCB contains all WTP control information. If multiprogramming is in the partition when the WTP routine receives control, this field is enqueued upon to ensure sequential use of the field. **3** For WTOR messages, control returns to the WTOR routines. If there are additional routing codes on a WTO message, or if MCS is in the system and there is a console that accepts routing code 11, control returns to the WTO routines. Otherwise, control returns to the caller via BR 14.		

Figure 2-70. System Management Facilities

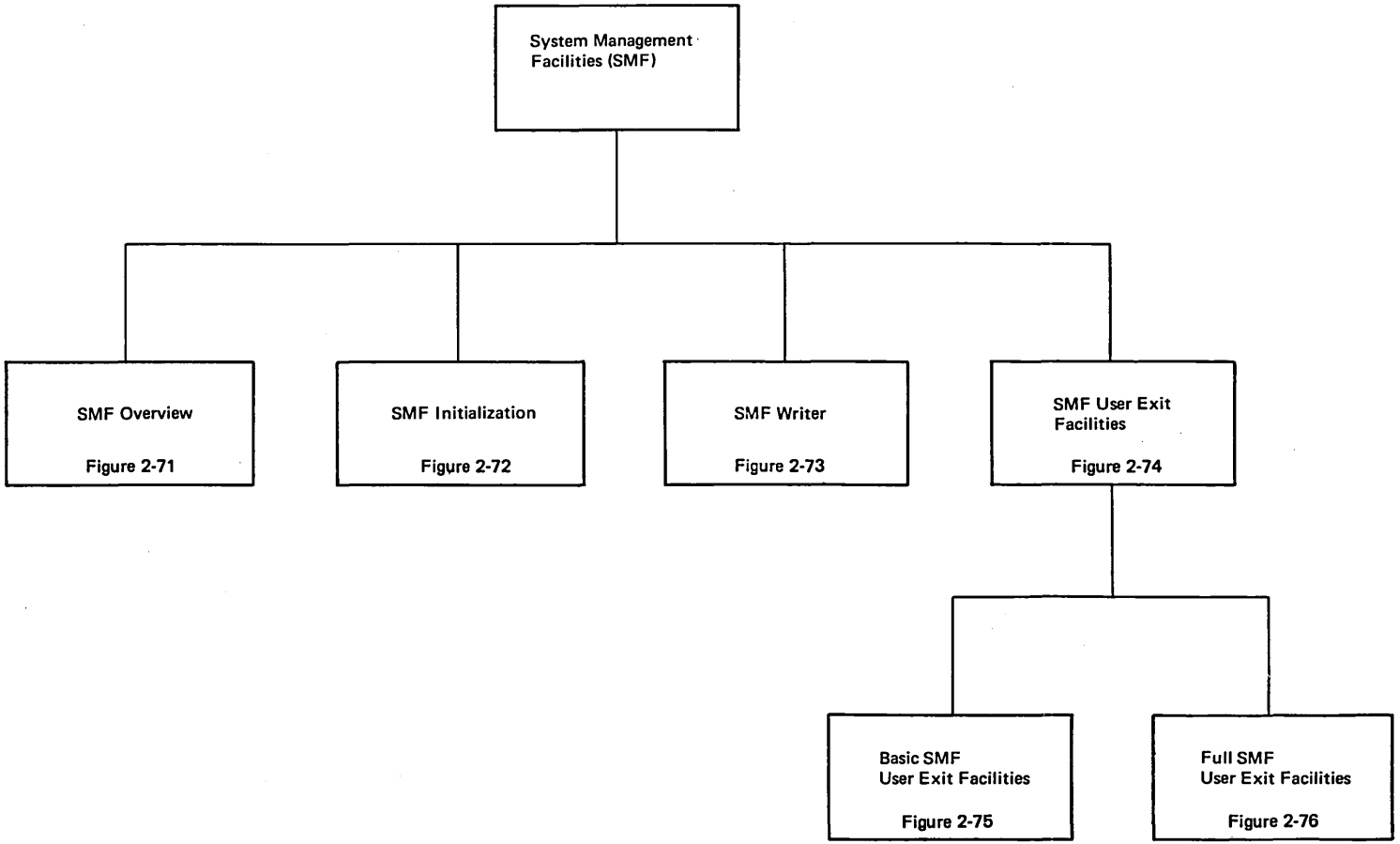


Figure 2-71. SMF Overview (Part 1 of 2)

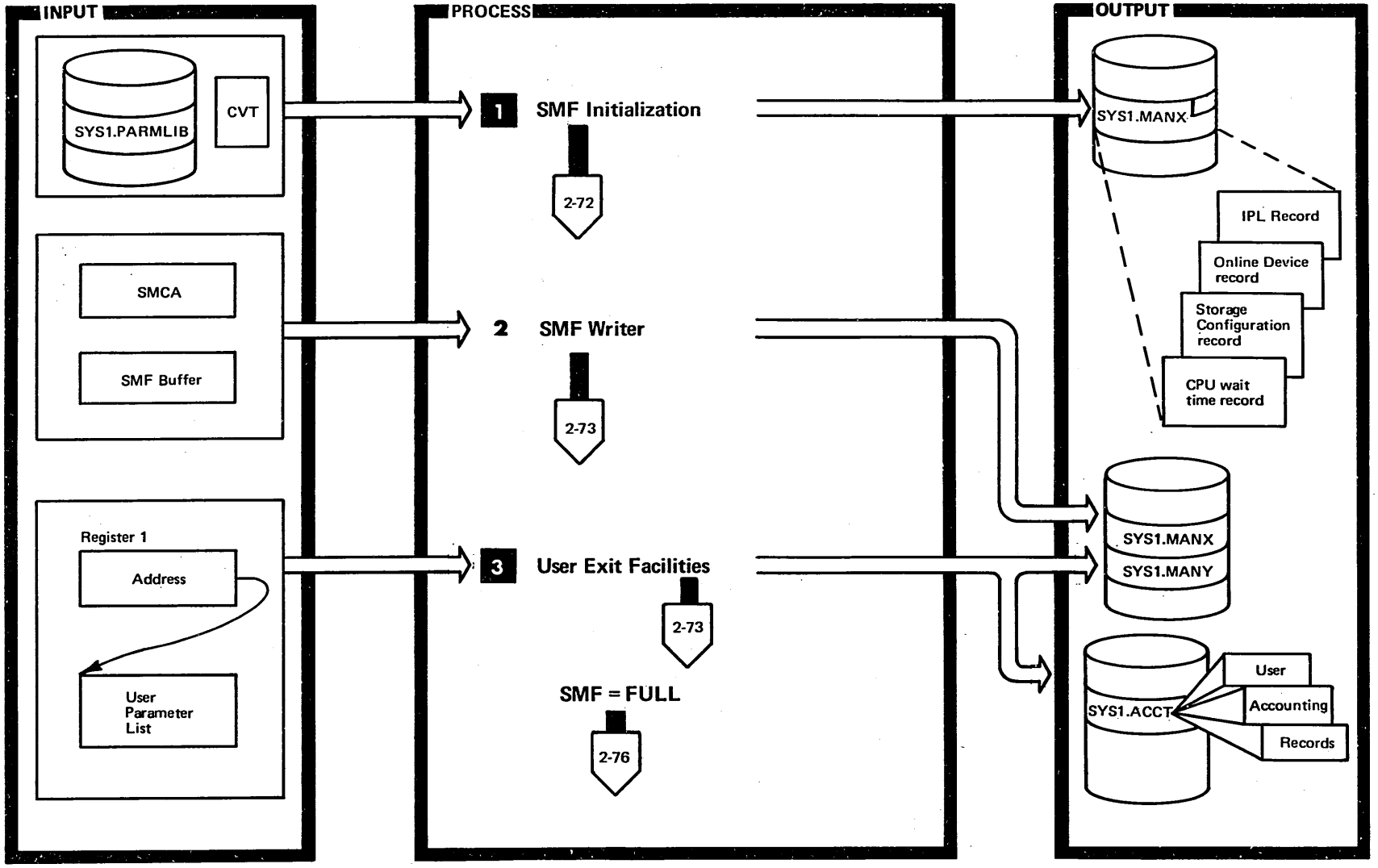


Figure 2-71. (Part 2 of 2)

Extended Description	Module	Figure
1 The System Management Facilities (SMF) consist of routines that collect data, exits for user-supplied data-collection routines, and routines that record the collected data in a data set. SMF uses two direct access data sets (SYS1.MANX and SYS1.MANY).		3-133
SMF initialization occurs as part of master scheduler initialization.		3-56
3 SMF data collection routines, and exits for user-supplied data collection routines are in the interpreter, in the initiator, in JEPS, and in the termination routine. SMF records are issued in the interpreter, initiator, terminator, System Output Writer and in the VARY command processor. Other data-collection facilities are provided in the supervisor (see the OS/VS1 Supervisor Logic manual listed in the Preface).		

Figure 2-72. SMF Initialization (Part 1 of 2)

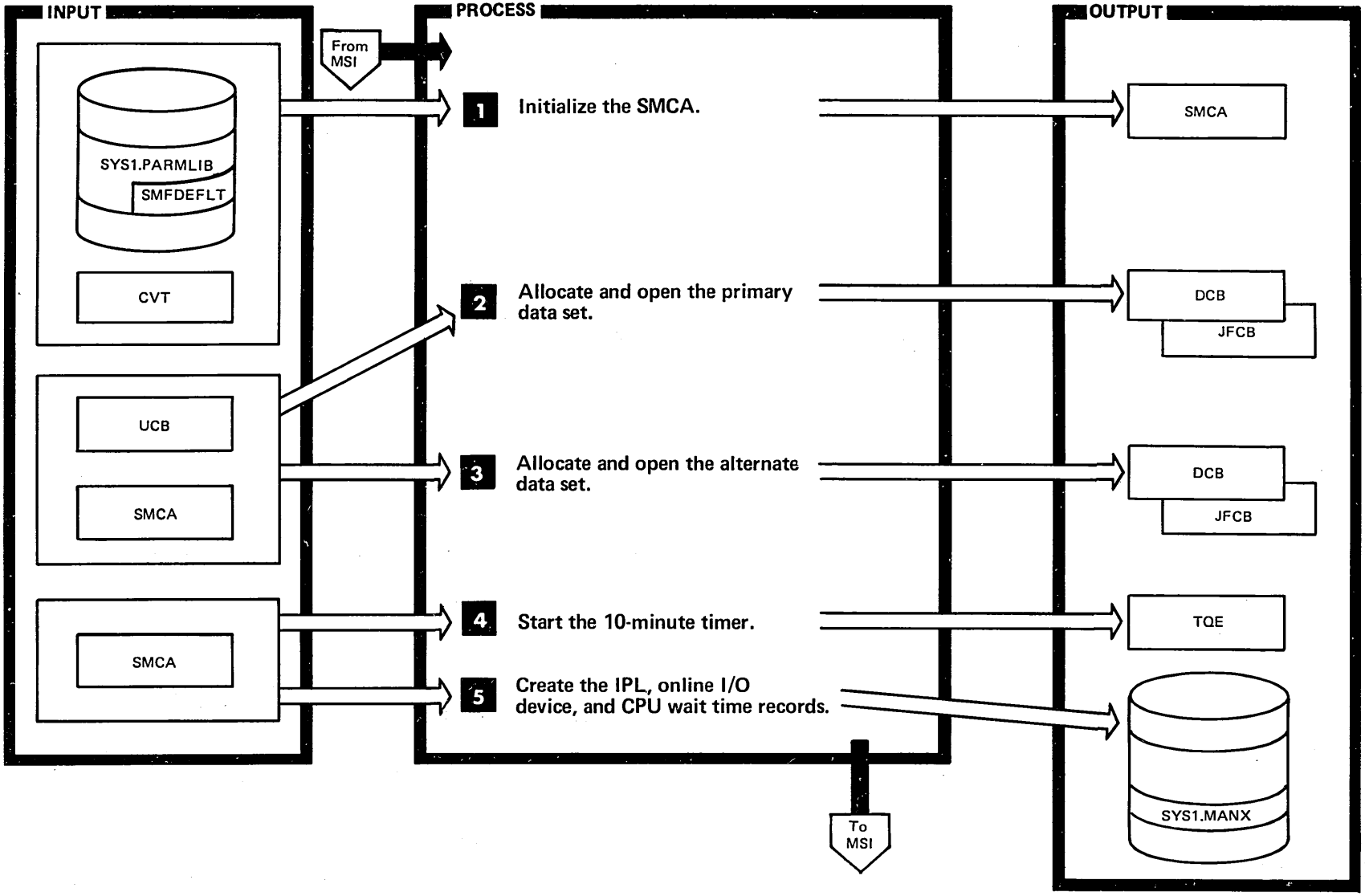


Figure 2-72. (Part 2 of 2)

Extended Description	Module	Figure
1 IEESMFIT obtains space for the SMCA and places a pointer to the area in the CVT. The SMCA is initialized with the parameters from the SMFDEFLT member of SYS1.PARMLIB. If the member is missing or an I/O error occurs, the parameters can be entered from the operator's console.	IEESMFIT IEESMFI3	3-56 3-133
2 3 The SMF allocation routine checks the device type and status of the volume requested for SMF to make sure the volume is online, not busy, public, and on a direct-access device.	IEESMFAL	3-56 3-133
The DCB and JFCB for the data set are initialized and the queue manager writes the JFCB to the job queue.	IEESMFOI	3-56 3-133

Extended Description	Module	Figure
4 The timer is started by enqueuing the timer queue element (TQE).	IEESMFI2	3-56 3-133
5 The CPU wait time record contains the amount of wait time collected during IPL, up until the ten-minute TQE is enqueued.	IEESMFI2	3-56 3-133
The SMF routine links to IEEDFINA to build the dynamic storage configuration record.	IEESMFI2	

Figure 2-73. SMF Writer (Part 1 of 2)

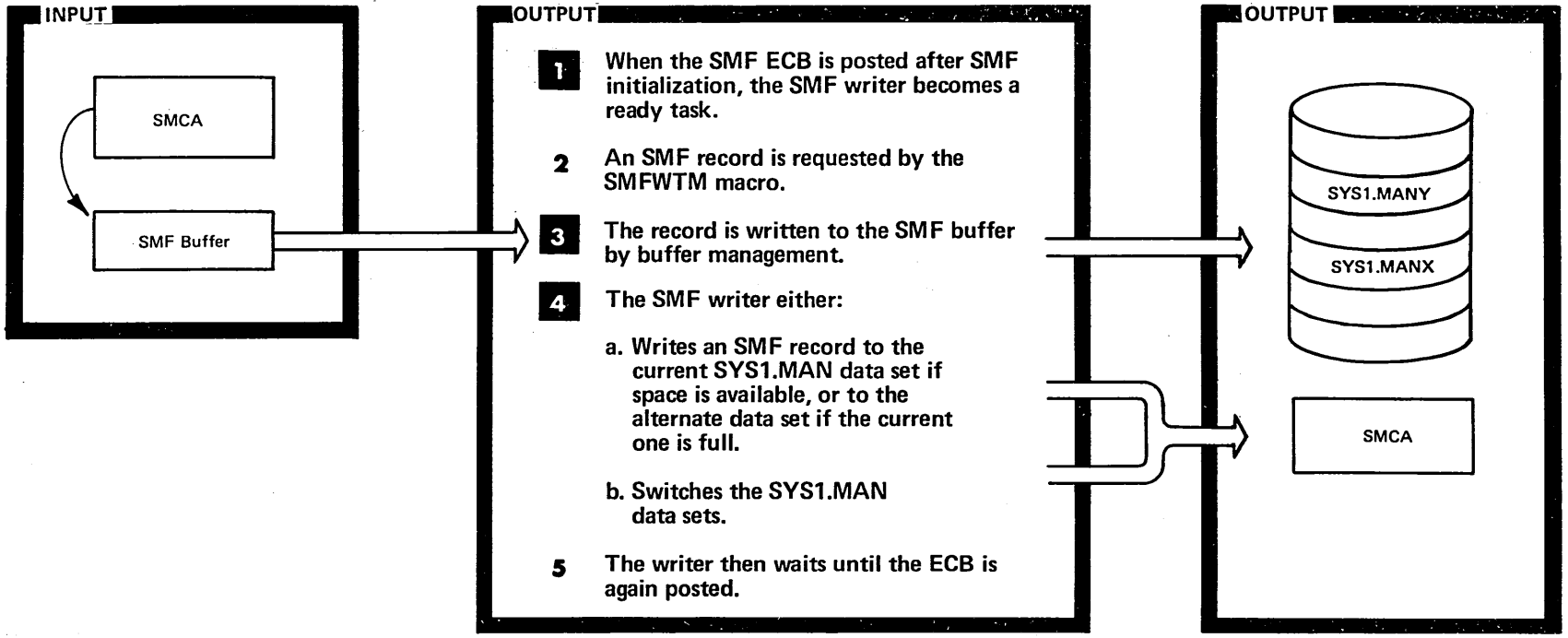


Figure 2-73. (Part 2 of 2)

Extended Description	Module	Figure
1 The SMF writer routine executes the SMF task. The writer receives control via an ATTACH macro instruction issued in the SMF open initialization routine. During SMF initialization, the pointer to the SMCA is stored in the CVT. After this occurs and a local ECB is posted, the SMF writer posts the SMF buffer ECB to indicate that the writer is ready for work. Then the writer waits on an ECB in the SMCA which is posted when the SMF buffer is full. The SMF task is never terminated.	IEESMFWT	

Extended Description	Module	Figure
3 When the active buffer is full, the SMCA ECB is posted.		
4A If the current data set does not have enough space for another record, the SMF routines request a dump of the current data set to tape, and then switch to the alternate data set. If the new data set can be obtained, writing continues. Otherwise, a data-lost condition exists causing a type 7 record to be written to the alternate buffer and message IEE361I to be written to the operator.	IEESMFWT	3-136
4B The status (current or alternate) of the SYS1.MANX and SYS1.MANY data sets can be switched, by request, when the current data set is not full.	IEESMFWT	3-133

Figure 2-74. SMF User Exit Facilities (Part 1 of 3)

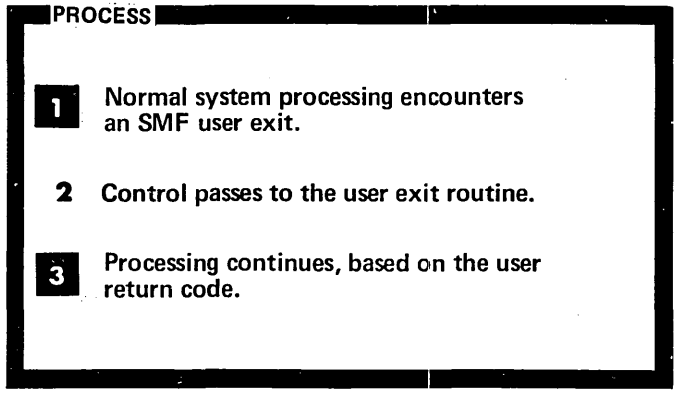


Figure 2-74. (Part 2 of 3)

Extended Description | **Module** | **Figure**

1 User exits occur at these times during processing:

SMF=BASIC (see Figure 2-75)

Module:	At:	Interfaces with:	For user exit:
IEFSD21Q	Allocation	IEFACTLK	IEFACTRT
IEFSD31Q	Job Termination	IEFACTLK	IEFACTRT
IEFZGST2	Step Termination	IEFACTLK	IEFACTRT
IEFSMPUT	Output Limit Expiration	IEFSMFSO	IEFUSO
IEFOSC05	Job Purge	IEFACTLK	IEFUJP

SMF=FULL (see Figure 2-76)

Module:	At:	Interfaces with:	For user exit:
IEFSD162	Job Initiation	IEFSMFIE	IEFUJI
IEFSD162	Step Initiation	IEFSMFIE	IEFUSI
IEFSD31Q	Job Termination	IEFSMFLK	IEFACTRT
IEFZGST2	Step Termination	IEFSMFLK	IEFACTRT
IEFVMB	JES Reader	IEFUIV	IEFUIV
IEFVHH	Interpreta-tion	IEFUJV	IEFUJV
IEFVHEB	Interpreta-tion	IEFUJV	IEFUJV

Figure 2-74. (Part 3 of 3)

Extended Description	Module	Figure

SMF = FULL (cont'd)

Module:	At:	Interfaces with:	For user exit:
IEFSMPUT	Output Limit Expiration	IEFSMFSO	IEFUSO
IEATLEXT	Timer Expiration*	IEFUTL	IEFUTL
IEFOSC05	Job Purge	IEFACTLK	IEFUJP

* See the Supervisor Logic Manual listed in the Preface.

3 User exits return to the calling module via a BR 14.

The user can add information to the SMF record before it is written to the SMF data set. He can also decide to cancel the job or, in the case of output limit expiration, extend the number of records in the output data set.

For information about use of the SMF user exists, see the System Management Facilities manual listed in the Preface.

Figure 2-75. Basic SMF User Exit Facilities (Part 1 of 2)

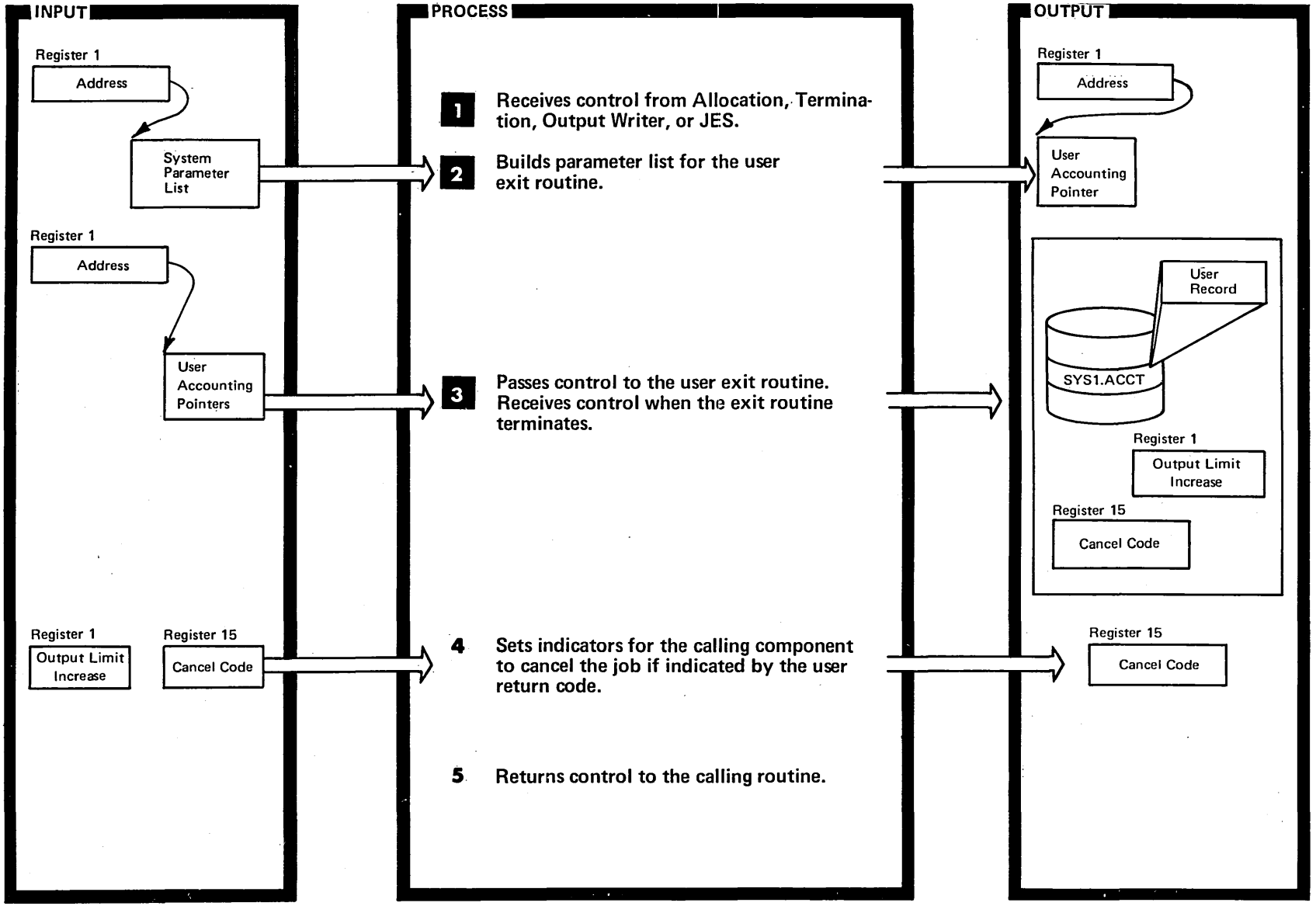


Figure 2-75. (Part 2 of 2)

Extended Description	Module	Figure
1 When the user specifies SMF=BASIC at system generation, interfaces are created between the control program and user exits. Figure 2-73 lists the calling modules, the times when the exits are taken, the interfaces, and the exit routines. Input to the SMF module creating the interface varies, but it is always in the form of a parameter list pointed to by register 1. A job can use more than one exit.		
2 The length and contents of the parameter list vary, depending on which user exit is taken.	IEFACTLK	

Extended Description	Module	Figure
3 For SMF=BASIC, the user cannot write to SYS1.MANX or SYS1.MANY. He can, however, write user records to the SYS1.ACCT data set.	IEFACTLK	
If the user indicates, by the return code in register 15, that the job is not to continue, the job is canceled. A job cannot be canceled during the job-purge exit because the job has already been completed.		
For the output limit exit, register 15 indicates whether or not the job is to be canceled. If the job is to continue, the user must place the output limit *increase* in register 1.	IEFSMFSO	

Figure 2-76. **Full SMF User Exit Facilities (Part 1 of 2)**

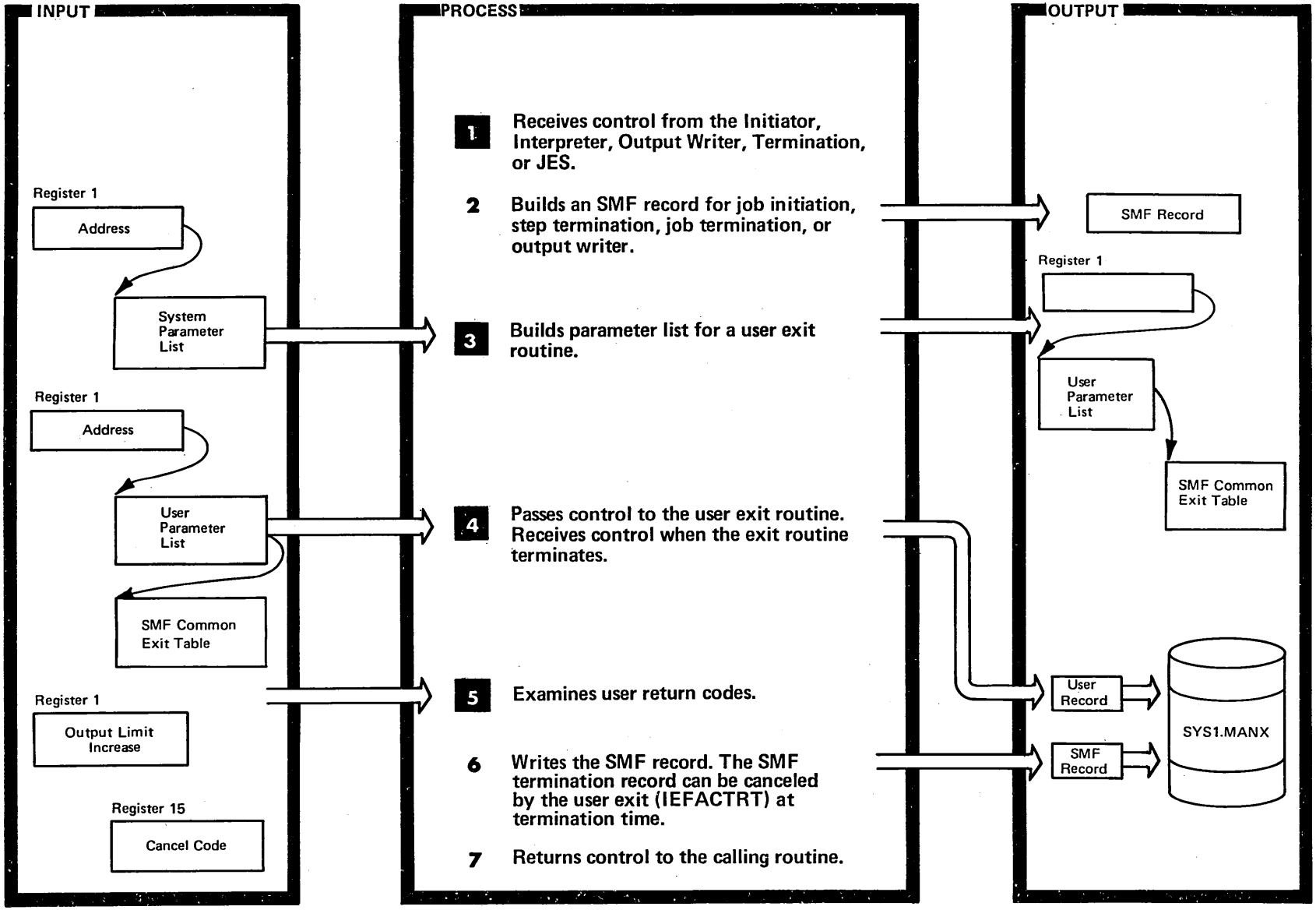


Figure 2-76. (Part 2 of 2)

Extended Description	Module	Figure
When the user specifies SMF-FULL at system generation, interfaces are created between the control program and user exits. Figure 2-74 lists the calling modules, the times when the exits are taken, the interfaces, and the exit routines. If EXT=NO, the exits are not taken. EXT is specified at IPL time via the SMFDEFLT or by the operator. Input to the SMF module creating the interface varies, but it is always in the form of a parameter list pointed to by register 1. A job can use more than one exit. The length and contents of the parameter list vary, depending on which user exit is taken. However, the first word of the list always points to the 76-byte SMF common exit table. For the format of the parameter list and details on writing exit routines, see the System Management Facilities manual listed in the Preface.	IEFSMFLK IEFACTLK	

Extended Description	Module	Figure
For SMF=FULL, and MAN=ALL or MAN=USER the user can write his own records to the current SMF data set using the SMFWTM macro instruction in the exit routine. For details, see the System Management Facilities manual listed in the Preface If the user indicates, by the return code in register 15, that the job is not to continue, the job is canceled by the calling routine. A job cannot be canceled from the job-purge exit because the job has already been completed. No SMF records are written for a job canceled by exit IEFUIV. For the output limit exit, register 15 indicates whether or not the job is to be canceled. If the job is to continue, the user must place the output limit *increase* in register 1. For the job and step termination exists, the user can cancel the SMF record by a return code in register 1. For the time limit exit, Register 15 indicates whether or not the job is to be canceled. If the job is to continue, the user must place the additional time in Register 1.	IEFSMFLK IEFACTLK IEFSMFSO IEFSMFLK IEFUSO IEFUTL	

Section 3: Program Organization

CONTENTS

Introduction

This section contains functional flow diagrams describing the program organization of job management. Each figure traces a flow of control, from module to module, for a particular function. The figures show common paths through the modules and serve as a guide for other possible execution conditions.

Each group of figures concerning one job scheduling element begins with a module interconnection diagram showing all the modules in that element. The functional flow diagrams follow. Each block that describes a module contains:

- the module's name
- a number in the upper right-hand corner of the diagram signifying the number of times the module is entered, for that function, up to that point in the flow.
- its common name
- its entry point
- the major functions the module performs
- a message code indicating which messages, if any, originate in that module.

If the block is a reference to another component, only the component name and function are given.

A *Dictionary of Abbreviations* appears at the end of this manual.

Figure 3-1. Initiator Interconnection Module Diagram (Part 1 of 3)

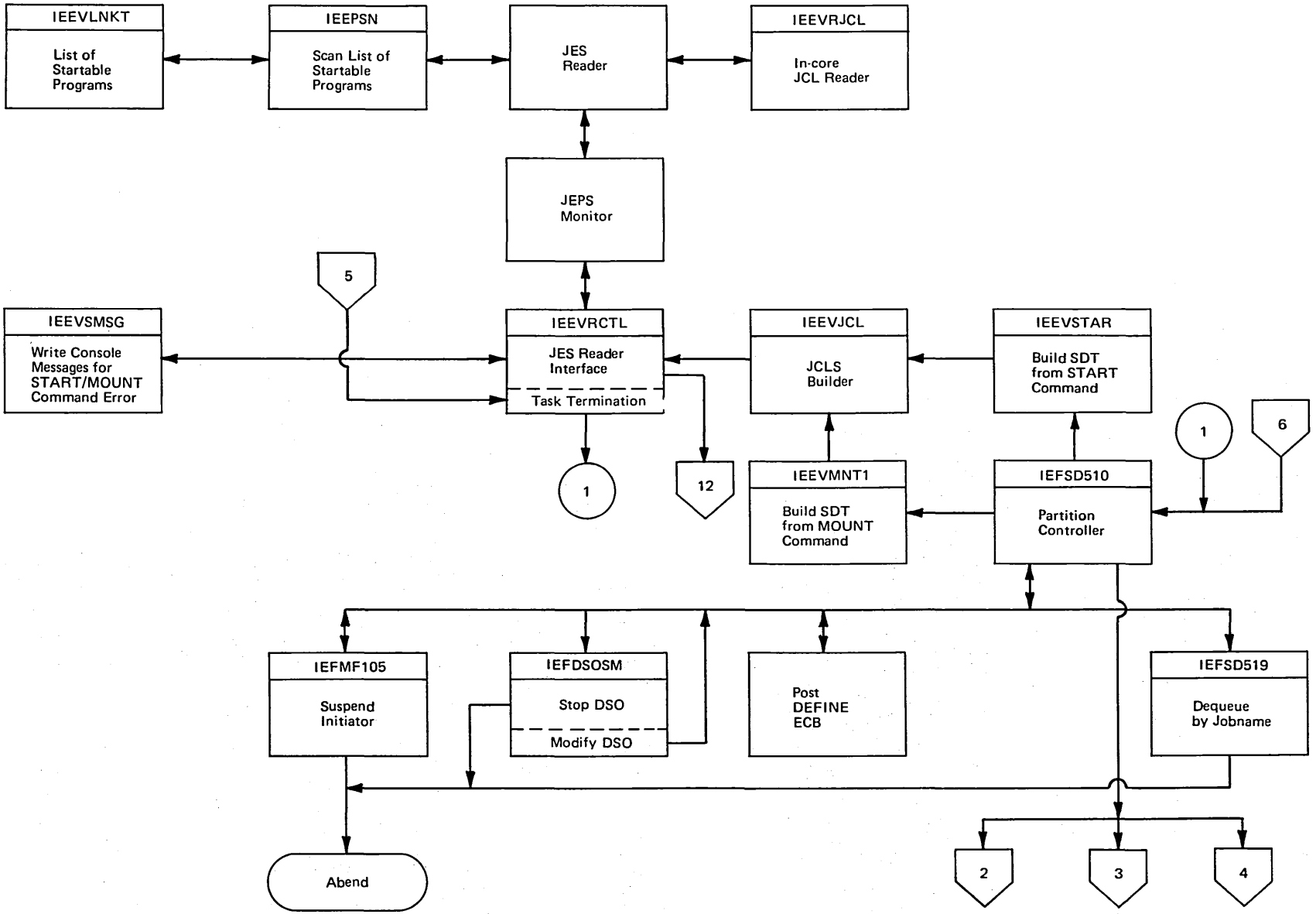


Figure 3-1. Initiator Interconnection Module Diagram (Part 2 of 3)

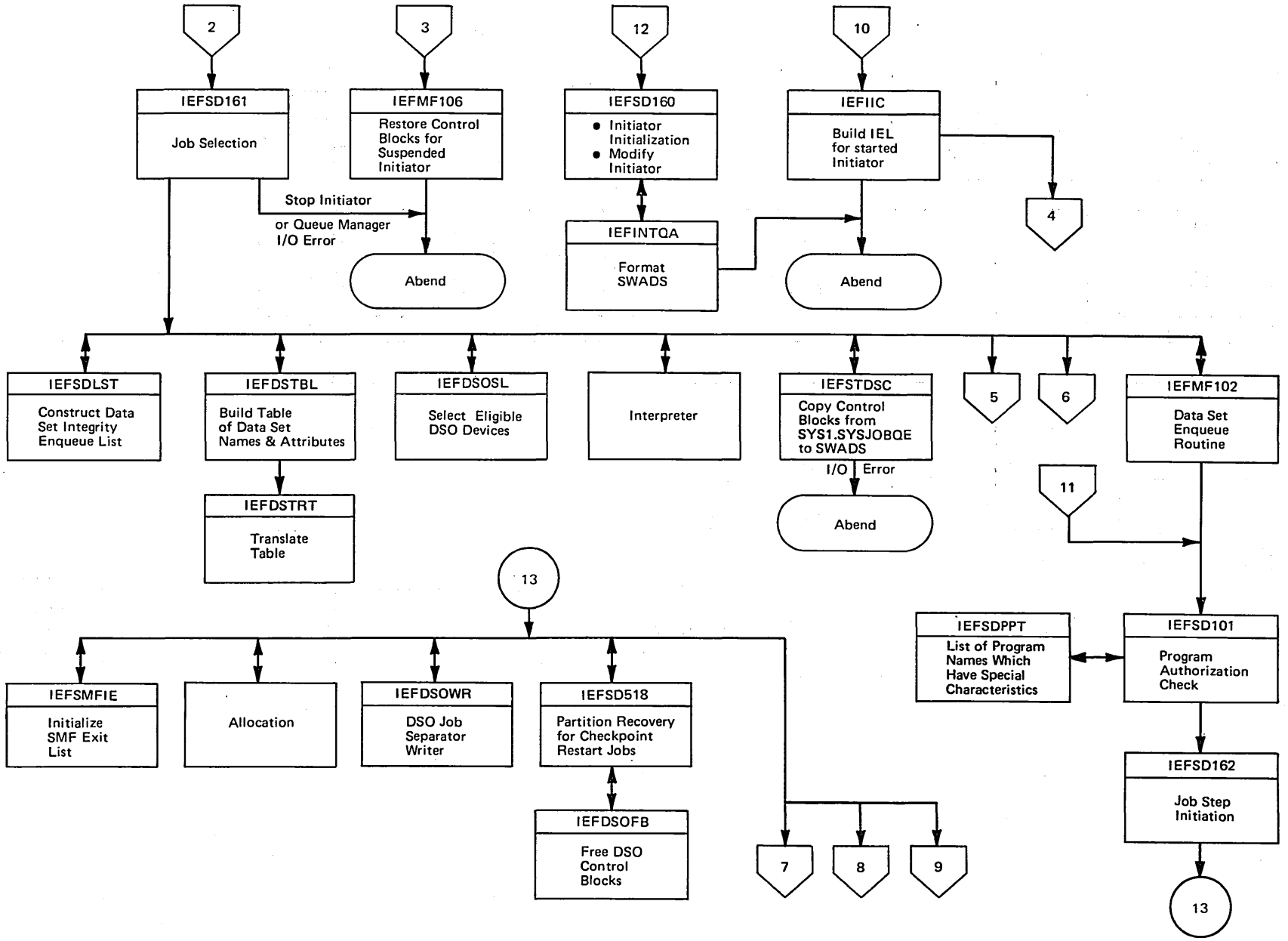


Figure 3-1. Initiator Interconnection Module Diagram (Part 3 of 3)

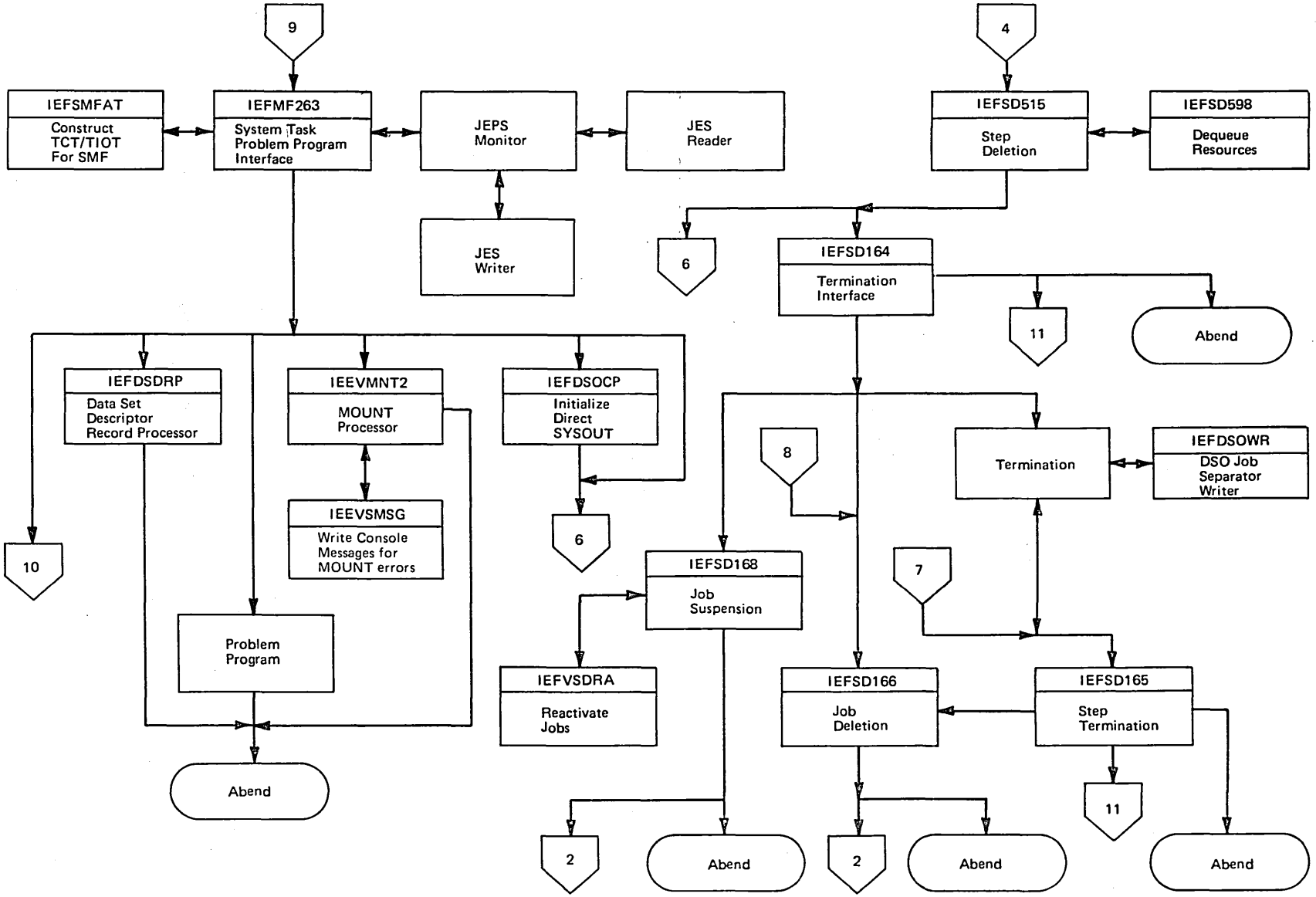


Figure 3-2. Starting an Initiator (Part 1 of 2)

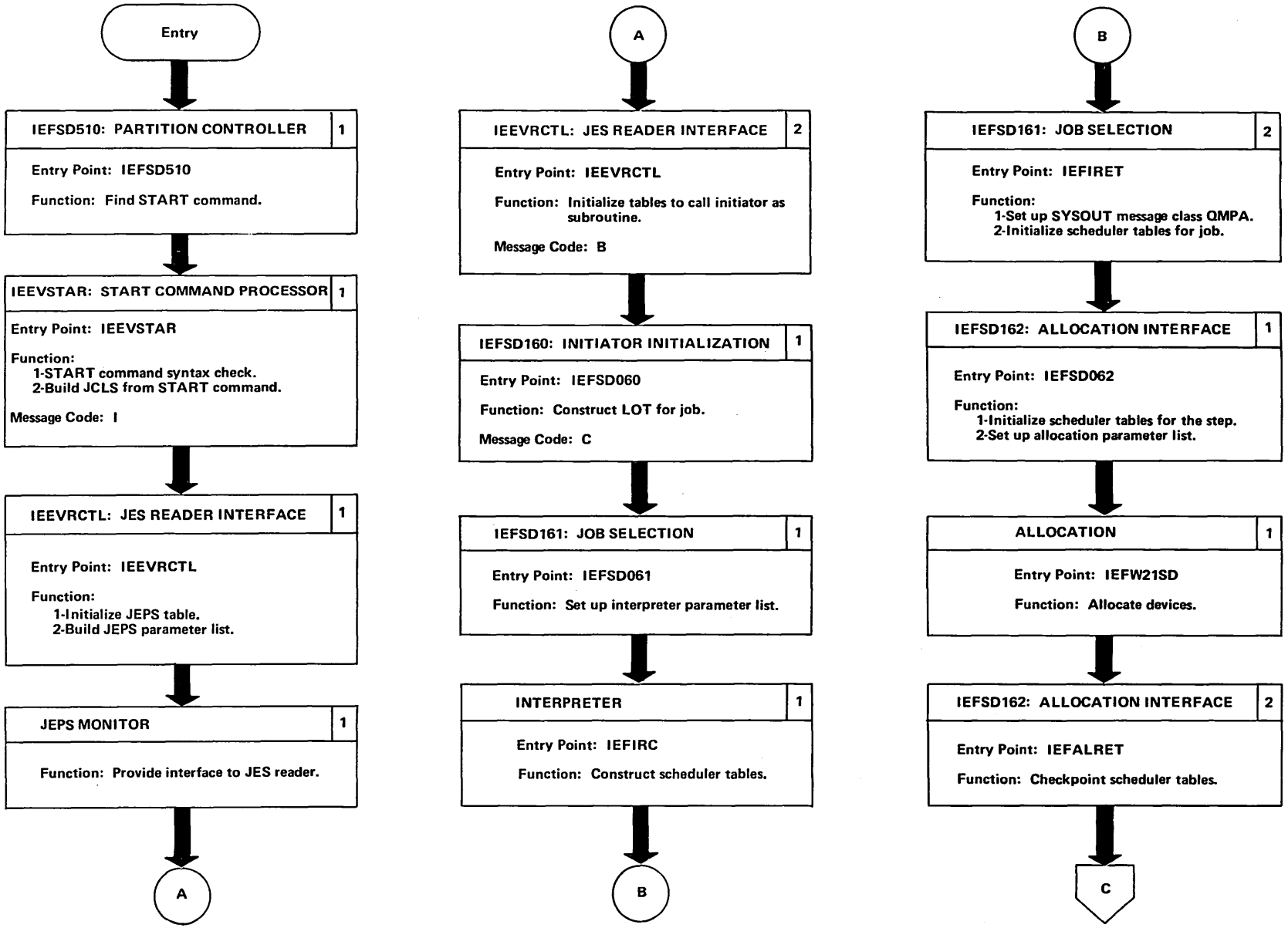
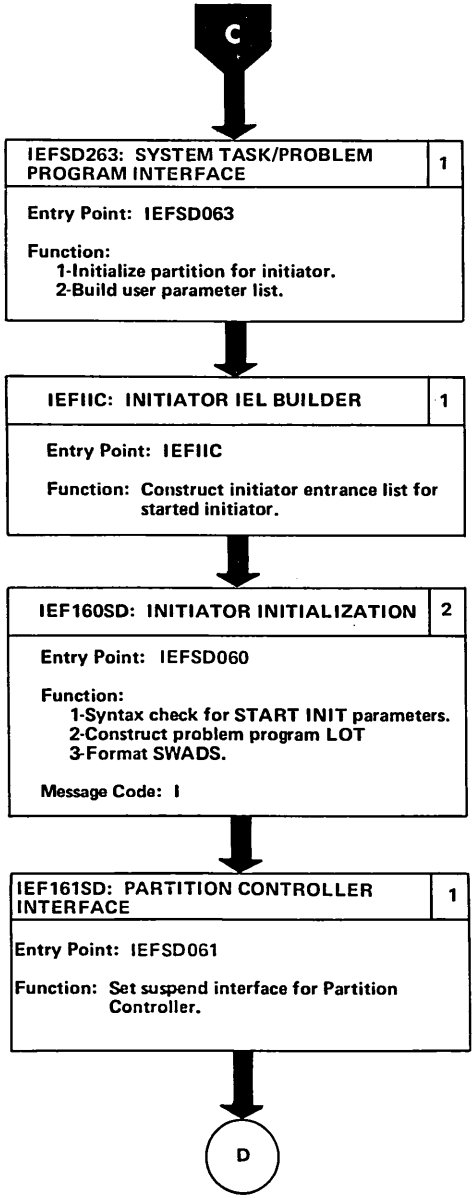
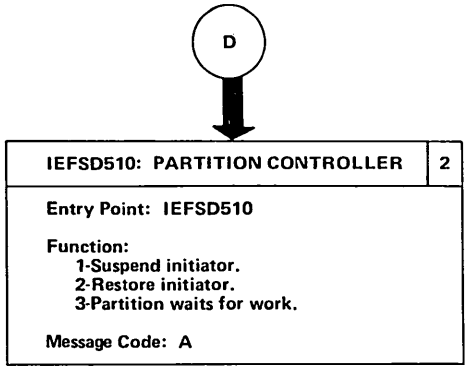


Figure 3-2. Starting an Initiator (Part 2 of 2)





Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
I	IEE307I IEE308I IEE309I IEE311I	Delimiter error Length error Unidentifiable keyword Parameter missing
B	IEE191I IEE192I IEE122I IEE132I IEE121I IEE002I	DD entry missing Invalid PARM field format JCL error Device allocation error Job queue I/O error Partition not specified
C	IEF427I IEF203I IEF204I IEF205I IEF206I IEF016I IEF017I	Insufficient queue space Too many operands Too many classes Invalid operand No classes specified Invalid unit SWADS OPEN failure

Figure 3-3. Starting a Reader/Writer or RTAM (Part 1 of 2)

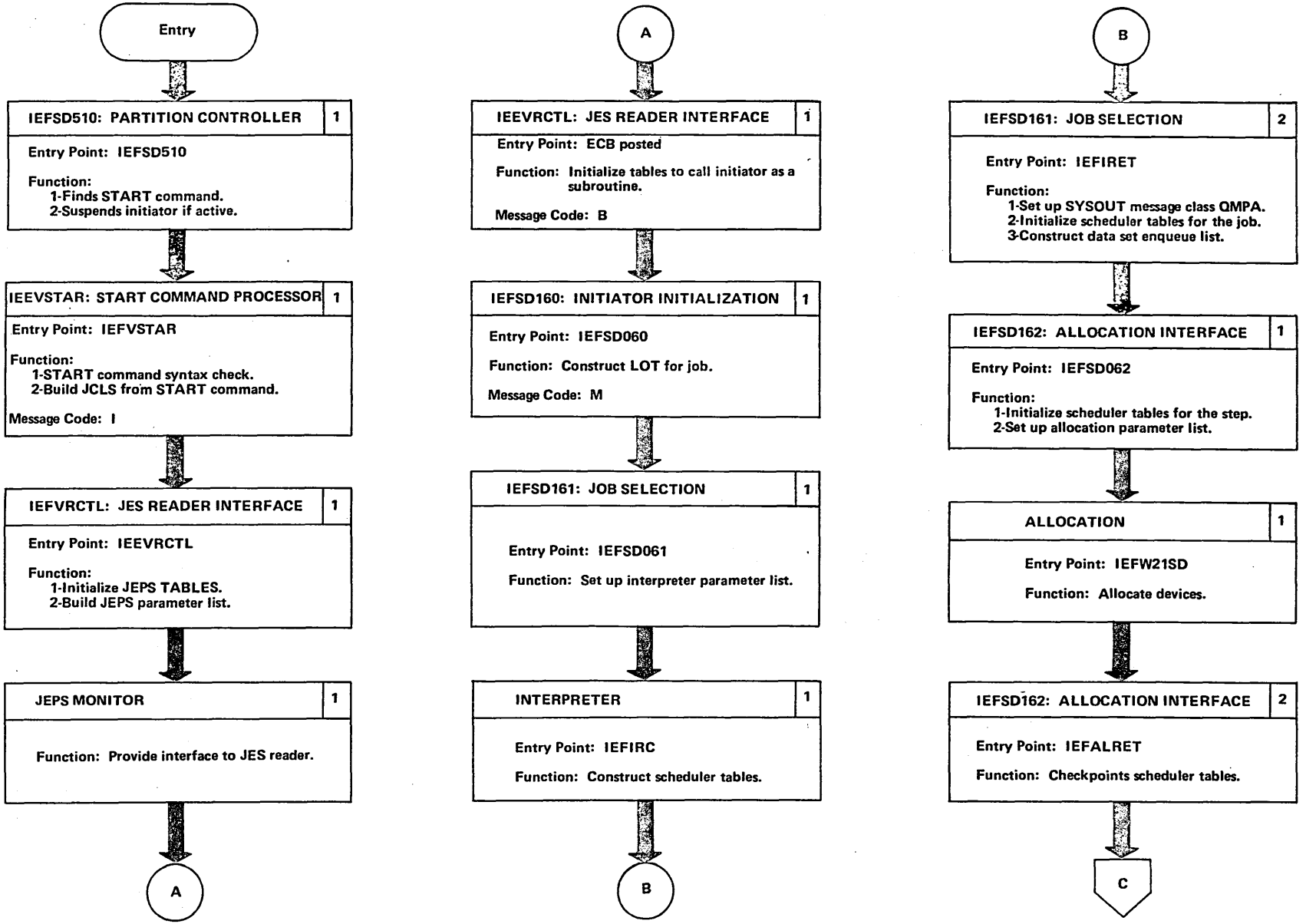
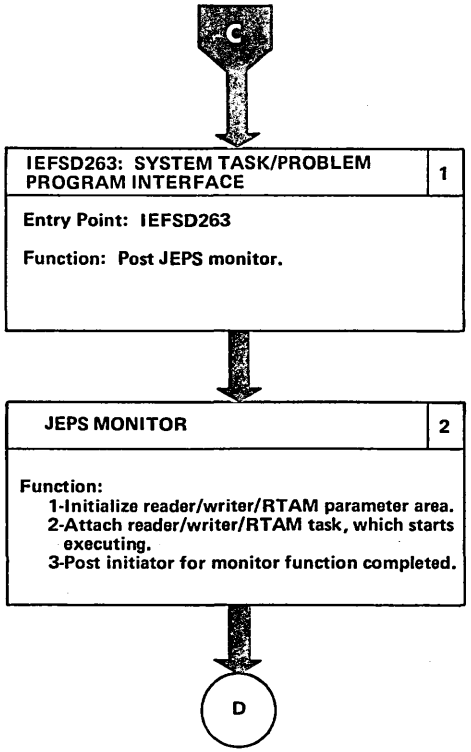
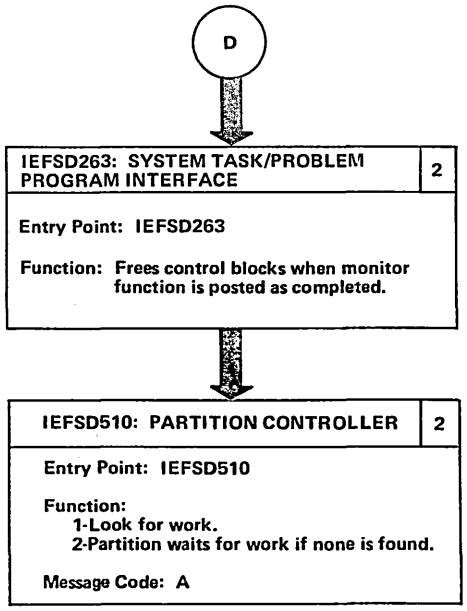


Figure 3-3. Starting a Reader or Writer (Part 2 of 2)





Messages Code	Number	Reason
L	IEE307I	Delimiter error
	IEE308I	Length error
	IEE309I	Unidentifiable keyword
	IEE311I	Parameter missing
M	IEF427I	Insufficient queue space
	IEF203I	Too many operands
	IEF204I	Too many classes
	IEF205I	Invalid operand
	IEF206I	No classes specified
	IEF016I	Invalid unit
A	IEF005E	Partition waiting for work
B	IEE191I	DD entry missing
	IEE192I	Invalid PARM field format
	IEE122I	JCL error
	IEE132I	Device allocation error
	IEE121I	Job queue I/O error
	IEF002I	Partition not specified

Figure 3-4. Selecting a Job From a Queue (Part 1 of 2)

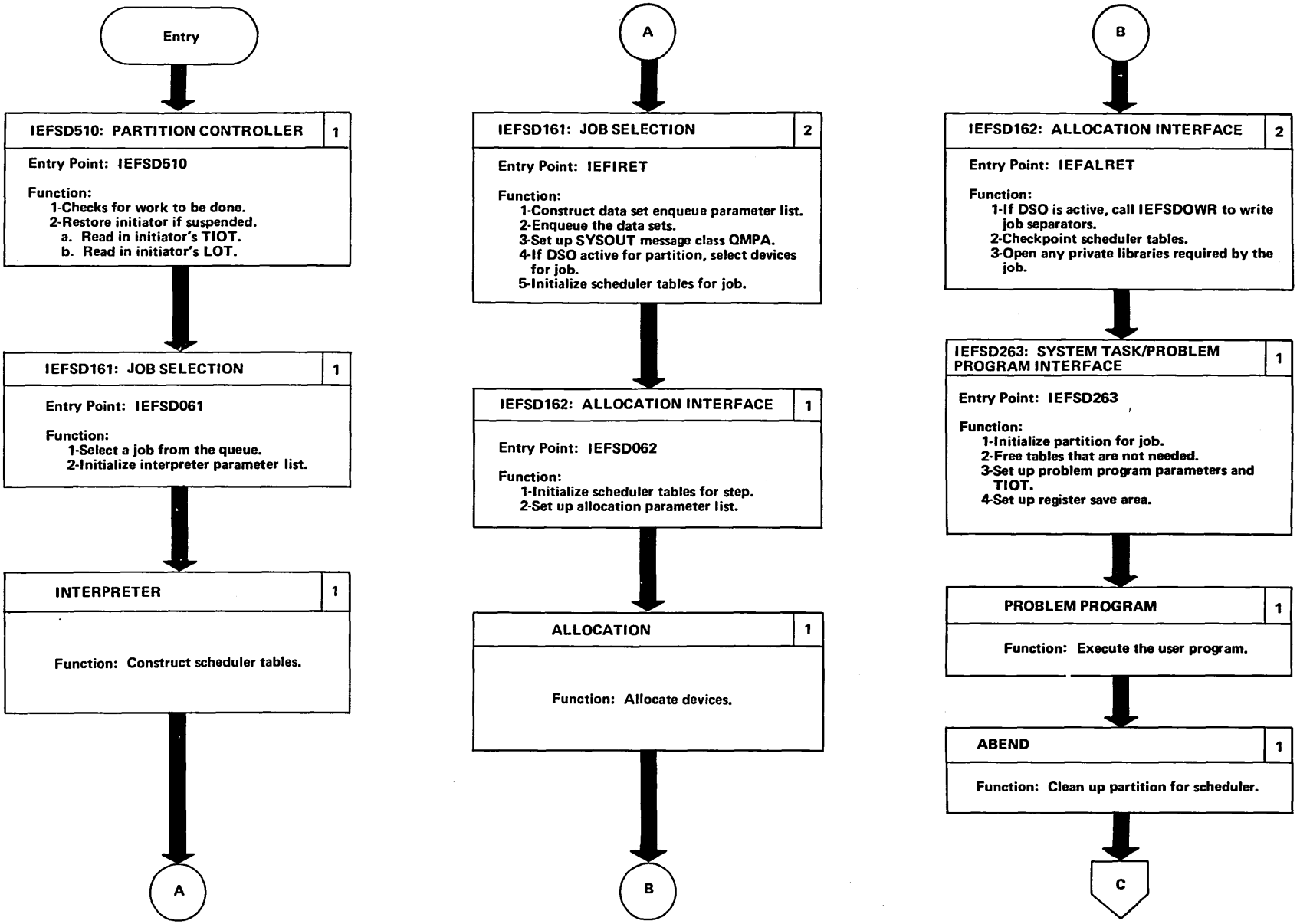
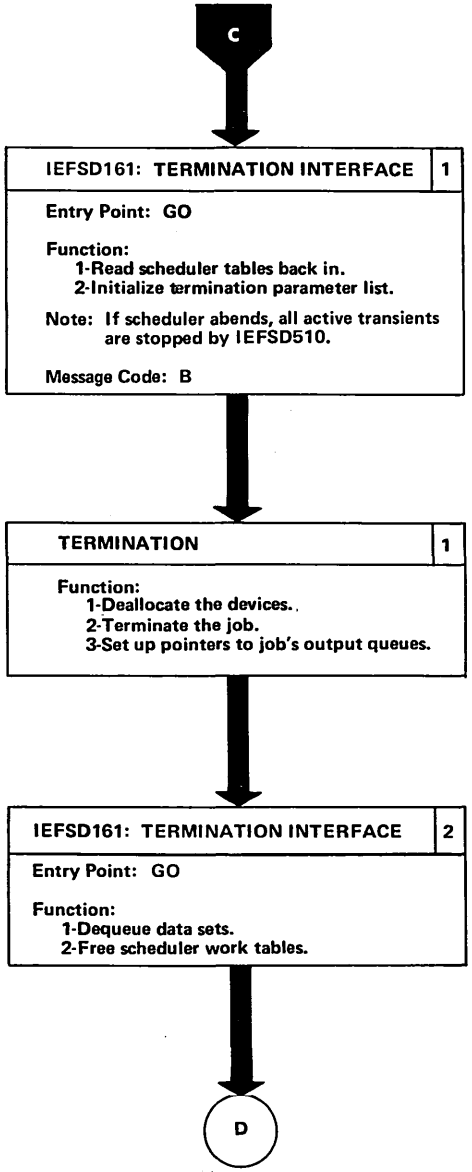
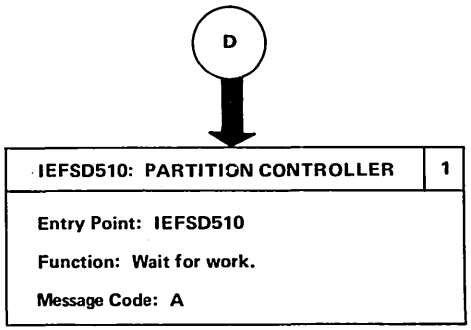


Figure 3-4. Selecting a Job From a Queue (Part 2 of 2)





Message Code	Number	Reason
A	IEF005E	Partition waiting for work
B	IEF866E	Scheduler ABEND completion code =

Figure 3-5. Selecting a Job From a Queue with DSO Active in the Partition (Part 1 of 2)

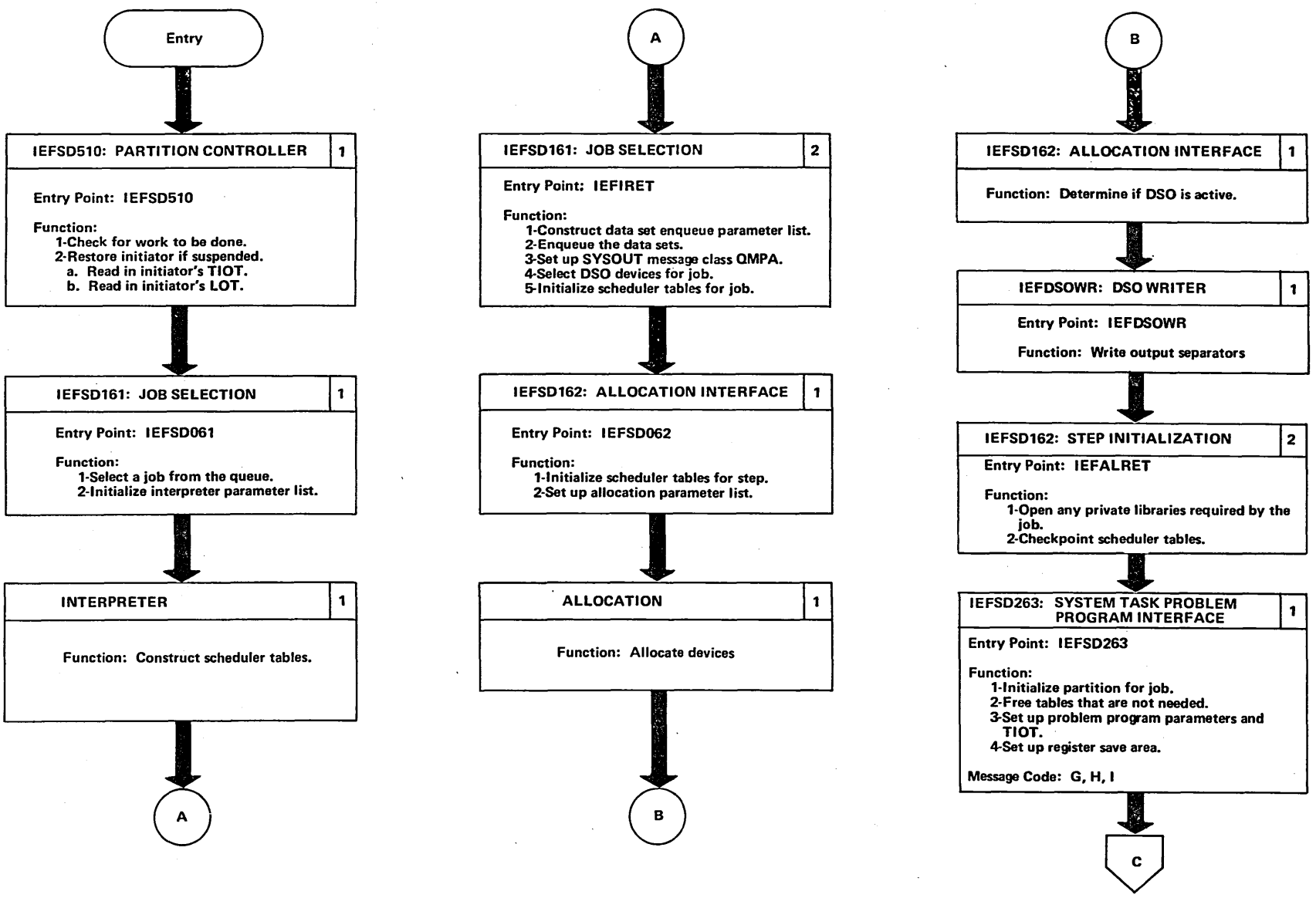
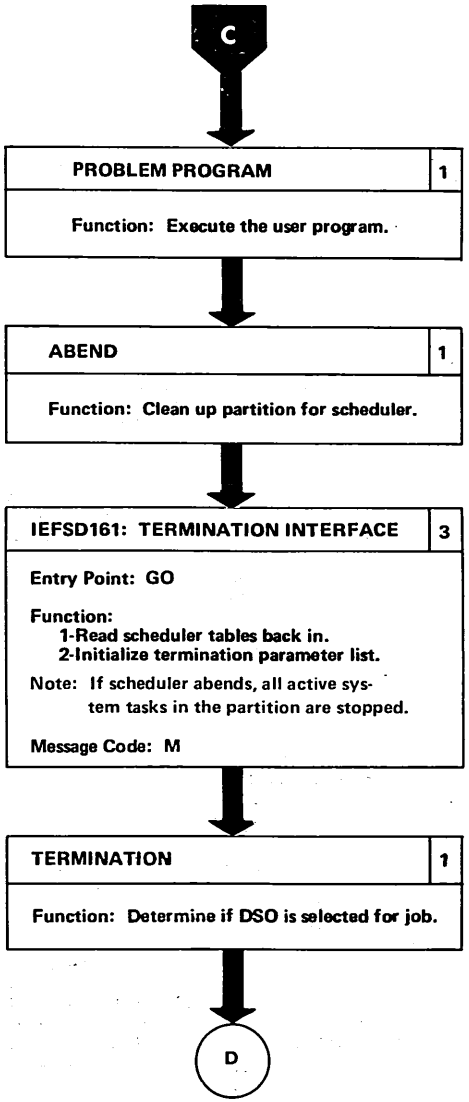
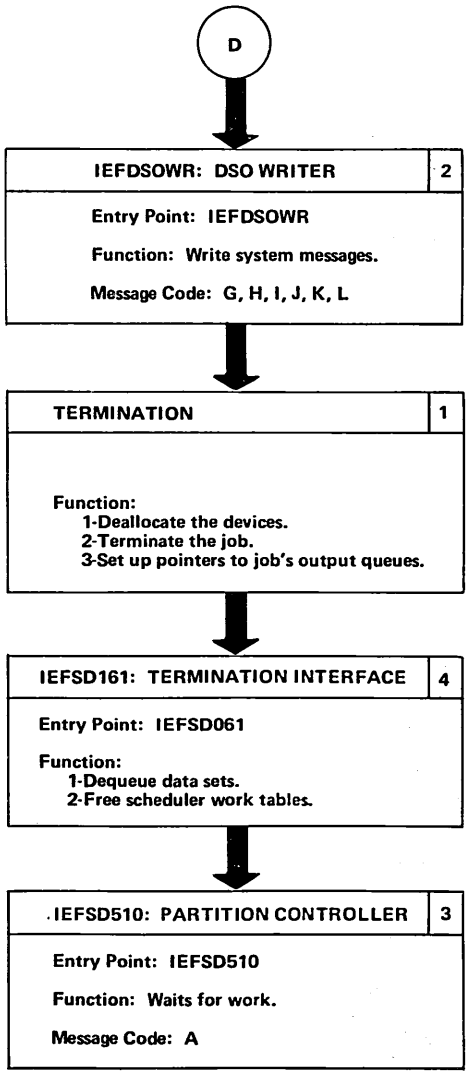


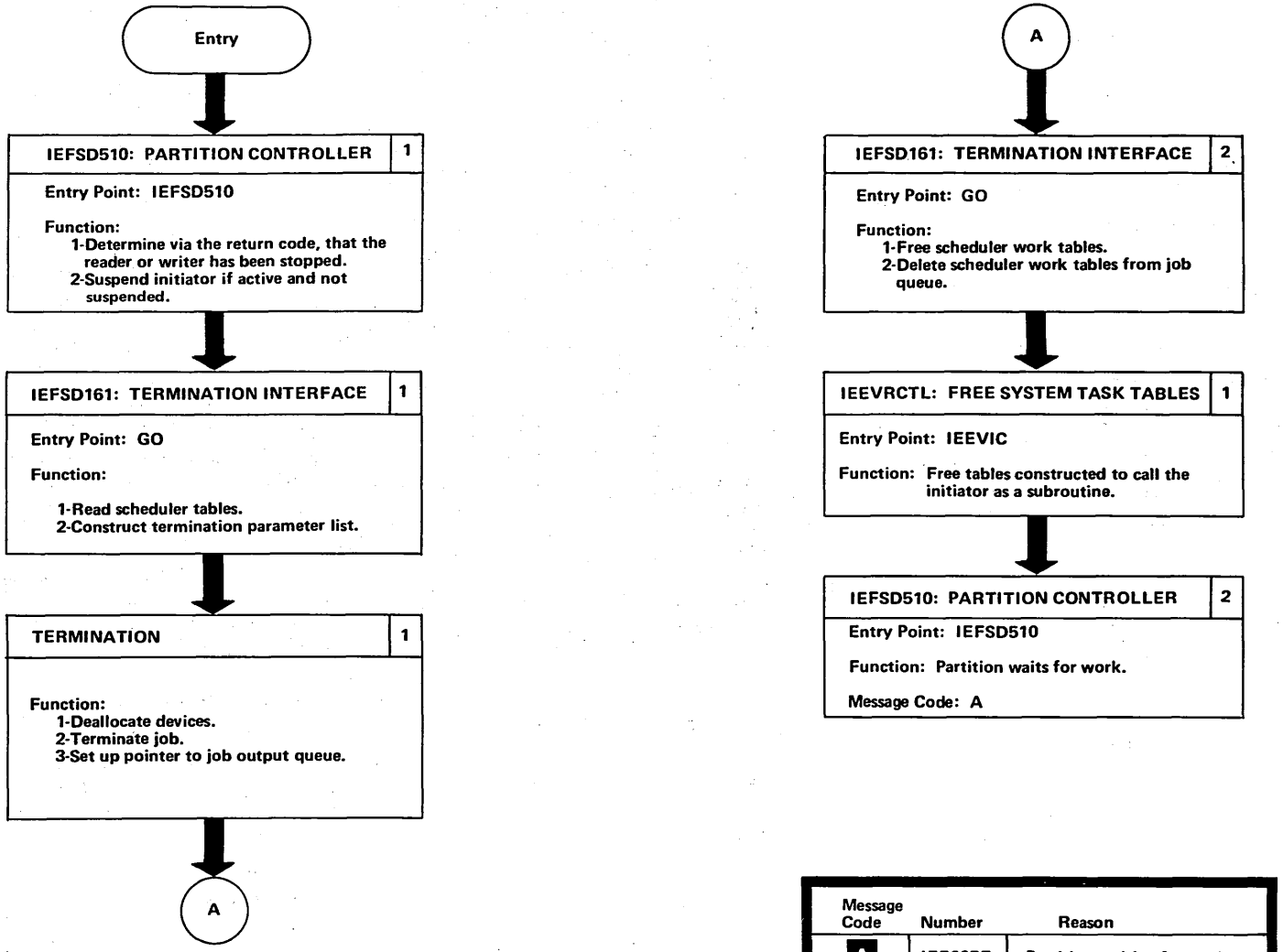
Figure 3-5. Selecting a Job From a Queue with DSO Active in the Partition (Part 2 of 2)





Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
G	IEF386I	Output I/O error
H	IEF385I	OPEN failure
I	IEF395I	Job queue I/O error
J	IEF008I	SYS1.SYSPOOL I/O error
K	IEF009I	System messages enqueued
L	IEF010I	DSOCB unavailable
M	IEF866E	Scheduler abend, completion code =

Figure 3-6. Stopping a Reader or Writer



Message Code	Number	Reason
A	IEF005E	Partition waiting for work

Figure 3-7. Data Set Integrity Processing

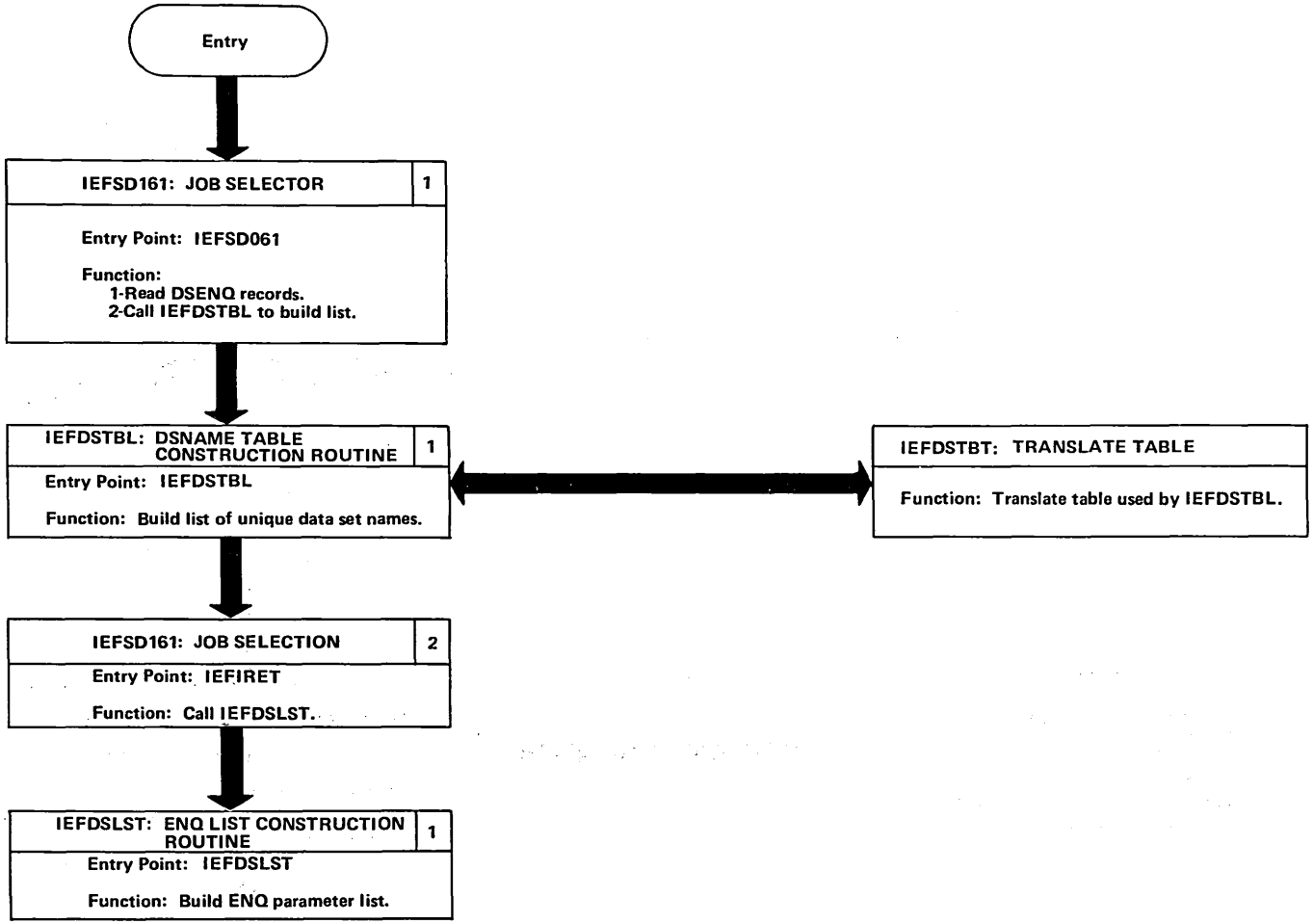
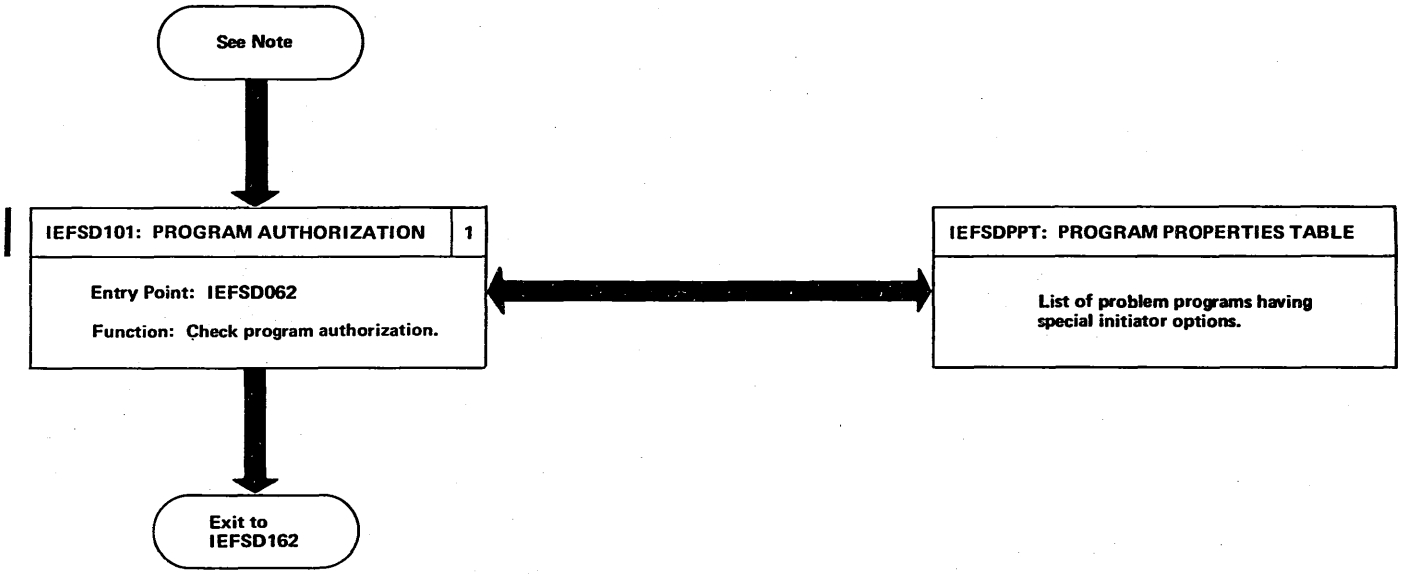


Figure 3-8. Program Authorization



Note: Entry from IEFMF102 for the first step of a job. Entry is from IEFSD164 or IEFSD165 for subsequent steps.

Figure 3-9. Starting DSO (Part 1 of 2)

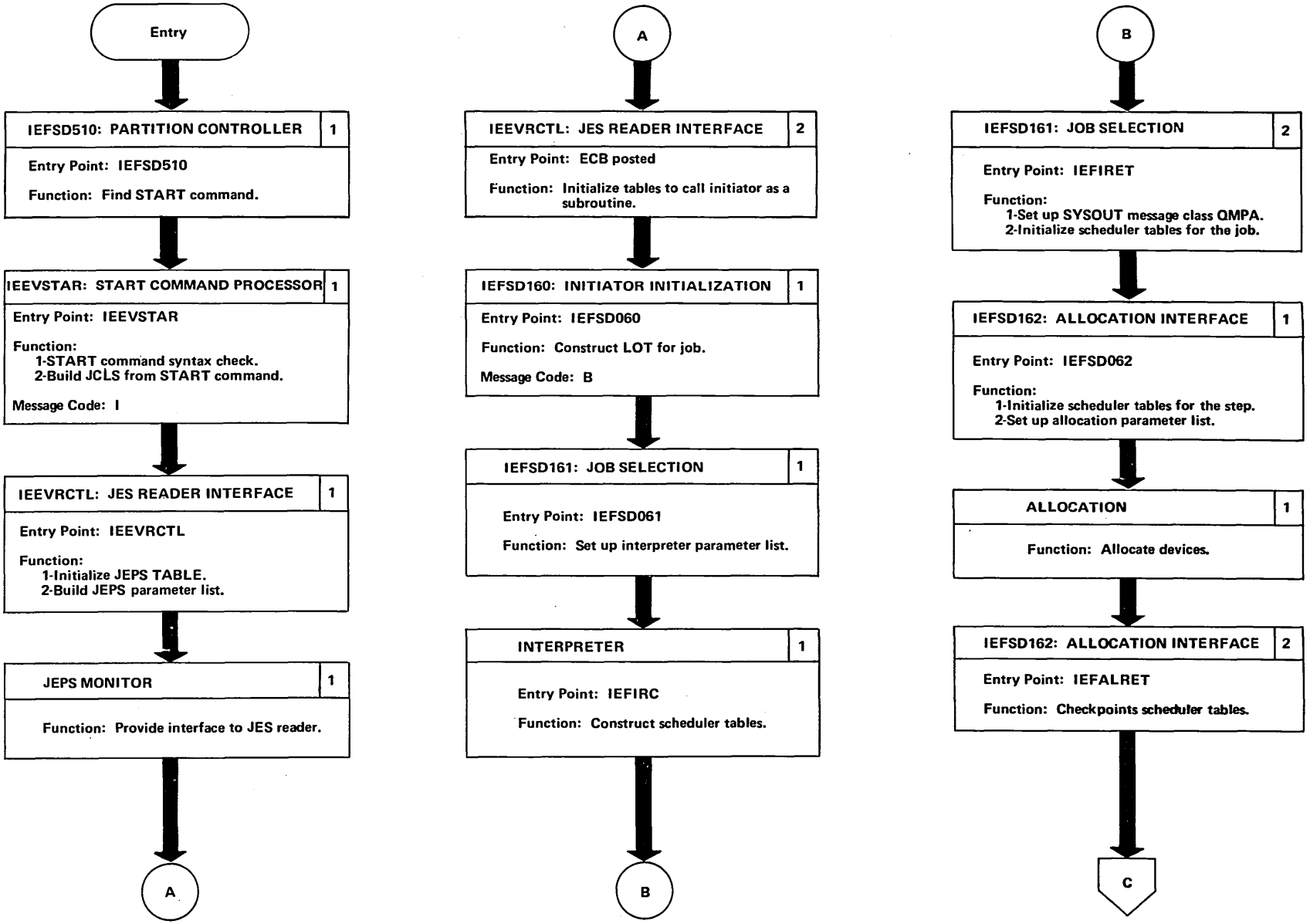
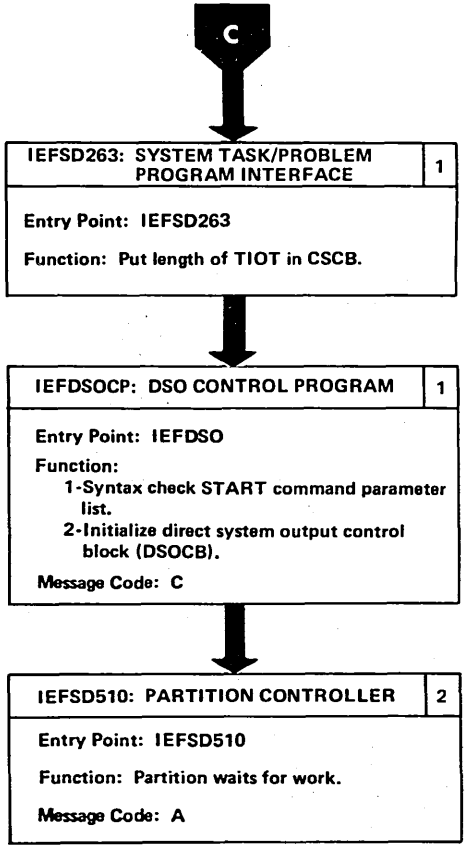
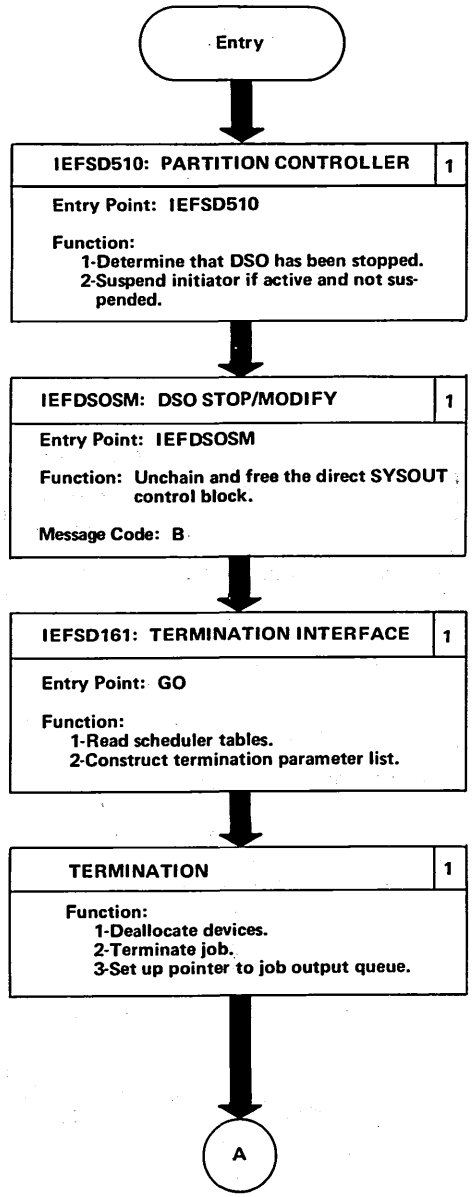


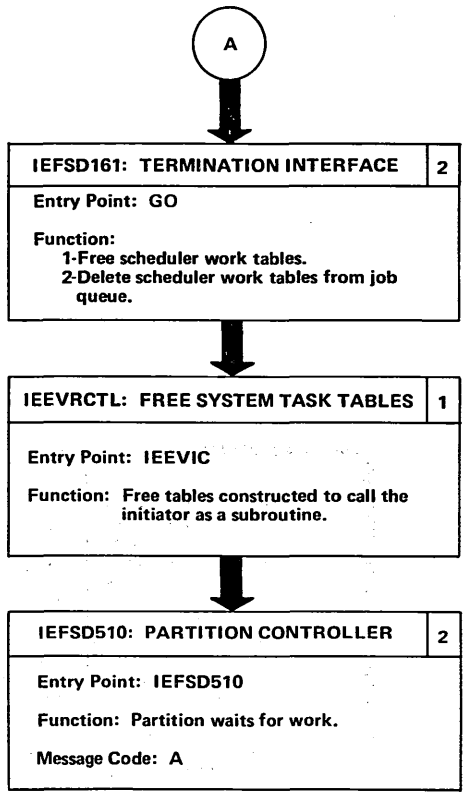
Figure 3-9. Starting DSO (Part 2 of 2)



Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
I	IEE3071 IEE3081 IEE3091 IEE3111	Delimiter error Length error Unidentifiable keyword Parameter missing
B	IEF4271 IEF2031 IEF2041 IEF2051 IEF2061 IEF0161 IEF0171	Insufficient queue space Too many operands Too many classes Invalid operand No classes specified Invalid unit SWADS OPEN failure
C	IEF0111 IEF0121 IEF3931 IEF3941 IEF3981	DSO already exists DSO job queue I/O error Command or procedure error Partition error DSO OPEN unsuccessful

Figure 3-10. Stopping DSO





Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
B	IEF391I IEF397I	DSO closed End of output record not written

Figure 3-11. Starting a Problem Program (Part 1 of 2)

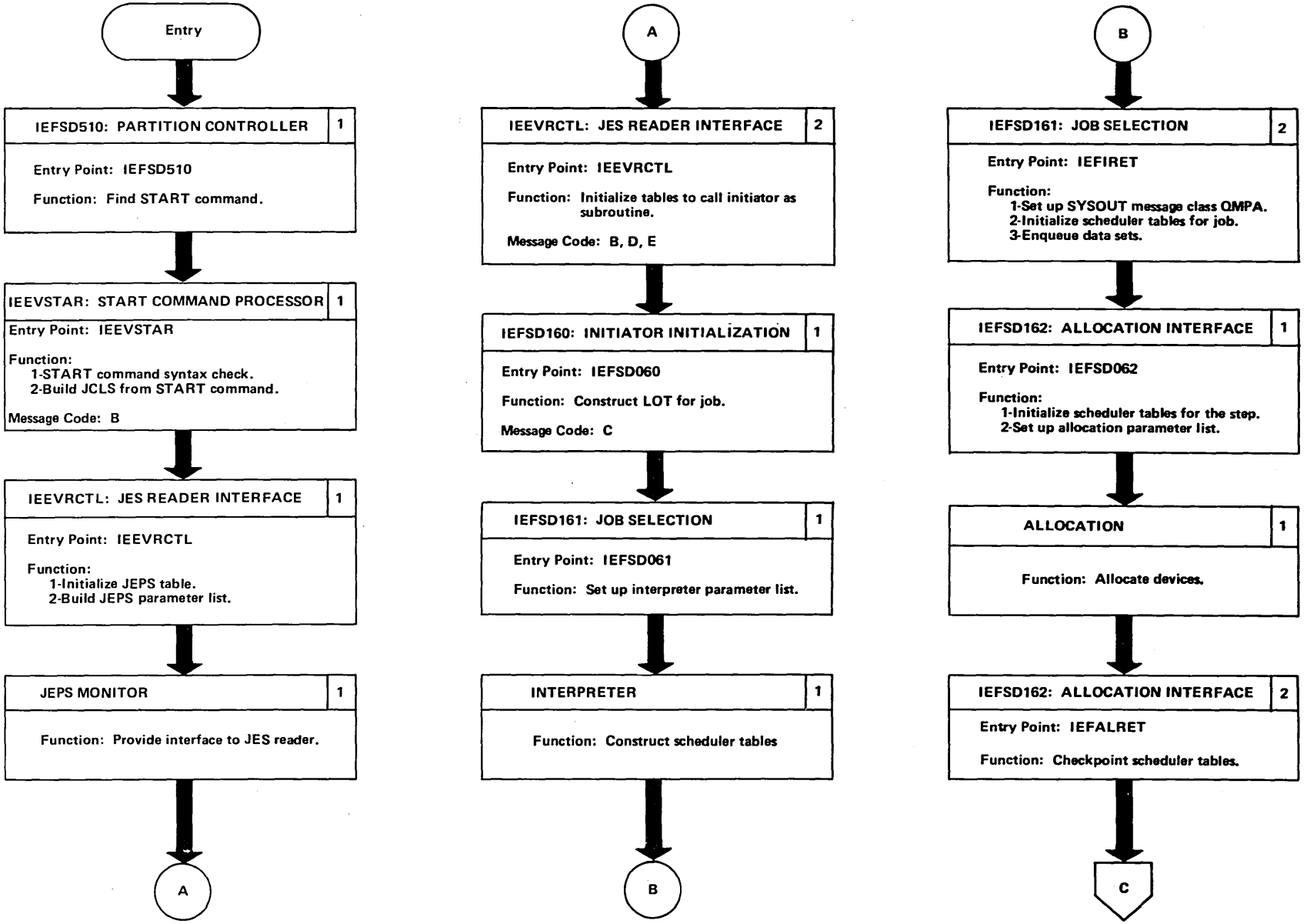
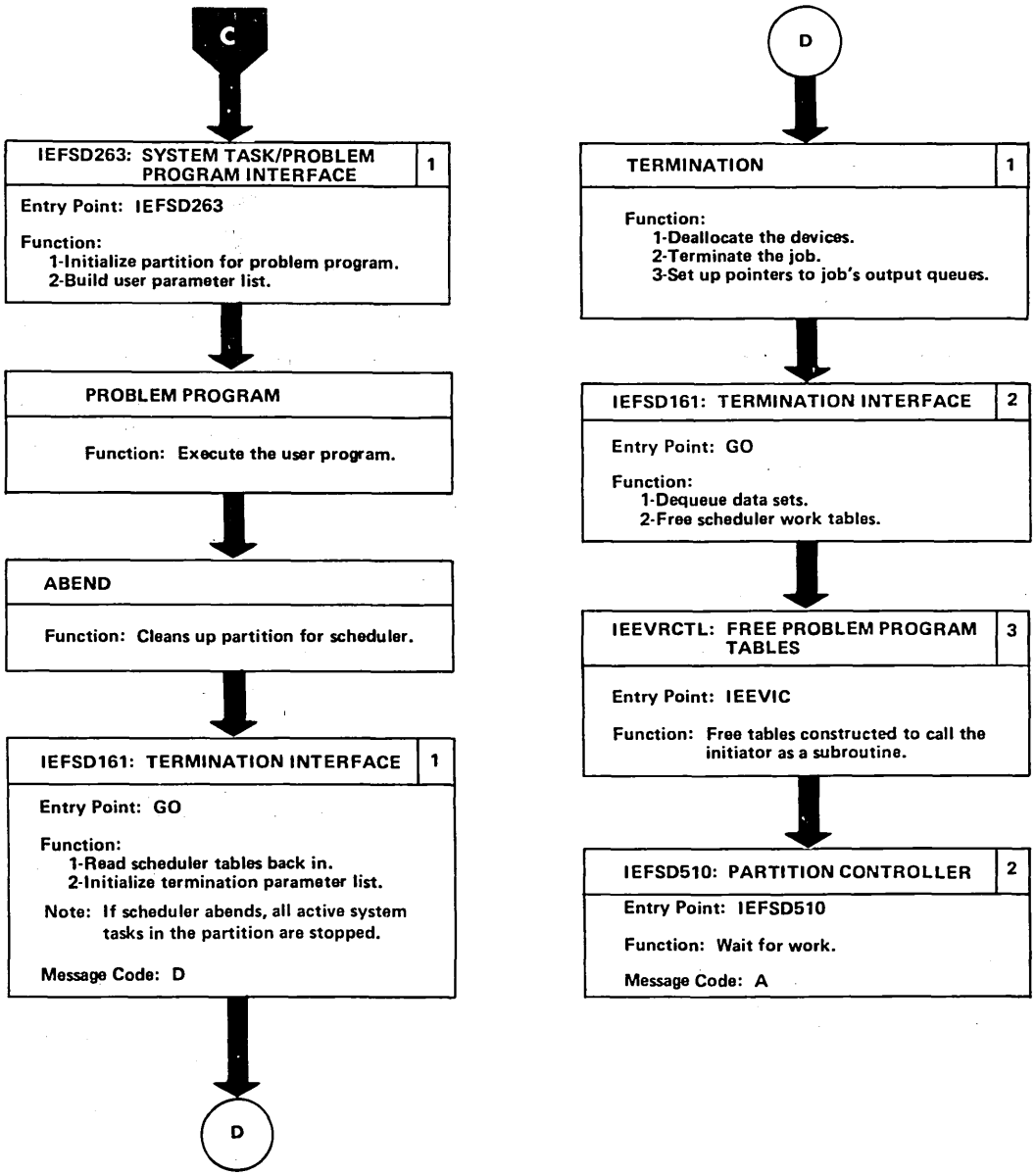


Figure 3-11. Starting a Problem Program (Part 2 of 2)



Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
B	IEE307I IEE308I IEE309I IEE311I	Delimiter error Length error Unidentifiable keyword Parameter missing
C	IEE427I IEE203I IEE204I IEE205I IEF016I IEF017I	Insufficient queue space Too many operands Too many classes Invalid operand Invalid unit SWADS OPEN failure
D	IEF866E	Scheduler abend, completion code =

Figure 3-12. MOUNT Command (Initiator Processing) (Part 1 of 2)

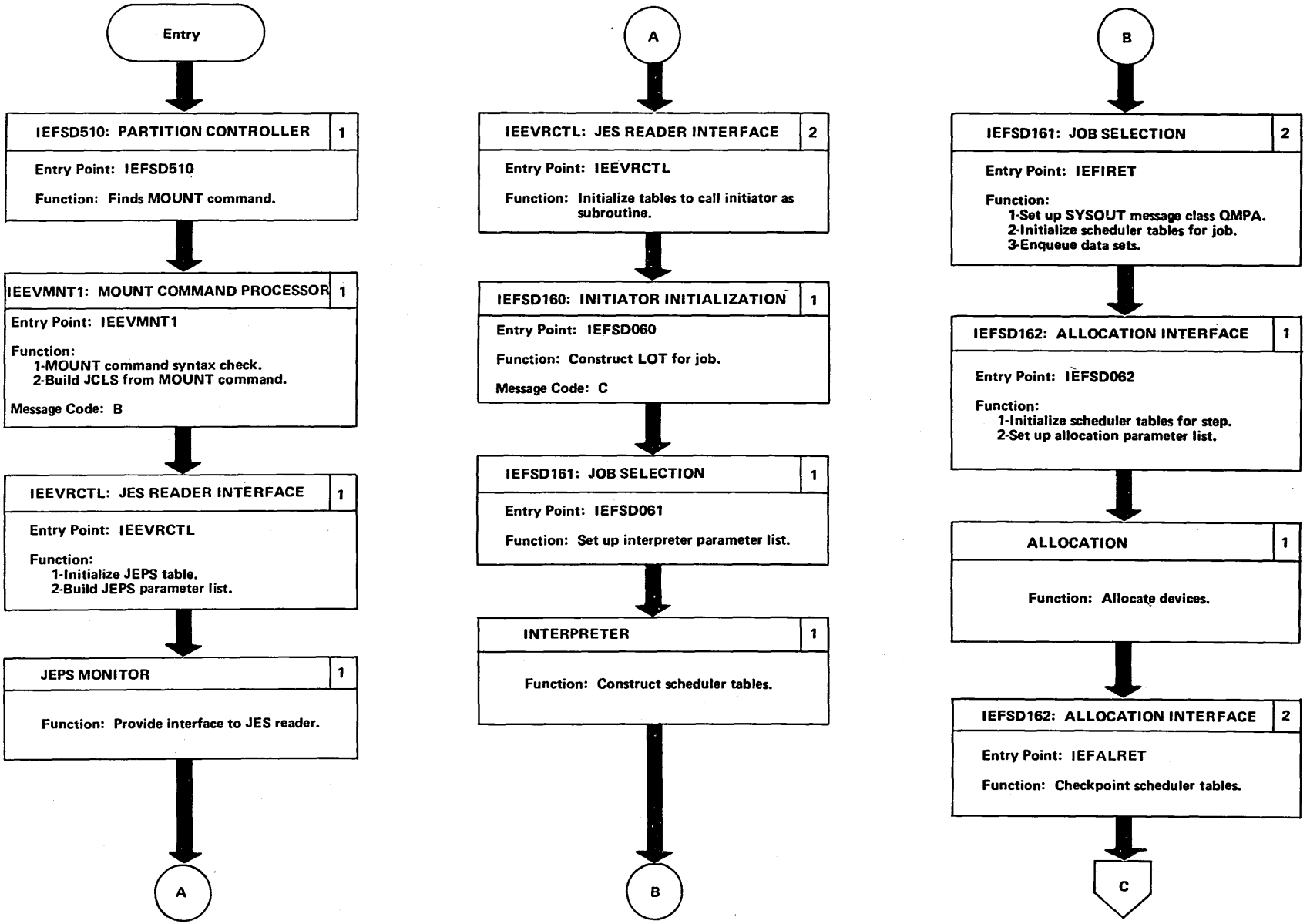
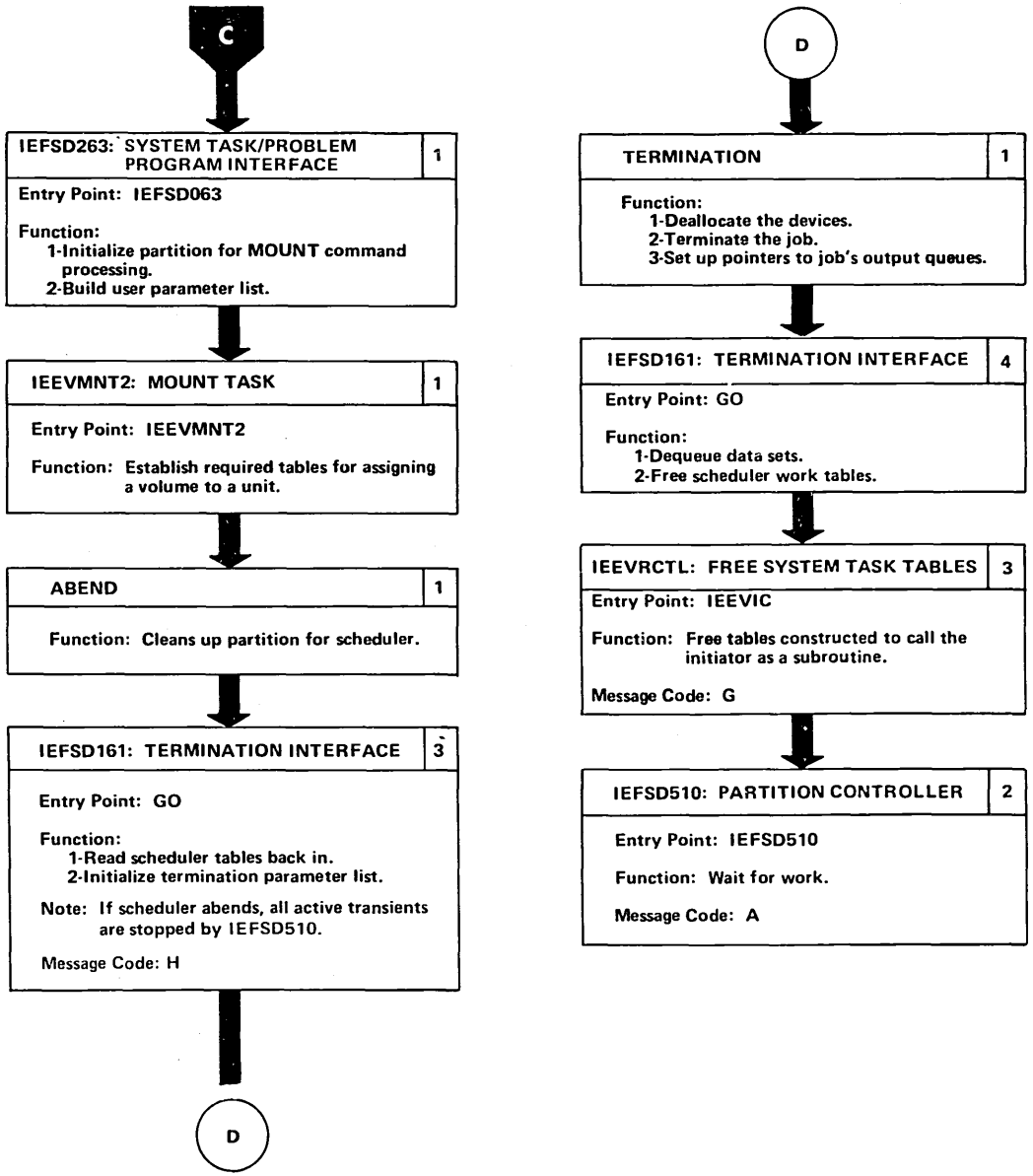


Figure 3-12. MOUNT Command (Initiator Processing) (Part 2 of 2)



Messages Code	Number	Reason
A	IEF005E	Partition waiting for work
B	IEE307I	Delimiter error
	IEE308I	Length error
	IEE309I	Unidentifiable keyword
	IEE311I	Parameter missing
C	IEF427I	Insufficient queue space
	IEF203I	Too many operands
	IEF204I	Too many classes
	IEF205I	Invalid operand
	IEF206I	No classes specified
	IEF016I	Invalid unit
	IEF017I	SWADS OPEN unsuccessful
G	IEE191I	DD entry missing
	IEE192I	Invalid PARM field format
	IEE122I	JCL error
	IEE132I	Device allocation error
	IEE121I	Job queue I/O error
	IEF002I	Partition not specified
H	IEF866E	Scheduler abend, completion code =

Figure 3-13. Interpreter Interconnection Module Diagram (Part 1 of 4)

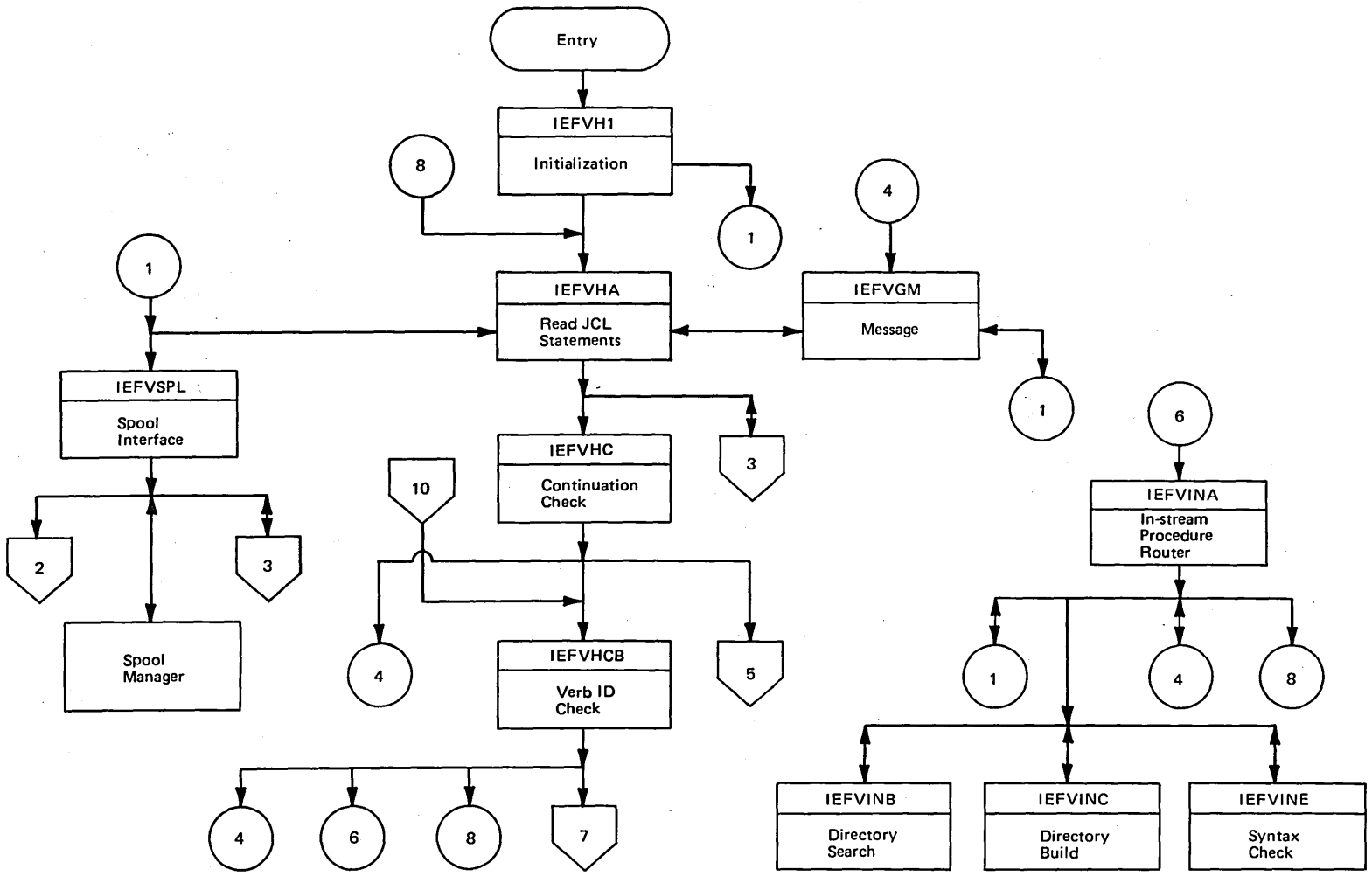


Figure 3-13. Interpreter Interconnection Module Diagram (Part 2 of 4)

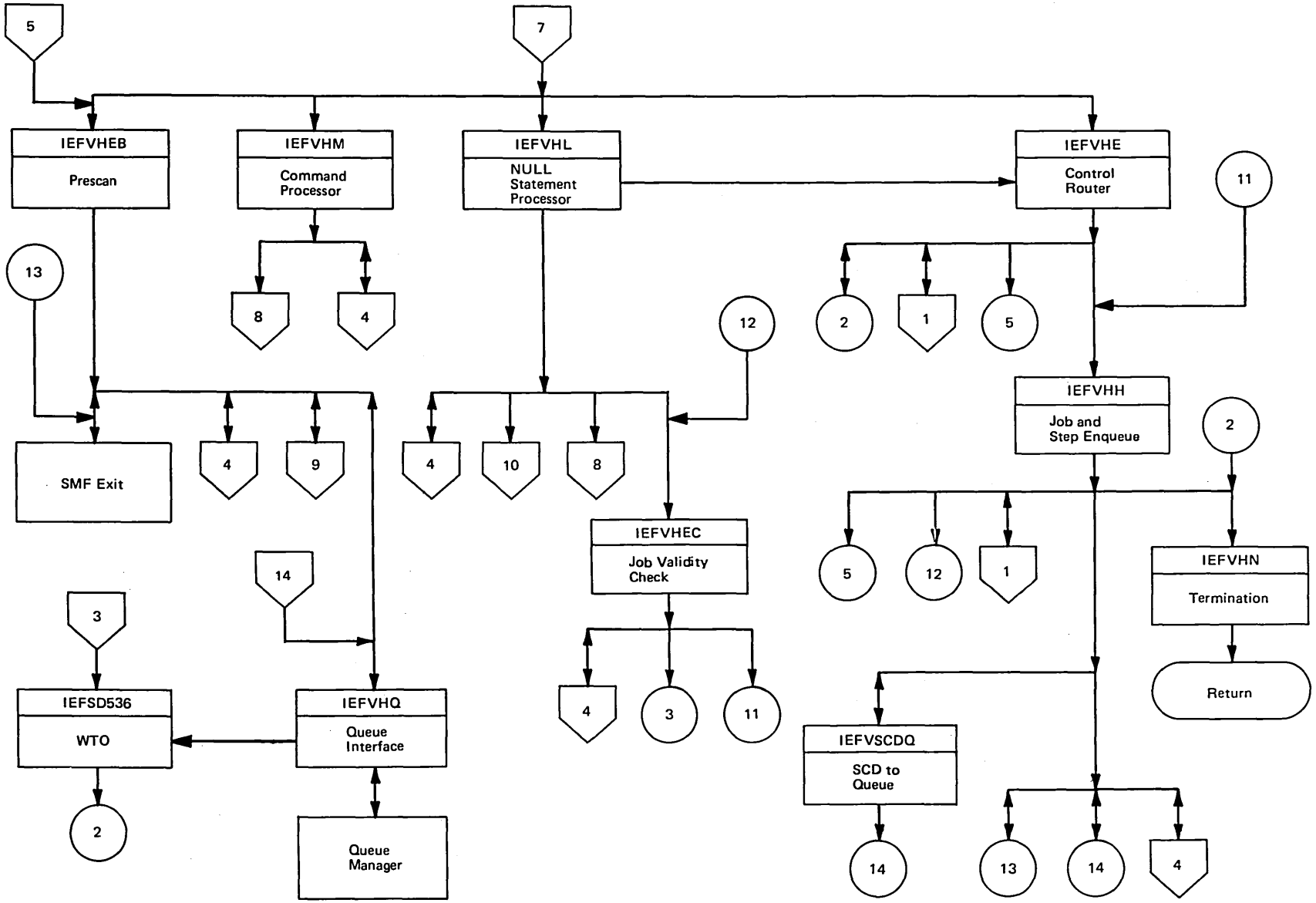


Figure 3-13. Interpreter Interconnection Module Diagram (Part 3 of 4)

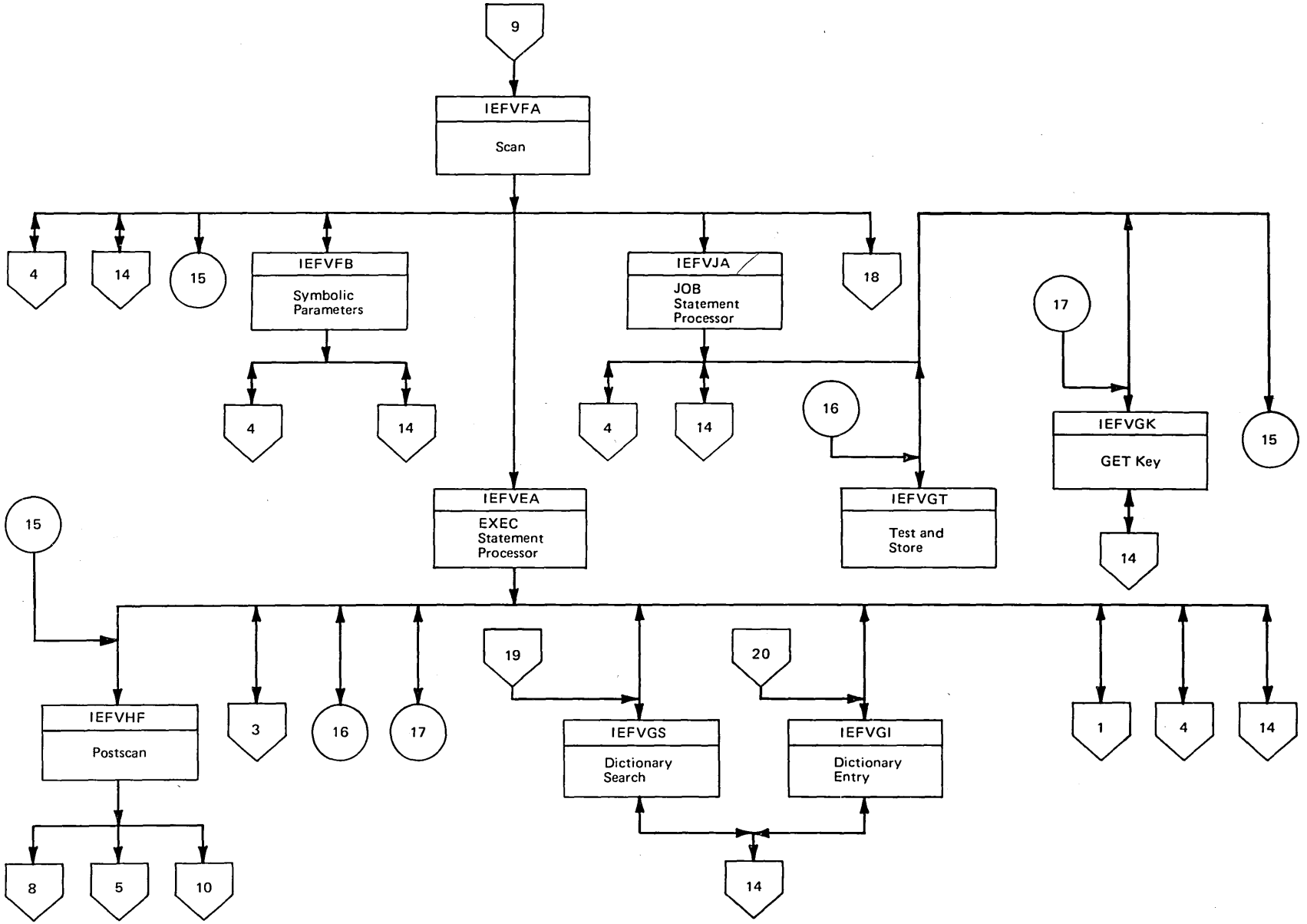


Figure 3-13. Interpreter Interconnection Module Diagram (Part 4 of 4)

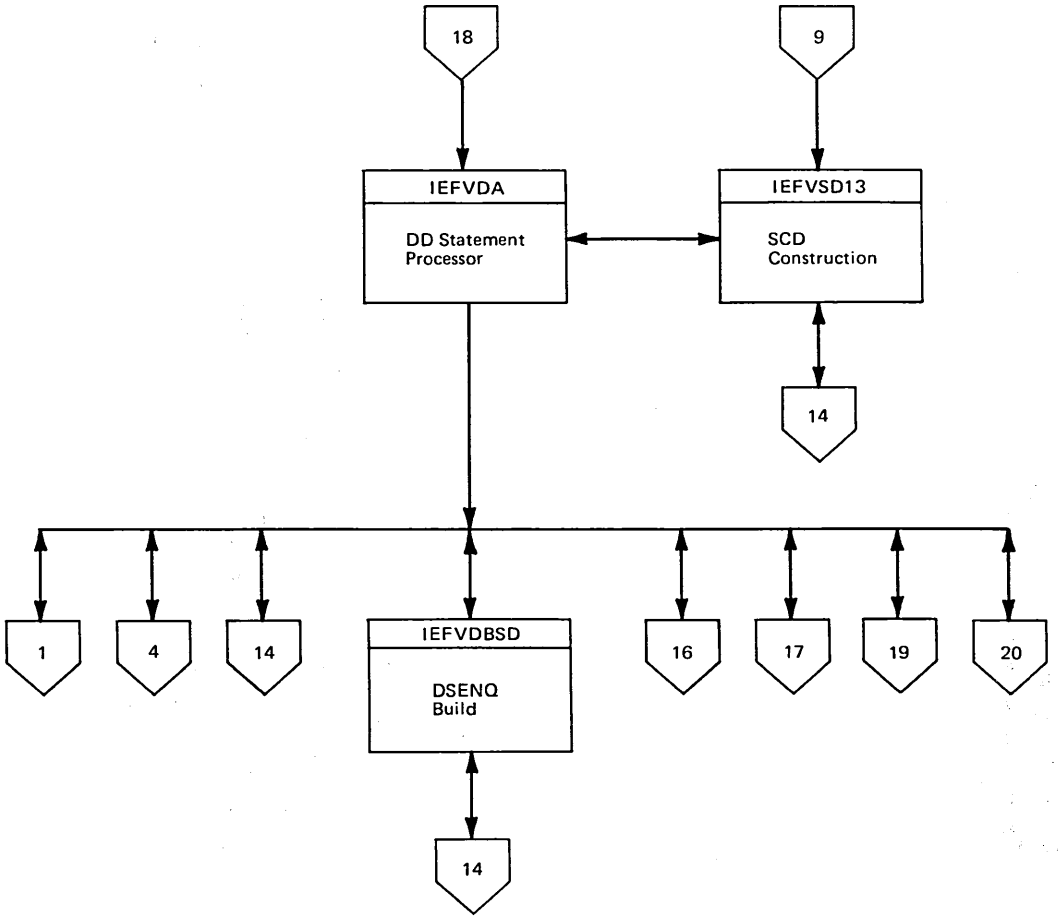
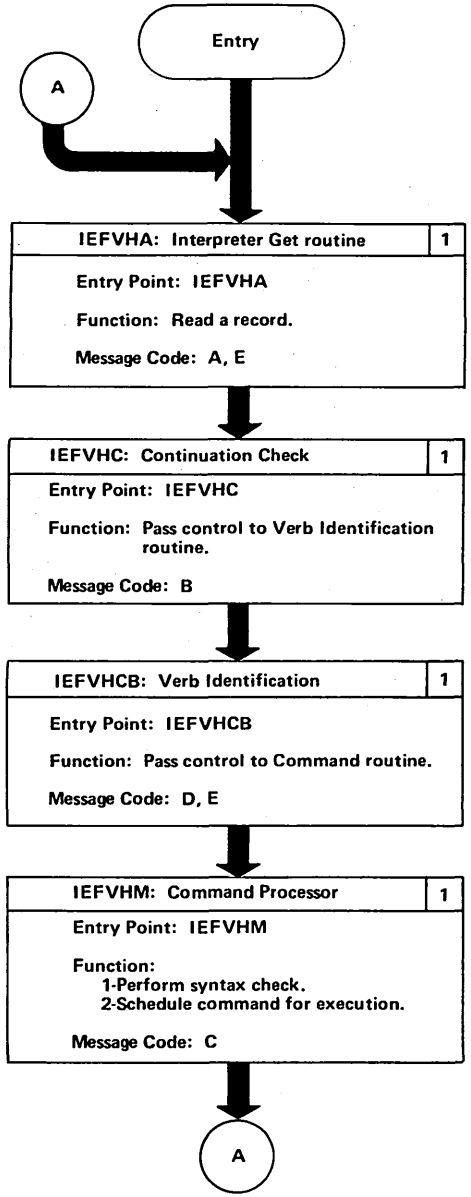
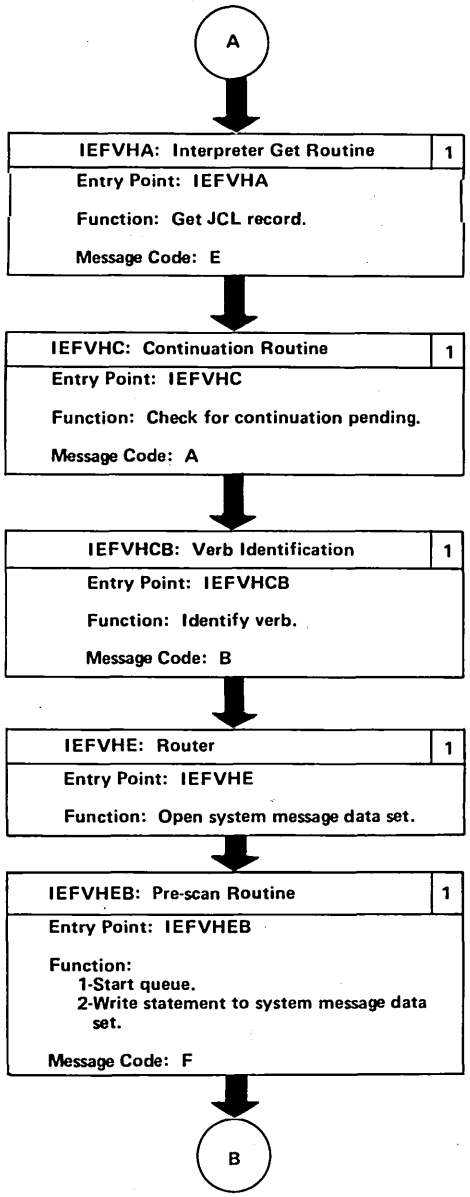


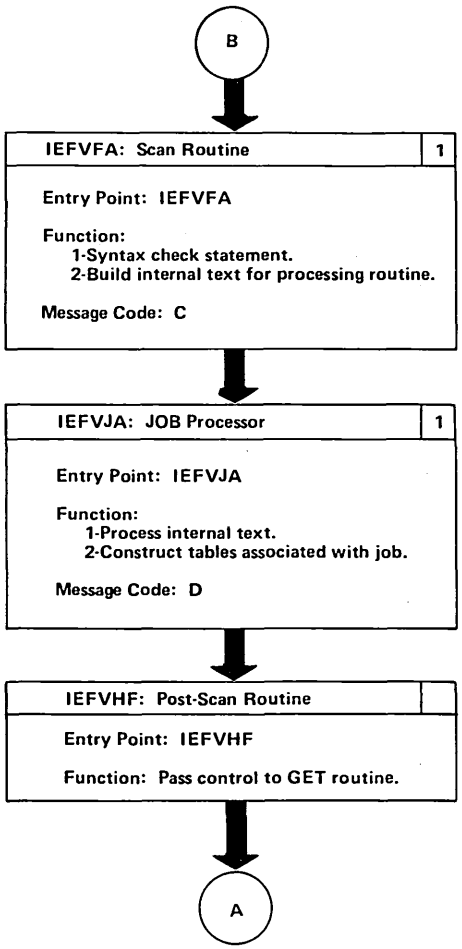
Figure 3-14. Processing Commands within a Job



Messages Code	Number	Reason
A	IEF417I IEF601I	Proclib device I/O error Invalid statement
B	IEF621I	No continuation
C	IEF165I	Command
D	IEF605I IEF601I IEF658I IEF660I IEF668I	Unidentified operand Invalid statement PROC out of place Misplaced SYSCHK DD PEND out of place
E	IEF610I	No step in procedure

Figure 3-15. Processing a JOB Statement





Messages Code	Number	Reason
A	IEF621I	No continuation
B	IEF605I	Unidentified operand
	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF658I	PROC out of place
	IEF660I	Misplaced SYSCHK DD
	IEF668I	PEND out of place
C	IEF627I	Incorrect use of ampersand
	IEF626I	Incorrect use of plus sign
	IEF624I	Incorrect use of period
	IEF625I	Incorrect use of left parentheses
	IEF623I	Invalid characters
	IEF622I	Unbalanced parentheses
	IEF618I	Invalid delimiter
	IEF616I	Invalid sublist
	IEF652I	Mutually exclusive keywords
	IEF651I	Incorrect use of hyphen
	IEF650I	Incorrect use of slash
	IEF640I	Too many positional parameters
	IEF636I	Jobname missing
	IEF632I	Format error
	IEF630I	Unidentified keyword
	IEF628I	Incorrect use of asterisk
	IEF601I	Invalid statement
	IEF629I	Incorrect use of apostrophe
D	IEF634I	Account no. missing
	IEF633I	Programmer name missing
	IEF632I	Format error
	IEF646I	Parameter missing
	IEF642I	Parameter too long
	IEF639I	Invalid class
	IEF637I	Account field too long
E	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF417I	Proclib device I/O error
F	IEF611I	Overridden step not found
	IEF659I	Misplaced SYSCHK DD

Figure 3-16. Processing an EXEC Statement (Part 1 of 2)

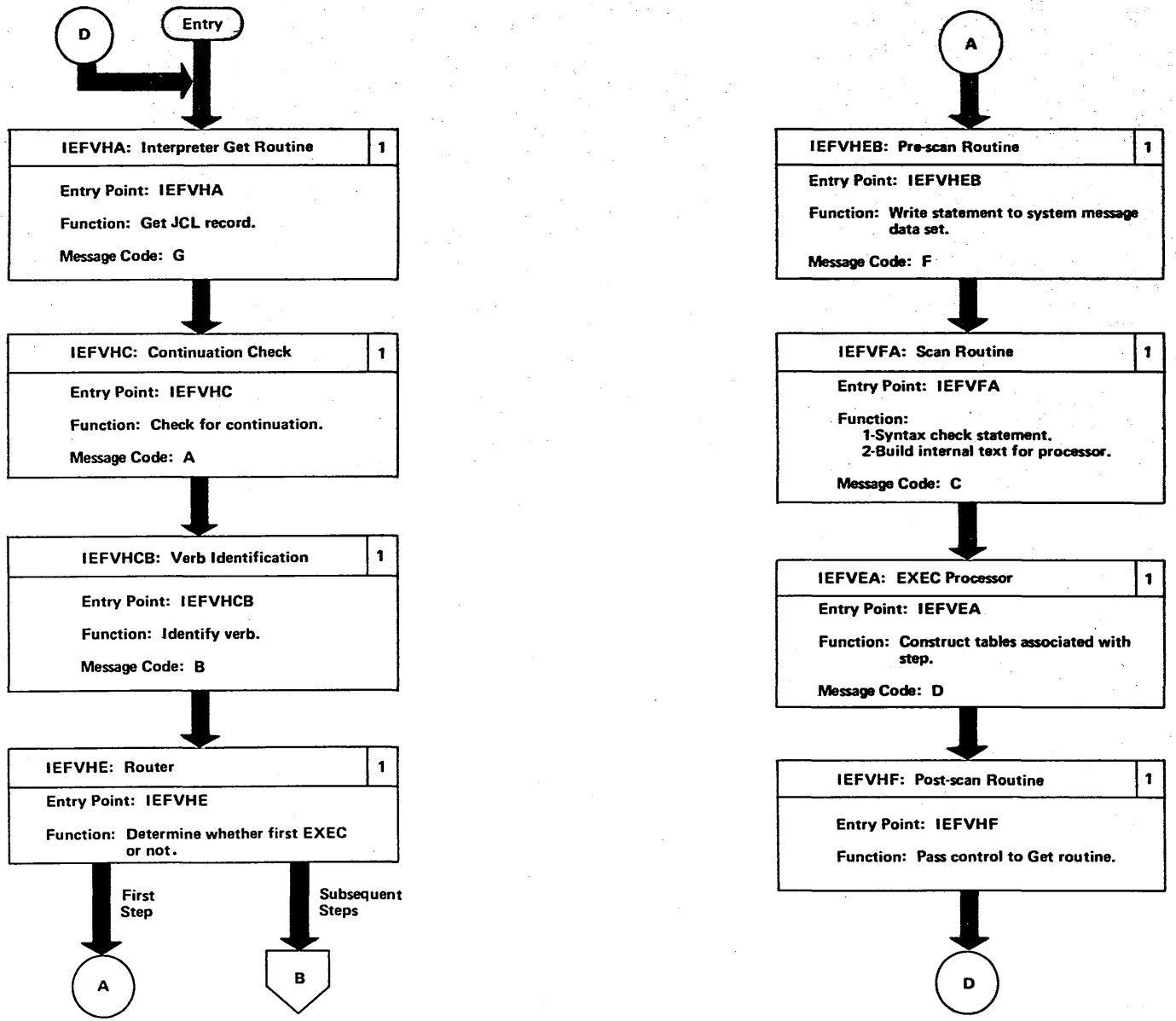
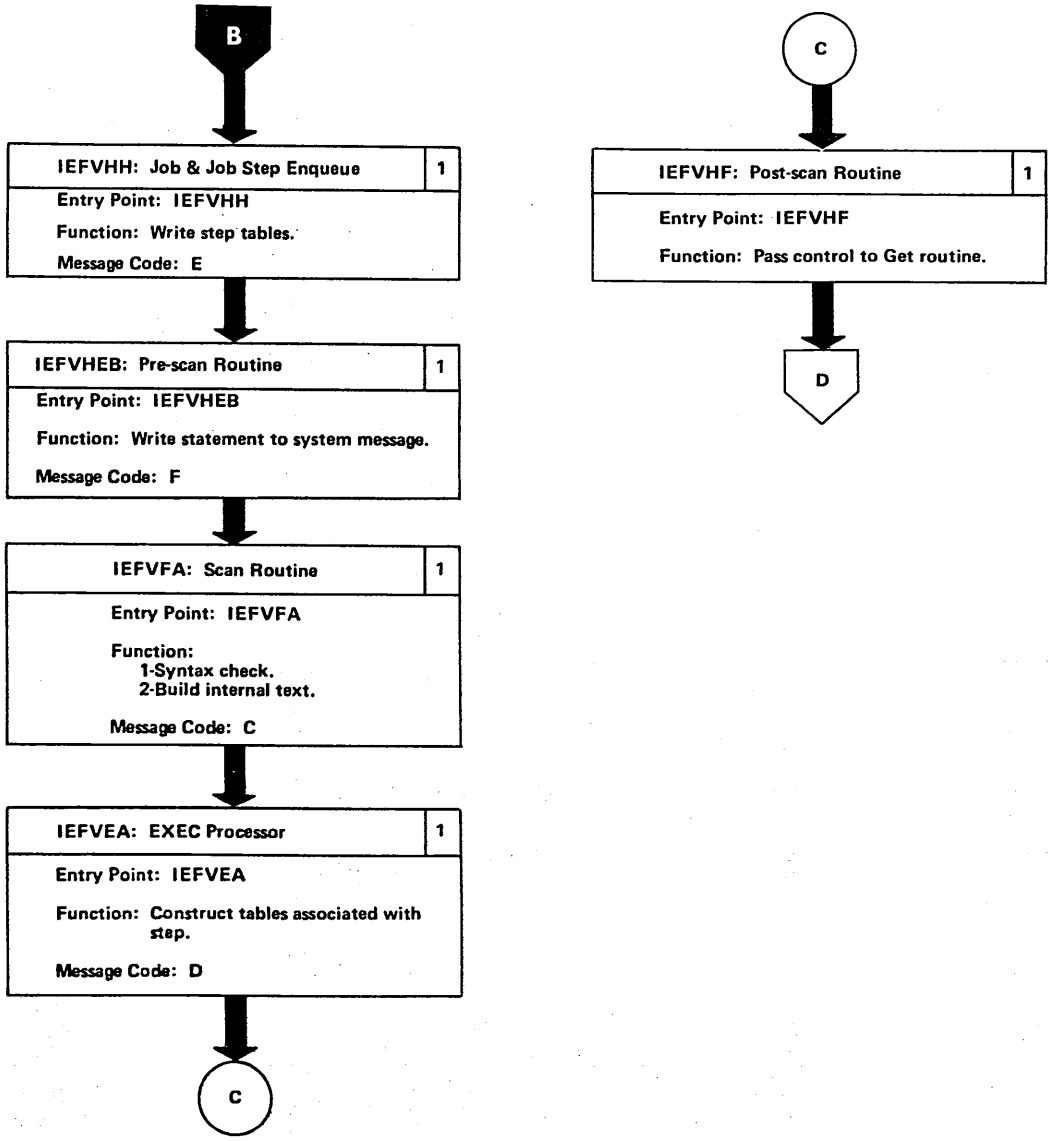


Figure 3-16. Processing an EXEC Statement (Part 2 of 2)



Messages Code	Number	Reason
A	IEF621I	No continuation
B	IEF605I	Unidentified operation
	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF658I	PROC out of place
	IEF660I	Misplaced SYSCHK DD
	IEF668I	PEND out of place
C	IEF601I	Invalid statement
	IEF629I	Incorrect use of apostrope
	IEF627I	Incorrect use of ampersand
	IEF626I	Incorrect use of plus sign
	IEF624I	Incorrect use of period
	IEF625I	Incorrect use of left parenthesis
	IEF623I	Invalid characters
	IEF622I	Unbalanced parentheses
	IEF618I	Invalid delimiter
	IEF516I	Invalid sublist
	IEF652I	Mutually exclusive keywords
	IEF651I	Incorrect use of hyphen
	IEF650I	Incorrect use of slash
	IEF640I	Too many positional parameters
	IEF635I	Job name missing
	IEF632I	Format error
	IEF630I	Unidentified keyword
	IEF628I	Incorrect use of asterisk
D	IEF612I	Procedure not found
	IEF613I	Procedure within a procedure
	IEF614I	Proclib I/O error
	IEF615I	Stepname too long
	IEF632I	Format error
	IEF646I	Parameter missing
	IEF642I	Parameter too long
	IEF645I	Invalid reference
	IEF639I	Invalid class
	IEF638I	Numeric too large
	IEF637I	Account file too long
	IEF671I	Misplaced JOBCAT DD statement
E	IEF657I	Undefined sysbolic parameter
	IEF629I	Step not found
F	IEF611I	Overriden step not found
	IEF659I	Misplaced SYSCHK DD
G	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF417I	Proclib I/O error

Figure 3-17. Processing a DD Statement (Part 1 of 2)

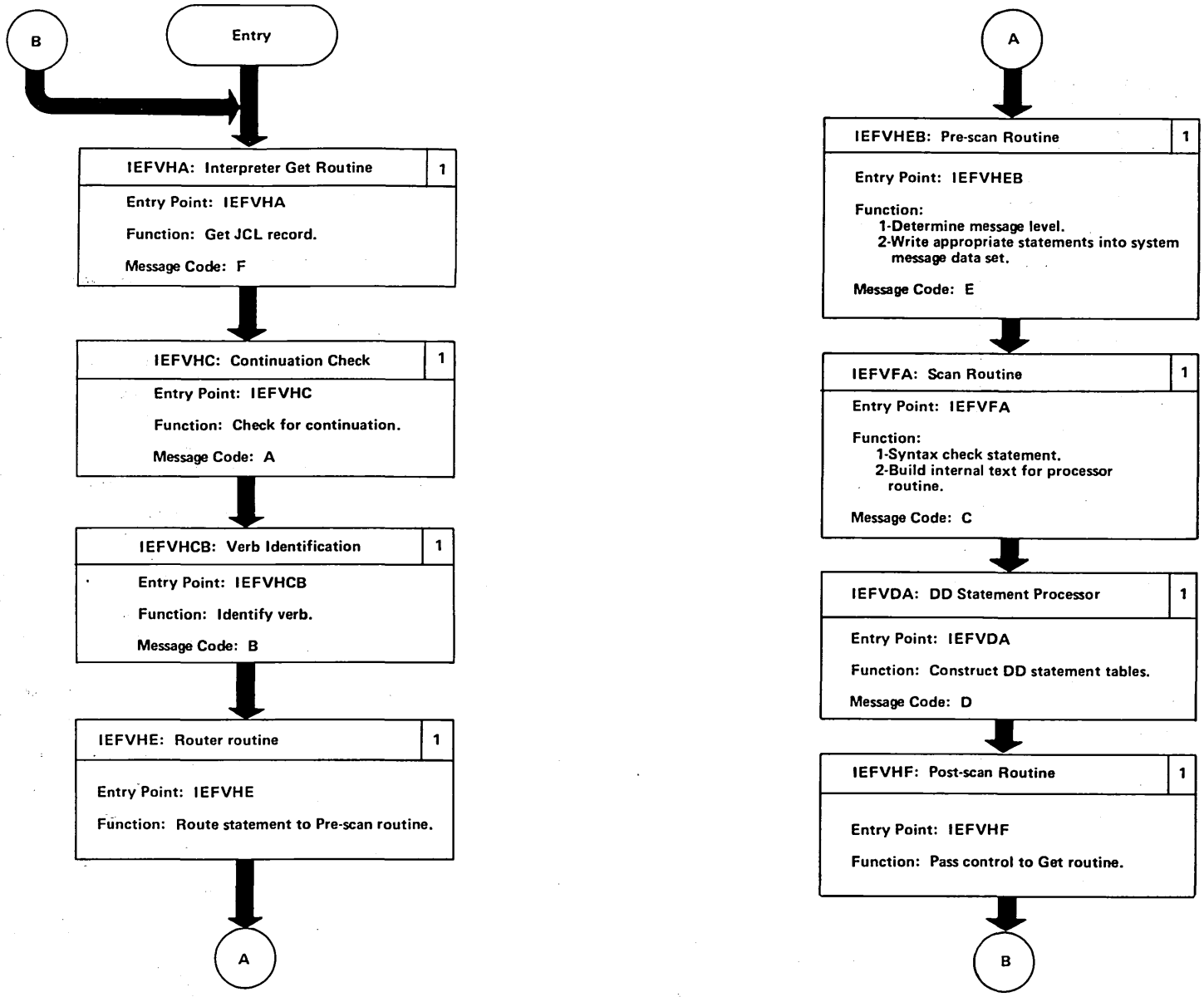
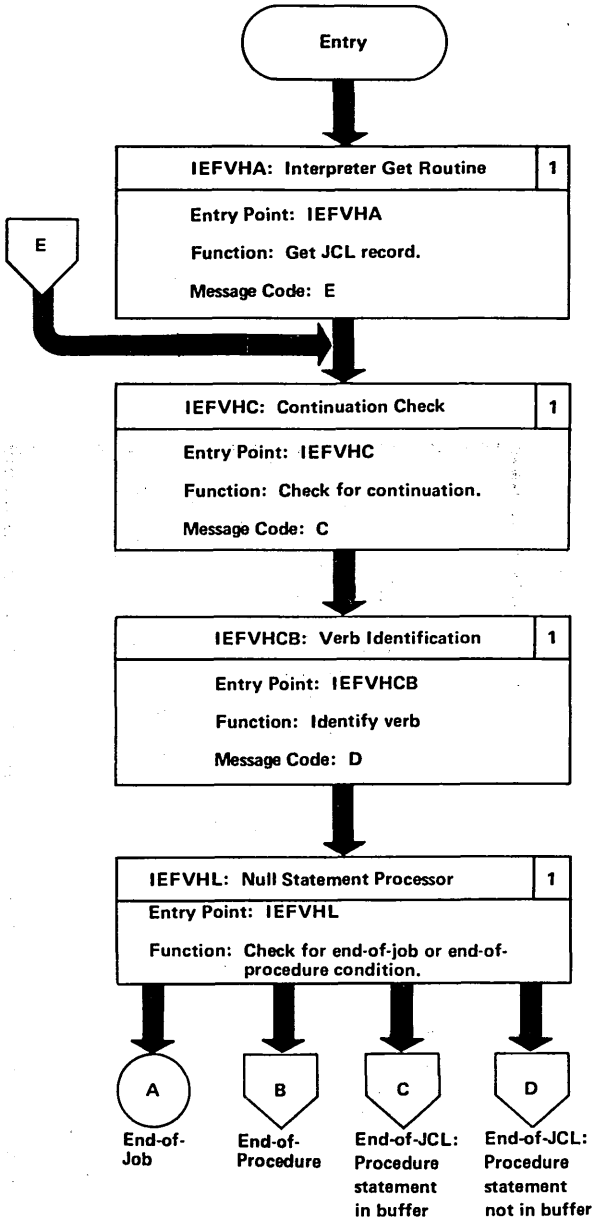


Figure 3-17. Processing a DD Statement (Part 2 of 2)

Messages Code	Number	Reason
A	IEF621I	No continuation
B	IEF605I	Unidentified Operation
	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF658I	PROC out of place
	IEF660I	Misplaced SYSCHK DD
	IEF668I	PEND out of place
C	IEF627I	Incorrect use of ampersand
	IEF626I	Incorrect use of plus sign
	IEF624I	Incorrect use of period
	IEF625I	Incorrect use of left parenthesis
	IEF623I	Invalid characters
	IEF622I	Unbalanced parentheses
	IEF618I	Invalid delimiter
	IEF616I	Invalid sublist
	IEF652I	Mutually exclusive keywords
	IEF651I	Incorrect use of hyphen
	IEF650I	Incorrect use of slash
	IEF640I	Too many positional parameters
	IEF635I	Jobname missing
	IEF632I	Format error
	IEF630I	Unidentified keyword
	IEF628I	Incorrect use of asterisk
	IEF601I	Invalid statement
	IEF629I	Incorrect use of apostrophe

Messages Code	Number	Reason
D	IEF042I	Invalid numerical value
	IEF517I	SYSIN parameters not compatible-default values used
	IEF670I	Value unassigned
	IEF669I	Invalid forward reference
	IEF654I	Multiple ddnames
	IEF649I	Too many DD statements
	IEF648I	Invalid disposition
	IEF647I	Invalid name
	IEF646I	Positional parameter missing
	IEF645I	Invalid referback
	IEF644I	Invalid numeric
	IEF643I	Unidentified parameters
	IEF642I	Parameter too long
	IEF640I	Too many positional parameters
	IEF636I	Misplaced JOBLIB statement
	IEF632I	Format error
	IEF631I	Too many ddnames
	IEF630I	Unidentified keyword
	IEF624I	Incorrect use of period
	IEF617I	ddname missing
	IEF606I	Misplaced DD statement
	IEF658I	PROC out of place
	IEF655I	Invalid ddname
	IEF671I	Misplaced JOBCAT DD statement
	IEF672I	Multiple STEPCAT DD statements in step
	IEF690I	Invalid userid
	IEF691I	Invalid routing of SYSOUT
	IEF692I	Invalid use of reserved parameter
E	IEF611I	Step not found
	IEF659I	Misplaced SYSCHK DD
F	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF417I	SYS1.PROCLIB

Figure 3-18. Processing a NULL Statement (Part 1 of 2)



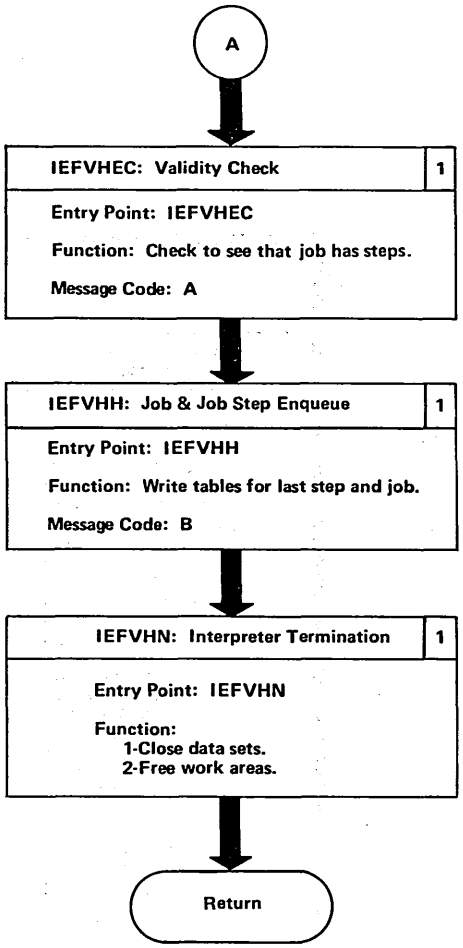
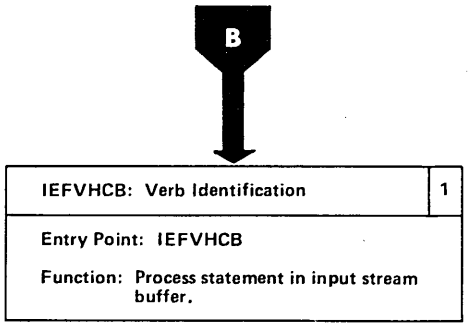
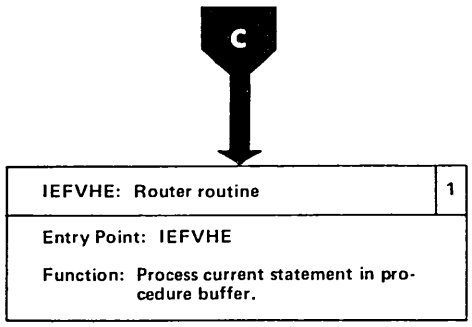
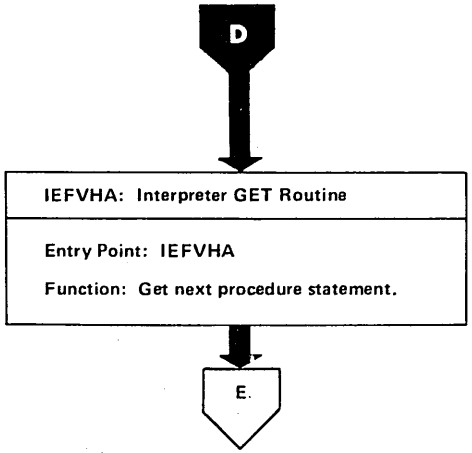


Figure 3-18. Processing a NULL Statement (Part 2 of 2)







Messages Code	Number	
A	IEF607I IEF661I IEF430I	Job has no steps No restart step No restart step
B	IEF657I IEF611I	Unidentified parameter Step not found
C	IEF621I	No continuation
D	IEF605I IEF601I IEF610I IEF658I IEF660I IEF668I	Unidentified operation Invalid statement No step in procedure PROC out of place Misplaced SYSCHK DD PEND out of place
E	IEF601I IEF610I IEF417I	Invalid statement In step in procedure Proclib I/O error

Figure 3-19. Processing Symbolic Parameters (Part 1 of 2)

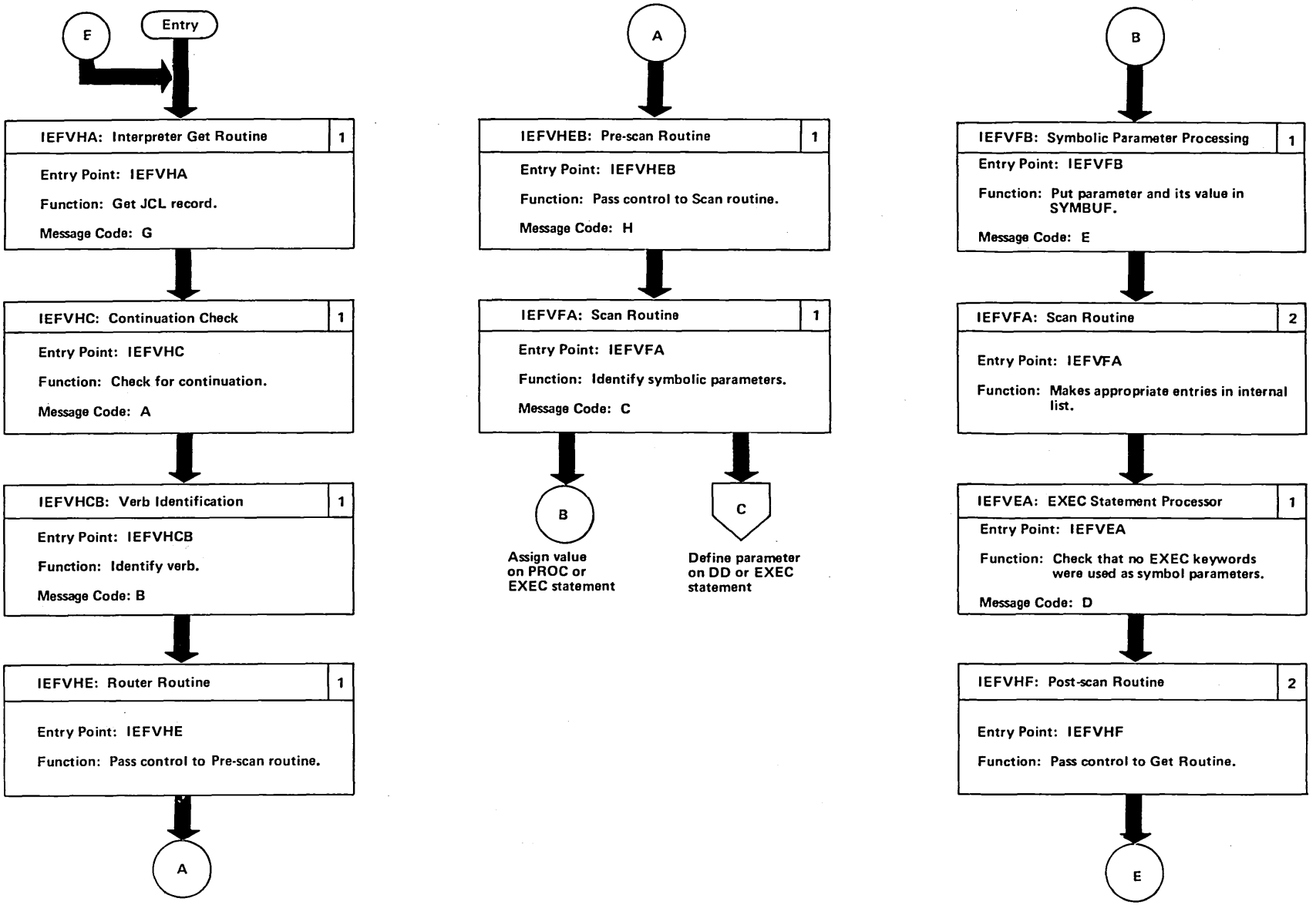
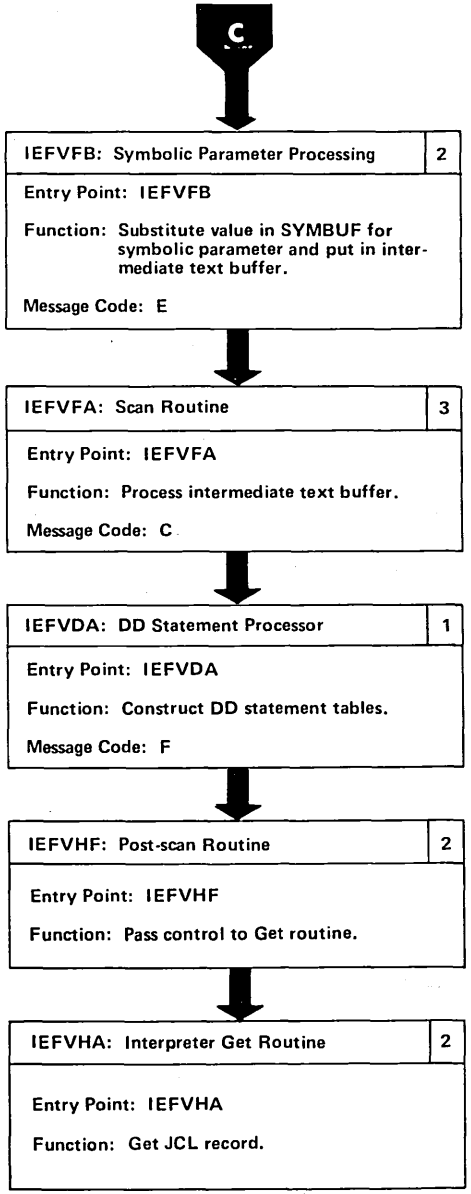


Figure 3-19. Processing Symbolic Parameters (Part 2 of 2)



Messages Code	Number	Reason
A	IEF621I	No continuation
B	IEF605I	Unidentified operation
	IEF601I	Invalid statement
	IEF660I	Misplaced SYSCHK DD
	IEF658I	Proc out of place
	IEF668I	PEND out of place
	IEF610I	No step in procedure
C	IEF080I	Parameter value ignored
	IEF601I	Invalid Statement
	IEF629I	Incorrect use of apostrophe
	IEF627I	Incorrect use of ampersand
	IEF626I	Incorrect use of plus sign
	IEF624I	Incorrect use of period
	IEF625I	Incorrect use of left parenthesis
	IEF623I	Invalid characters
	IEF622I	Unbalanced parentheses
	IEF618I	Invalid delimiter
	IEF616I	Invalid sublist
	IEF652I	Mutually exclusive keywords
	IEF651I	Incorrect use of hyphen
	IEF650I	Incorrect use of slash
	IEF640I	Too many positional parameters
	IEF635I	Jobname missing
	IEF632I	Format error
	IEF630I	Unidentified keyword
	IEF628I	Incorrect use of asterisk
D	IEF050I	Invalid start of system task
	IEF612I	Procedure not found
	IEF613I	Procedure within a procedure
	IEF614I	Proclib I/O error
	IEF615I	Stepname too long
	IEF632I	Format error
	IEF646I	Parameter missing
	IEF642I	Parameter too long
	IEF670I	No value assigned to symbolic parameter

Messages Code	Number	Reason
D (cont.)	IEF645I	Invalid reference
	IEF639I	Invalid class
	IEF638I	Numeric too large
	IEF637I	Account file too long
E	IEF653I	Substitution JCL
	IEF624I	Incorrect use of period
	IEF642I	Parameter too long
	IEF647I	Invalid character
	IEF630I	Unidentified keyword
	IEF623I	Illegal characters
	IEF618I	Invalid delimiter
F	IEF042I	Invlaid numerical value
	IEF517I	SYSIN parameters not compatible-default values used
	IEF670I	Value unassigned
	IEF669I	Invalid forward reference
	IEF654I	Multiple ddnames
	IEF649I	Too many DD stats
	IEF648I	Invalid disposition
	IEF647I	Invalid name
	IEF646I	Parameter missing
	IEF645I	Invalid referback
	IEF644I	Invalid numeric
	IEF643I	Unidentified parameter
	IEF642I	Parameter too long
	IEF640I	Too many parameters
	IEF636I	Misplaced JOBLIB statement
	IEF632I	Format error
	IEF631I	Too many ddnames
	IEF630I	Unidentified keyword
	IEF624I	Incorrect use of period
	IEF617I	ddname missing
	IEF606I	Misplaced DD stat
	IEF671I	Misplaced JOBCAT DD statement
	IEF672I	Multiple STEPCAT DD statements in step
	IEF690I	Invalid userid
	IEF691I	Invalid routing of SYSOUT
	IEF692I	Invalid use of reserved parameter
G	IEF610I	No step in procedure
	IEF601I	Invalid statement
	IEF417I	Proclib I/O error
H	IEF611I	Overridden step not found
	IEF659I	Misplaced SYSCHK DD

Figure 3-20. Processing In-Stream Procedures (Part 1 of 2)

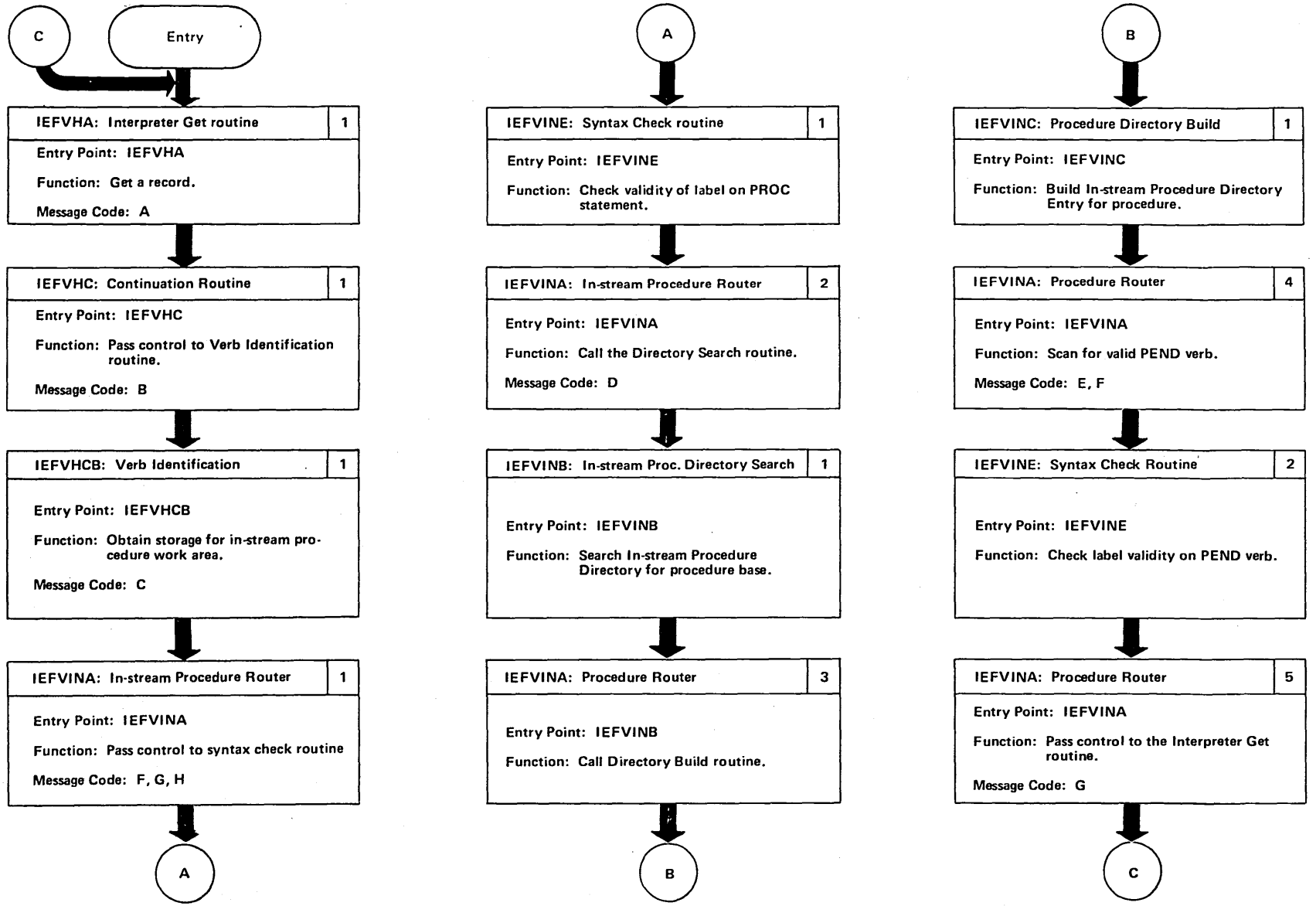
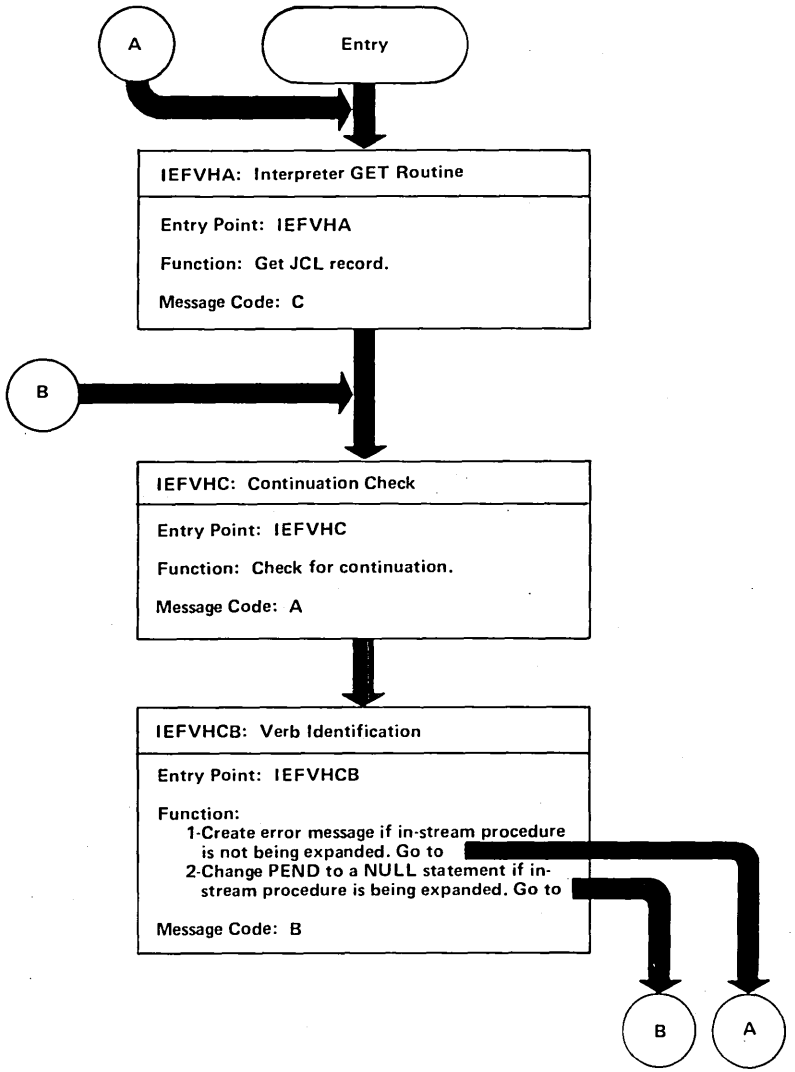


Figure 3-20. Processing in-Stream Procedures (Part 2 of 2)

Messages Code	Number	Reason
A	IEF610I IEF601I IEF417I	No step in procedure Invalid statement Proclib I/O error
B	IEF621I	No continuation
C	IEF605I IEF610I IEF658I IEF660I IEF668I IEF601I	Unidentified operation No EXEC statement Stmt out of place Missing SYSCHK DD PEND out of place Invalid statement
D	IEF663I	No procedure name
E	IEF665I	Too many instream procedures
F	IEF041I	Missing PEND verb
G	IEF662I	Invalid label
H	IEF663I IEF665I	Missing label Too many procedures

Figure 3-21. Processing the PEND Statement



Messages Code	Number	Reason
A	IEF621I	No continuation
B	IEF668I	PEND out of place
	IEF605I	Unidentified operation
	IEF610I	No EXEC statement
	IEF658I	Statement out of place
	IEF660I	Missing SYSCHK DD
	IEF601I	Invalid statement
C	IEF601I	Invalid statement
	IEF610I	No step in procedure
	IEF417I	PROCLIB I/O error

Figure 3-22. Allocation Interconnection Module Diagram (Part 1 of 3)

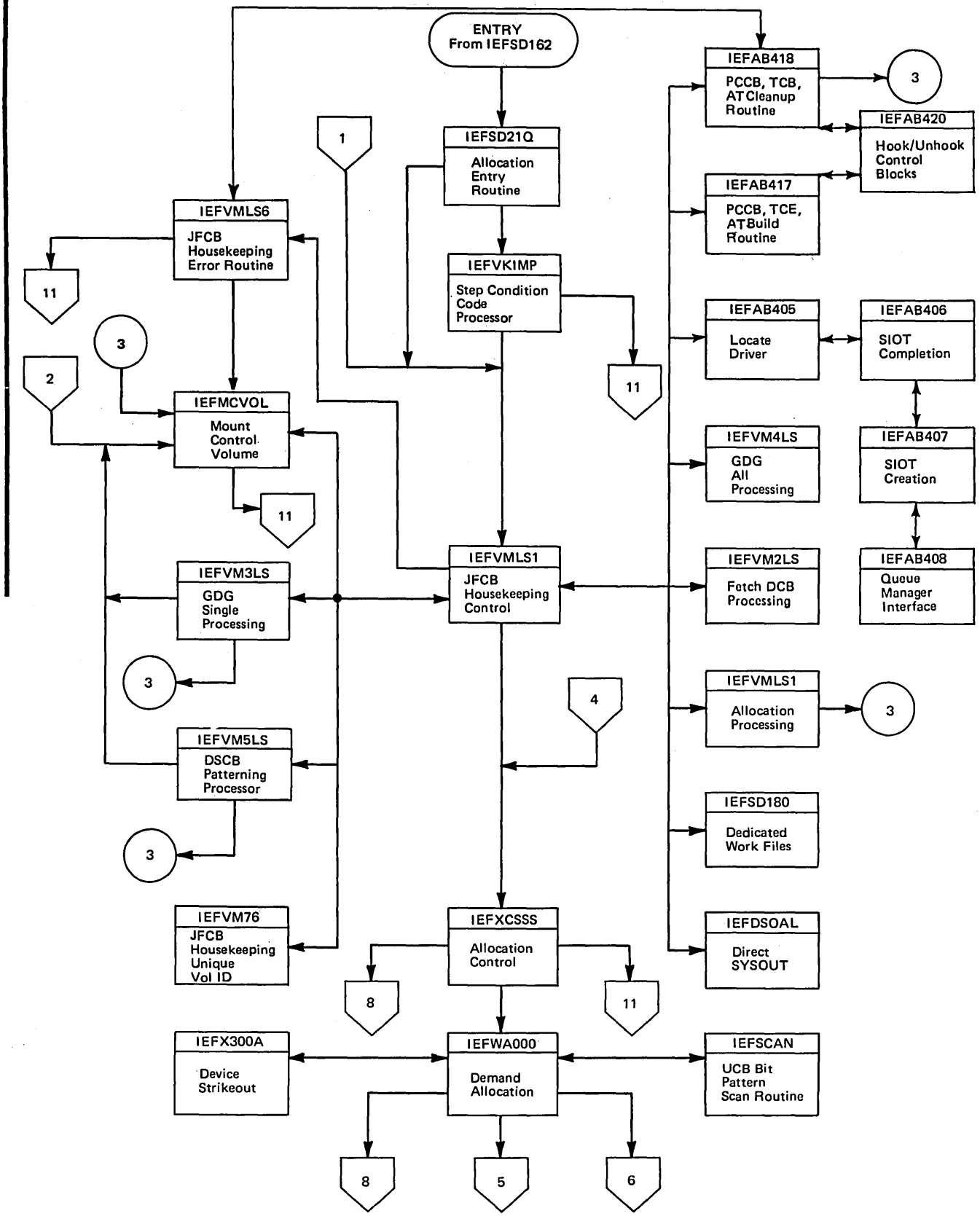


Figure 3-22. Allocation Interconnection Module Diagram (Part 2 of 3)

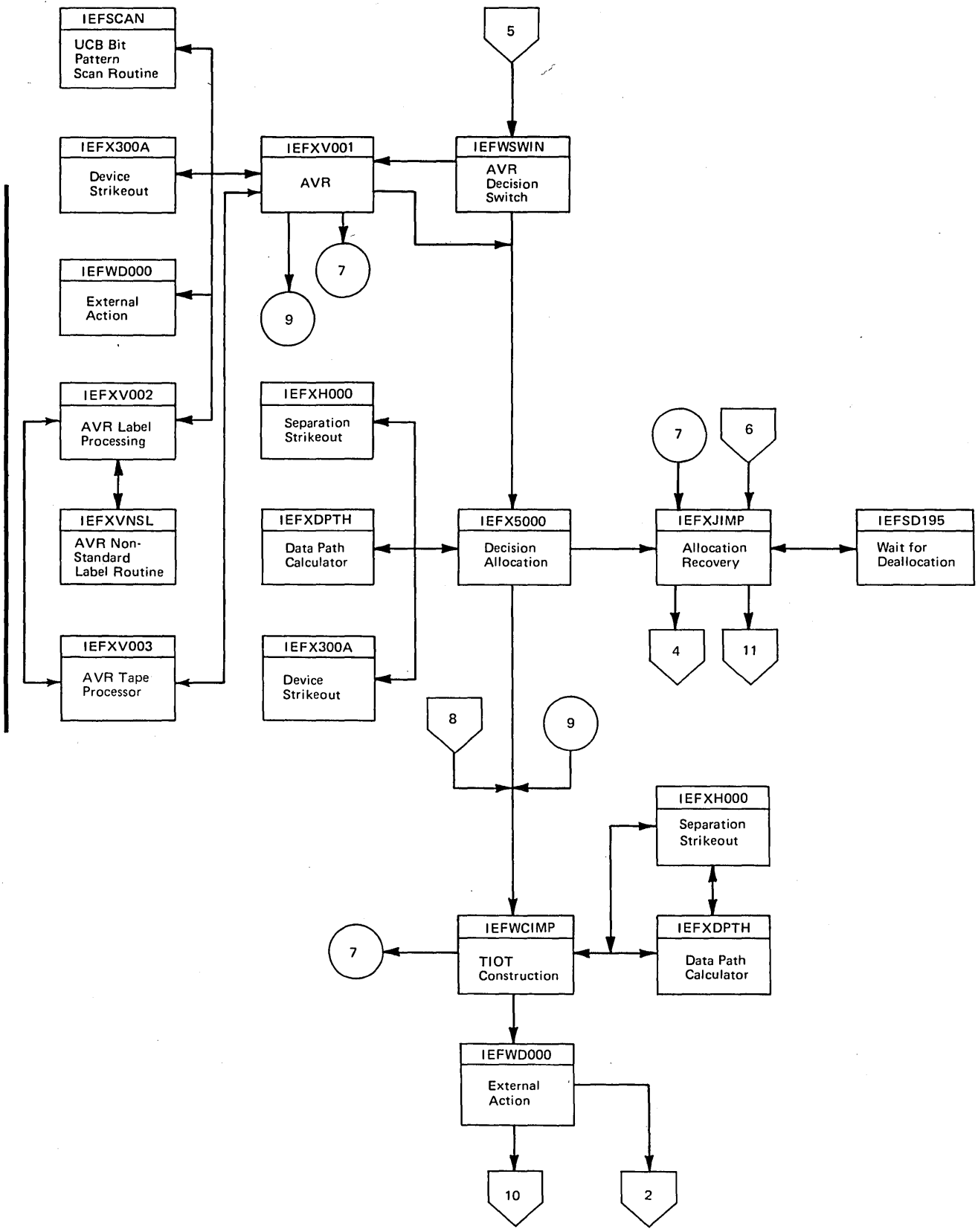


Figure 3-22. Allocation Interconnection Module Diagram (Part 3 of 3)

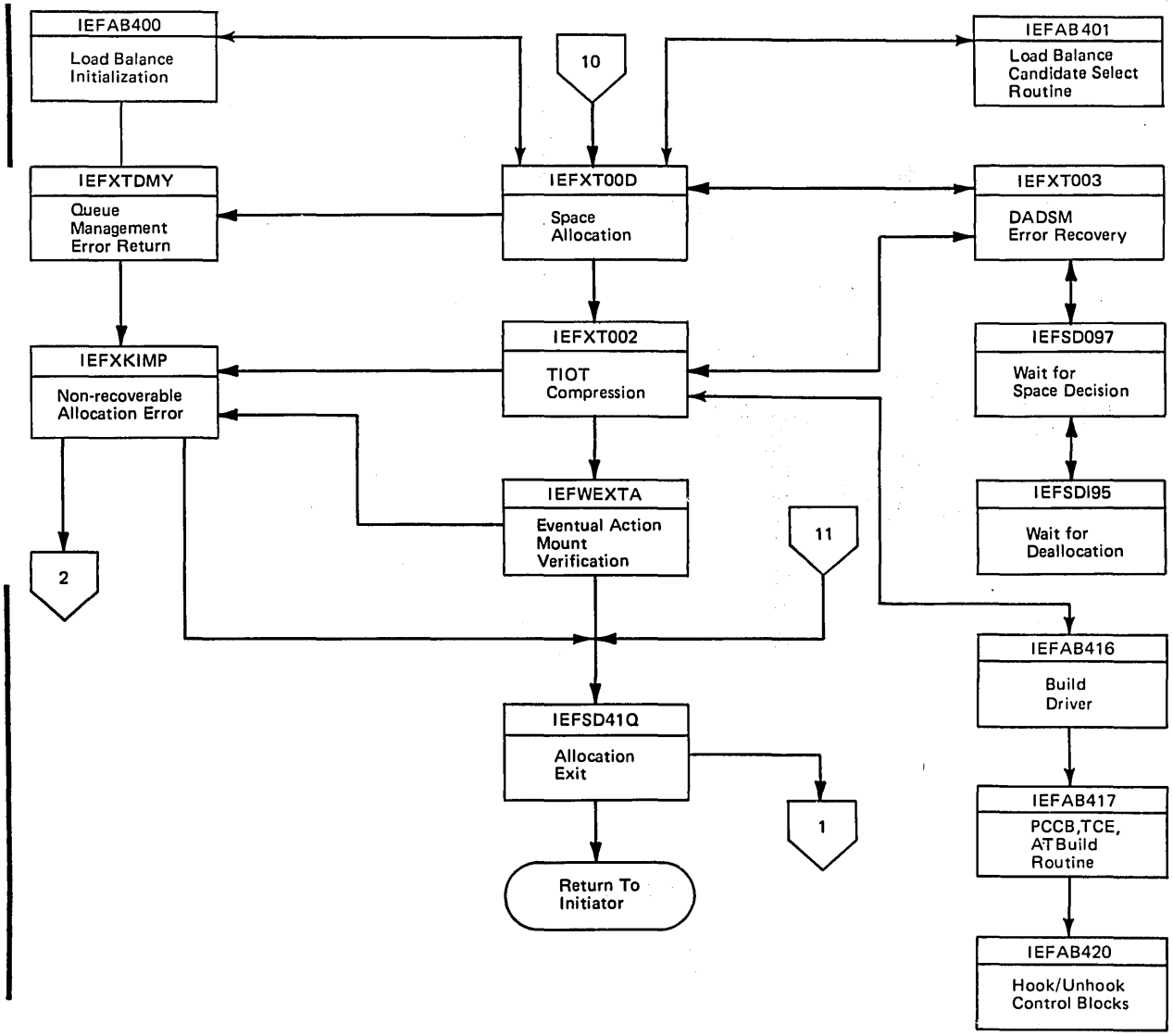


Figure 3-23. I/O Device Allocation Routines (Part 1 of 3)

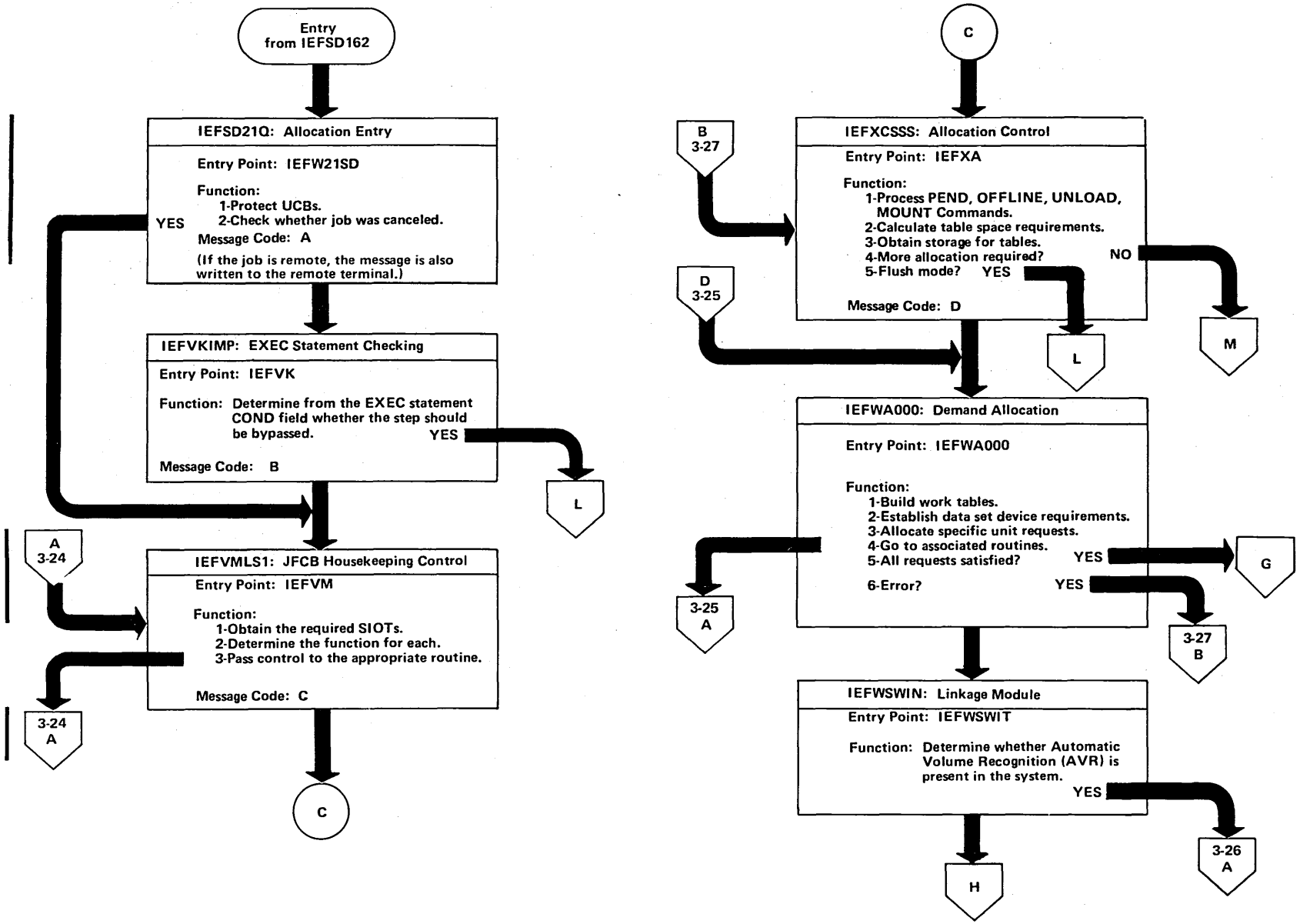


Figure 3-23. I/O Device Allocation Routines (Part 2 of 3)

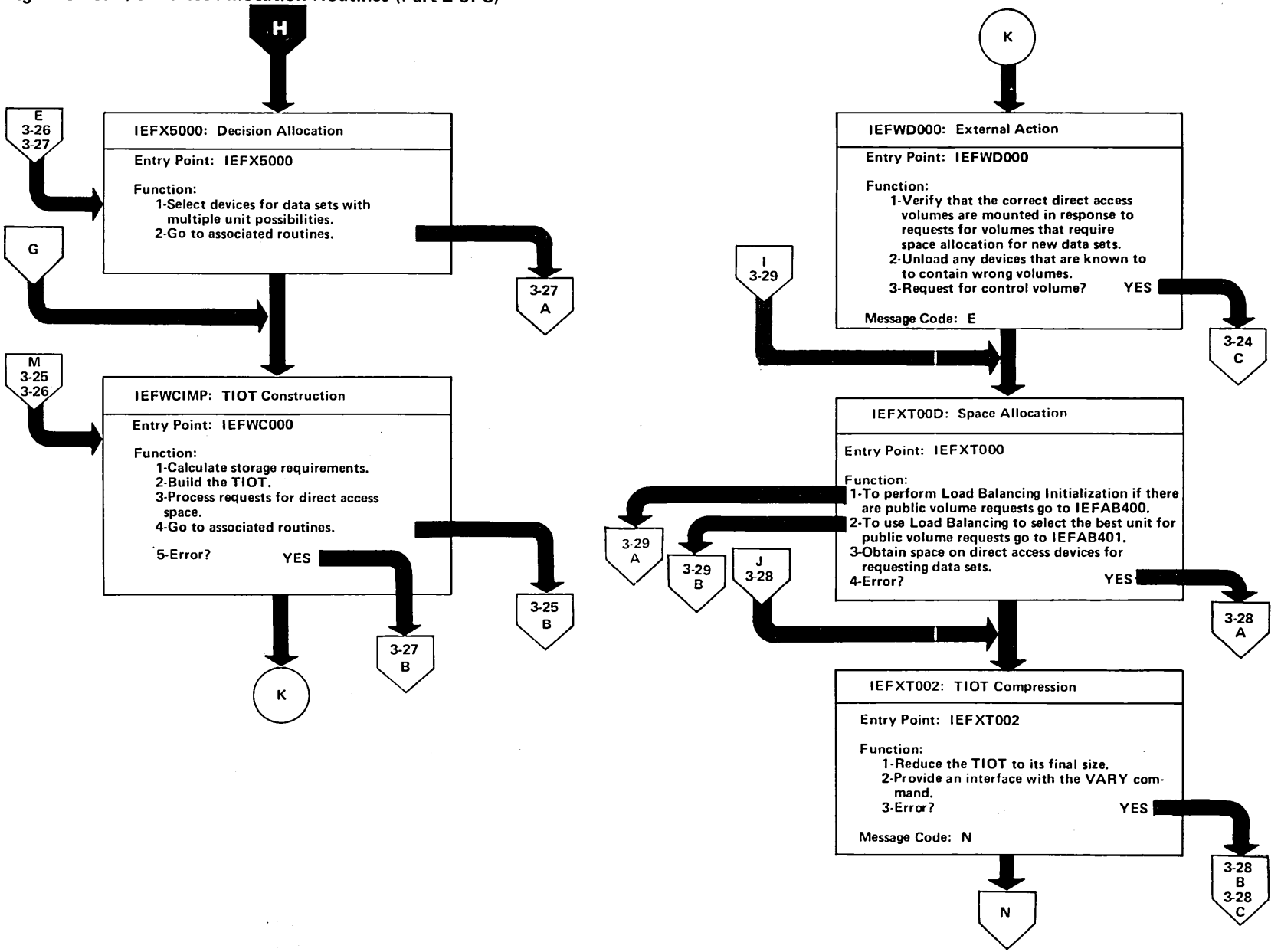
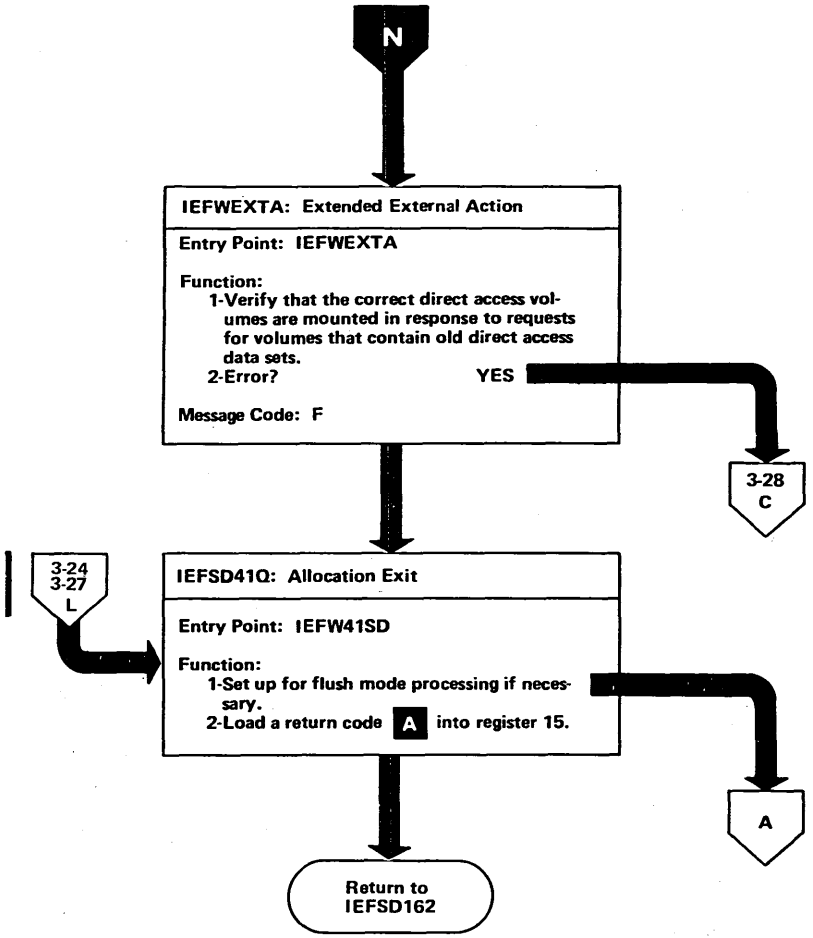


Figure 3-23. I/O Device Allocation Routines (Part 3 of 3)



A Code	Meaning
0	Successful Allocation
4	Unsuccessful Allocation

Figure 3-24 JFCB Housekeeping Routines (Part 1 of 2)

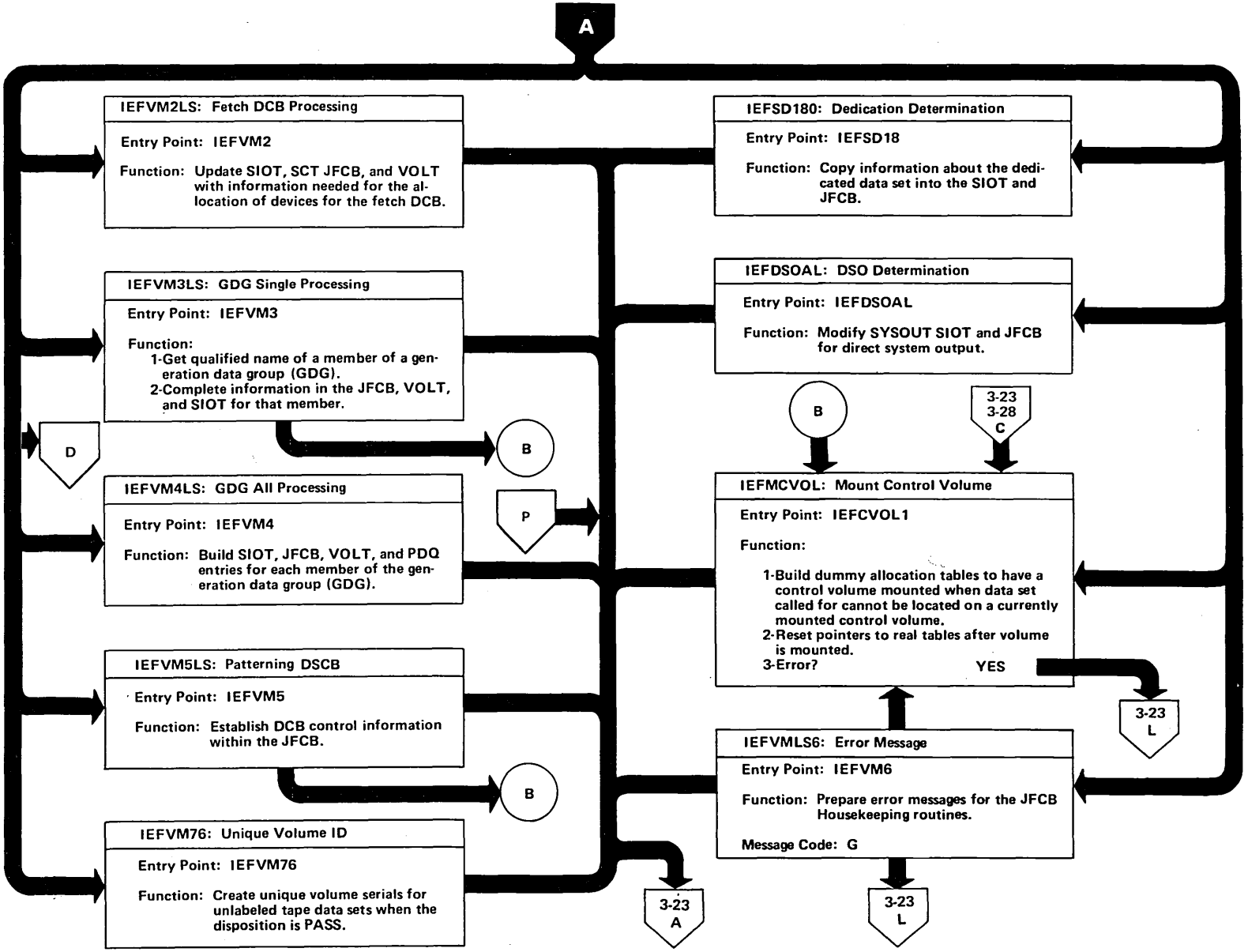


Figure 3-24 JFCB Housekeeping Routines (Part 2 of 2)

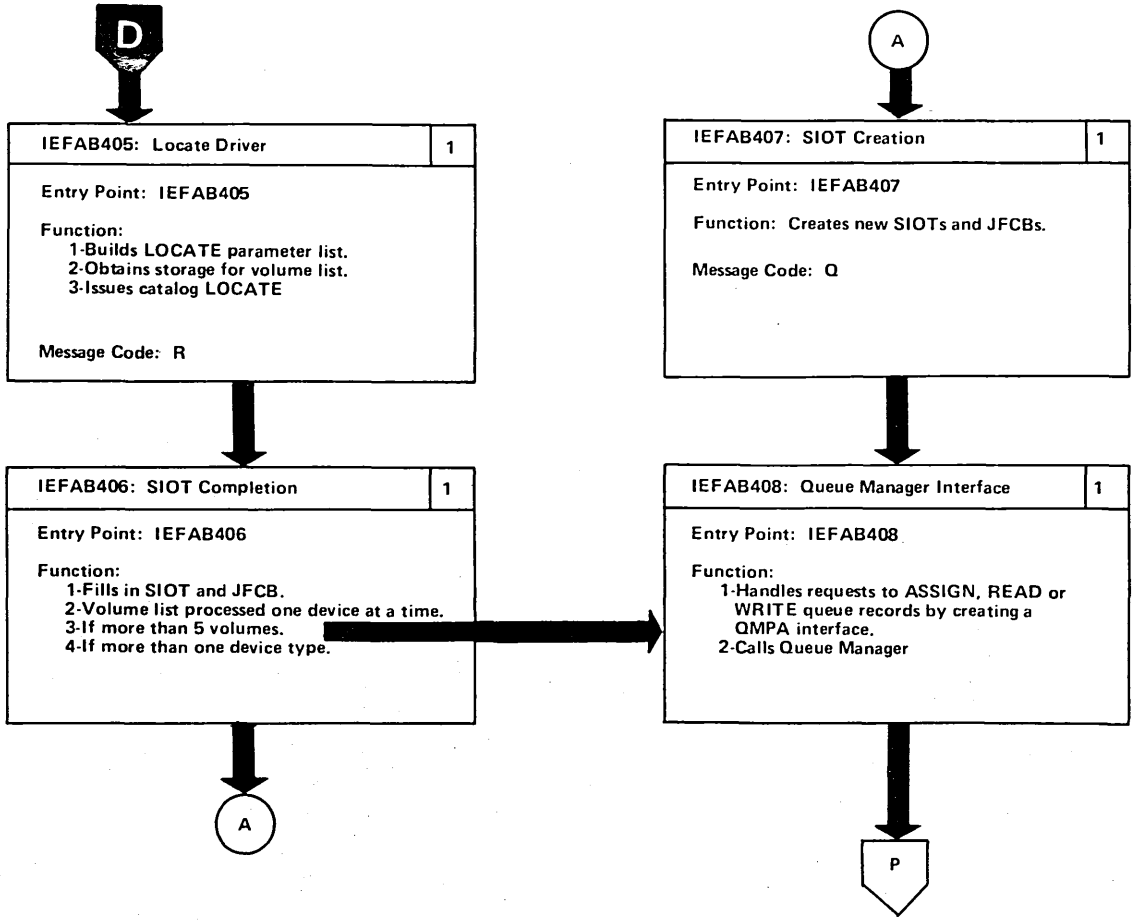


Figure 3-25. Demand Allocation and TIOT Construction

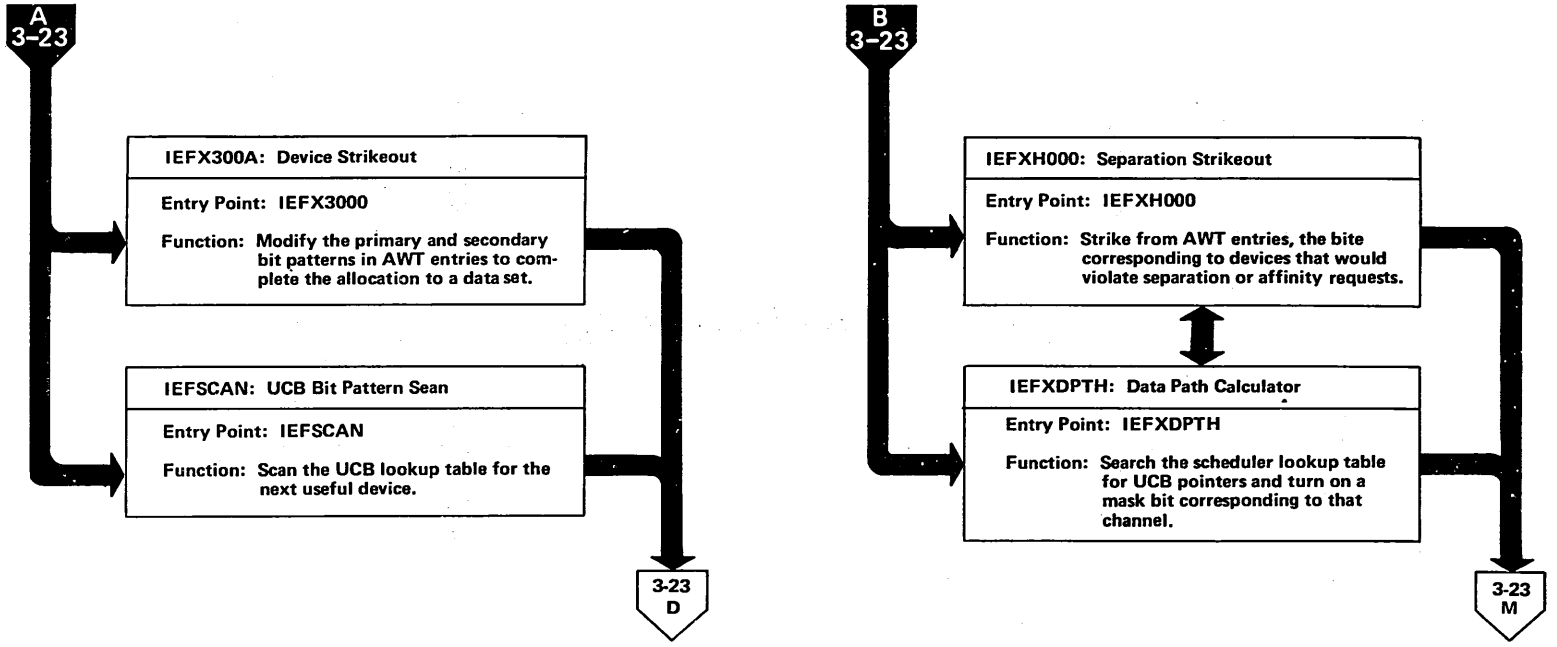


Figure 3-26. Automatic Volume Recognition (AVR) Routines

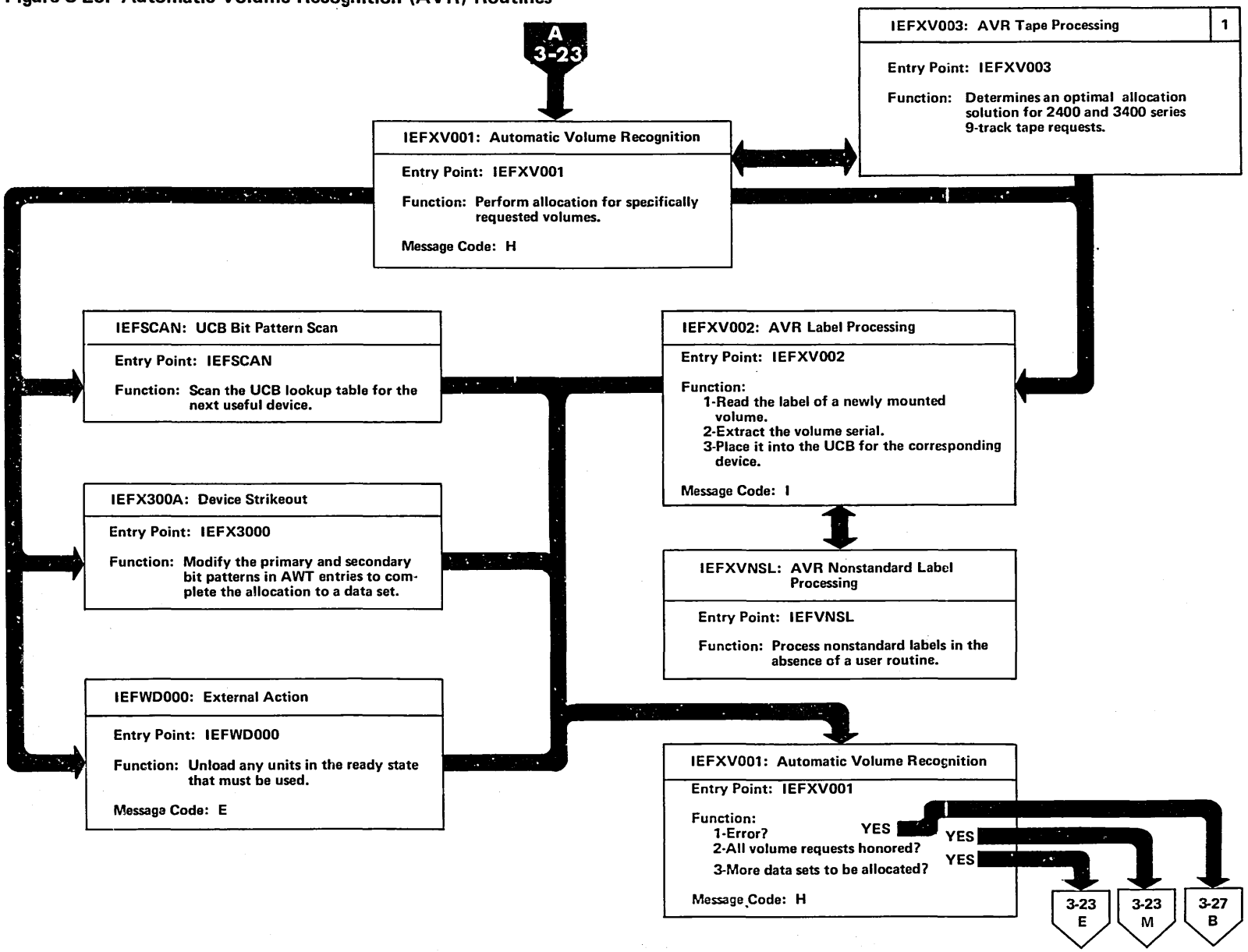


Figure 3-27. Decision Allocation

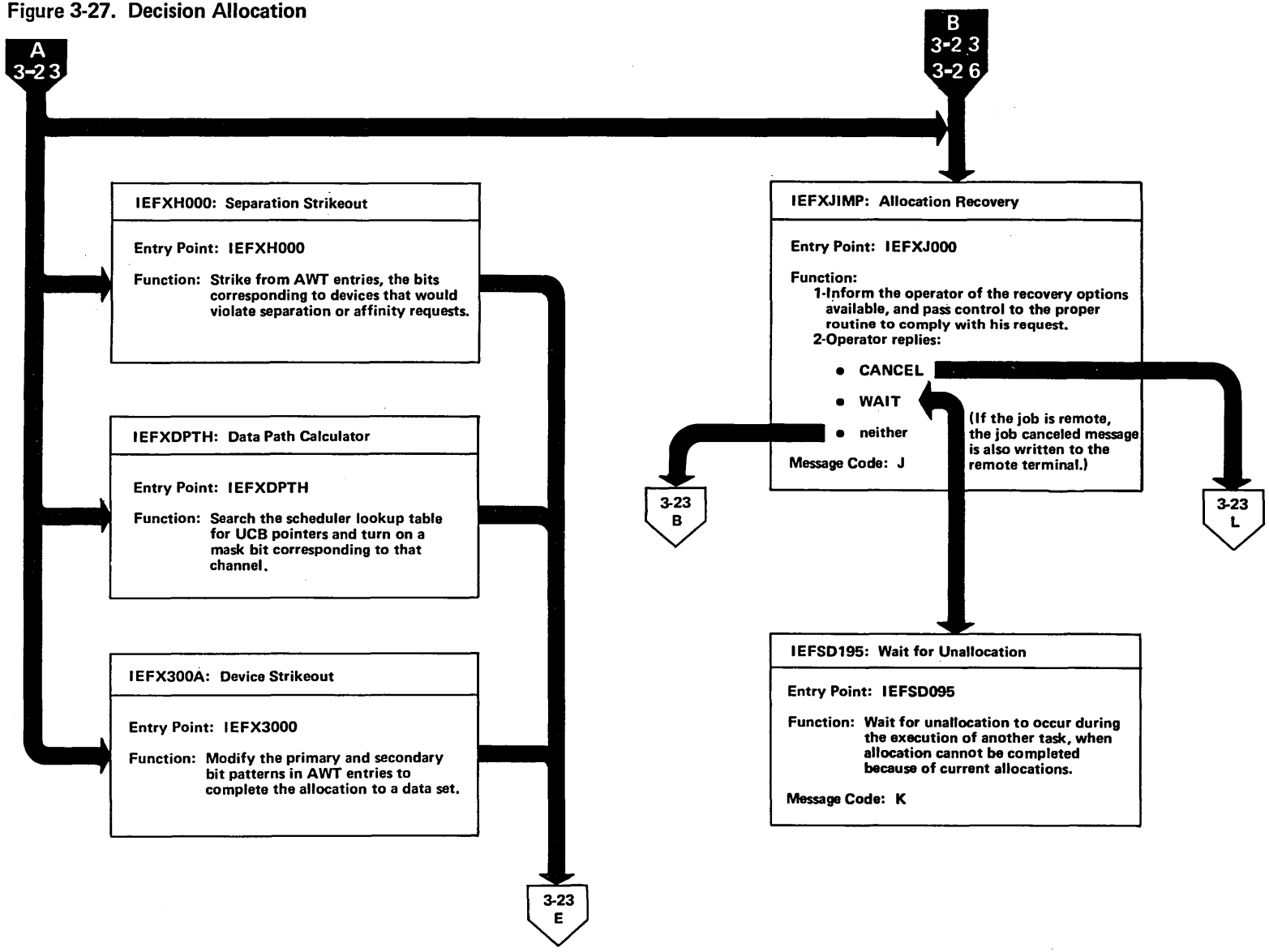


Figure 3-28. Space Allocation, TIOT Compression, and Extended Action Routines

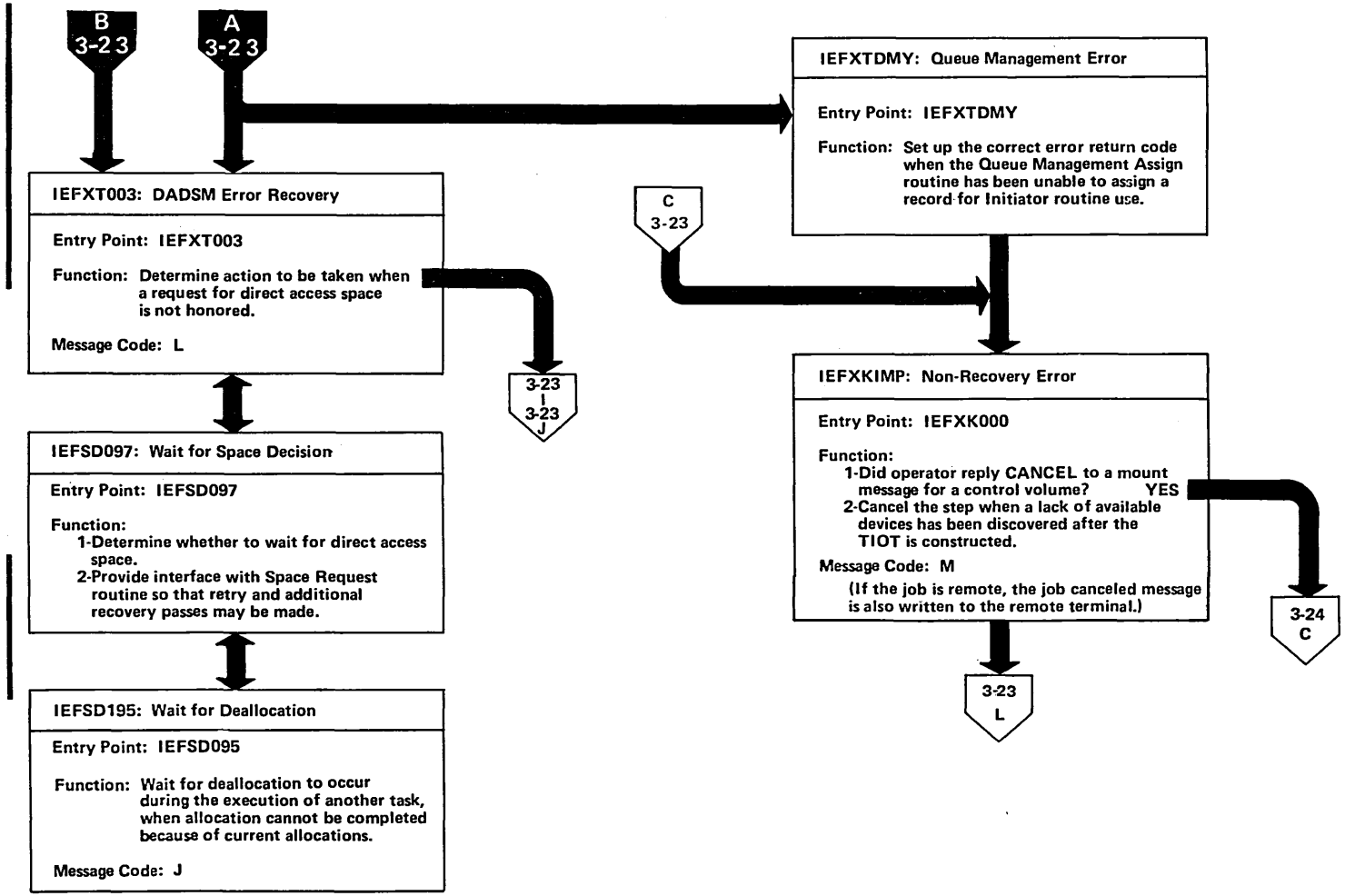


Figure 3-29. I/O Load Balancing Routines

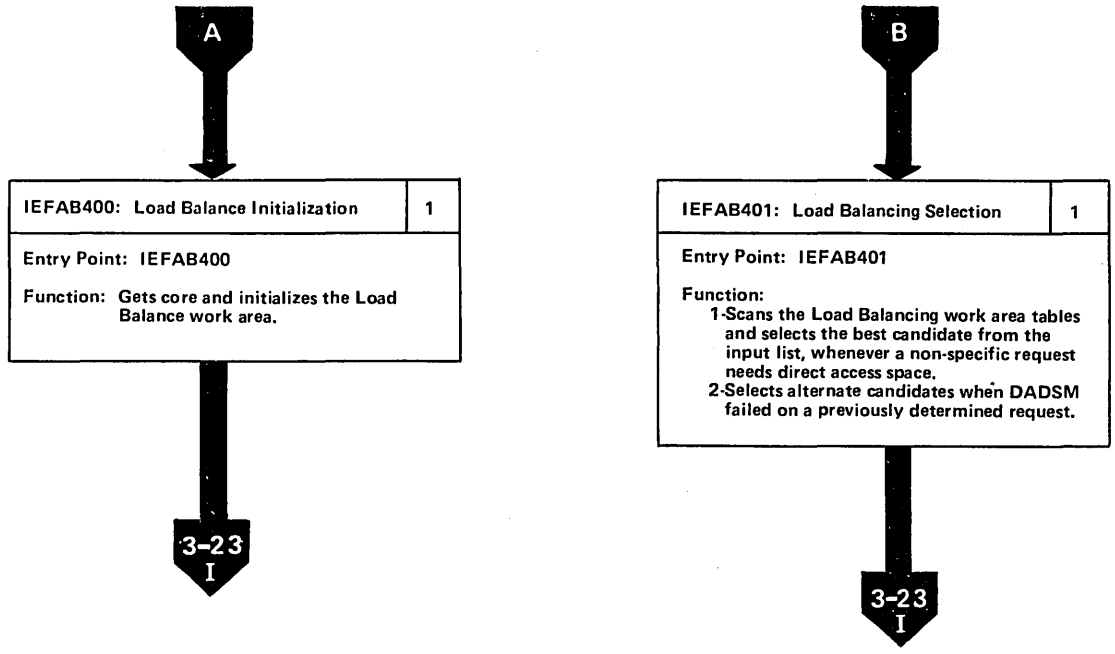


Figure 3-30. Messages Issued by Allocation Modules

Messages Code	Number	Reason
A	IEF403I * IEF452I *	Job started. Job not run - JCL error.
B	IEF202I	Step not run.
C	IEF210I IEF231I IEF232I IEF308I IEF309I IEF310I IEF315I IEF492I IEF4931	Incorrect device name. Queue manager I/O error. No space on job queue. Not enough storage. Queue manager read error on SIOT. Queue manager read error on VOLT. Volume count inconsistent with unit count. Data set resides on more than one volume. Invalid JCL parameter.
D	IEF230I IEF240I IEF243E IEF248I IEF279I IEF281I IEF282I	Too many separations. Too many DD cards. Volume has ANS label. Volume not mounted, I/O error. Volume now mounted. Device now offline. Device now unloaded.
E	IEF233A IEF234E IEF533A	Mount volume message. Demount volume message. Mount volume message.
F	IEF234E IEF533A	Demount volume message. Mount volume message.
G	IEF210I IEF211I IEF212I IEF213I IEF214I IEF215I IEF216I IEF217I IEF218I IEF219I IEF220I IEF221I IEF231I IEF232I IEF277I IEF286I IEF294I IEF308I IEF309I IEF318I IEF372I IEF310I IEF315I	Incorrect device name. Control volume not mounted Data set not found. Data set not specified. Data set not lowest level. DSNAME syntax error. No cataloged data set. Required volume not mounted. Record not found in VTOC. GDG group name exceeds 35 characters GDG ALL causes total DDs to exceed 255. Backward referenced step was not executed. Queue manager I/O error. No space on job queue. Control volume not mounted. DISP not compatible with DSNAME. More than 8 output classes. Not enough storage. Queue manager read error on SIOT. Invalid unit affinity request. Invalid volume field reference. Queue manager read error on VOLT. Volume count inconsistent with unit count.

Messages Code	Number	Reason
H	IEF500I IEF501I IEF502I IEF503I IEF505I IEF508I	Volume needed on different unit. Unit needs different volume. Duplicate serial. Wrong density or label. Unit required. Queue full.
I	IEF504A	Mount volume message.
J	IEF235I IEF238A IEF240I IEF241I IEF244I IEF245I IEF247I IEF251I * IEF265I IEF296I IEF388I IEF389I IEF506I IEF594I	Waiting for volumes. Device or space unavailable. Too many DD cards. Resident device offline. Unable to allocate. Inconsistent name and serial. Allocation recovery Job canceled. Invalid SUBALLOC/SPLIT request. Maximum number of devices exceeded. Waiting for devices. Separation request ignored. No storage volumes. Invalid split requests.
K	IEF238I IEF239I	Device or space unavailable. Waiting for allocation.
L	IEF454I	DOS VTOC cannot be converted.
M	IEF127I IEF128I IEF129I IEF130I IEF131I IEF132I IEF133I IEF134I IEF135I IEF136I IEF137I IEF138I IEF139I IEF140I IEF141I	No space parameter given for a new data set or zero space request at ABSTR zero. Invalid request for ISAM index. Multivolume index not allowed. DSNAME element wrong. Must be INDEX, OVFLOW, or PRIME. Multivolume OVFLOW request not allowed. Space parameter wrong. ABSTR and CYL conflict. Space parameter wrong. CYL and CONTIG conflict. Subparameter wrong in SPACE parameter. Must be CYL or ABSTR. Primary space request cannot be zero. Duplication in allocation. Index area requested twice. Suballoc data set not found. Suballoc data set has more than 1 extent; not allowed. Suballoc data set is password-protected. Directory space request is larger than the amount available on this volume. Index request must precede prime for ISAM data set.

Messages Code	Number	Reason
M (cont.)	IEF143I IEF144I IEF145I IEF146I IEF240I IEF246I IEF251I * IEF252I IEF253I IEF254I IEF255I IEF256I IEF257I IEF258I IEF260I IEF261I IEF262I IEF263I IEF264I IEF265I IEF266I IEF267I IEF265I IEF266I IEF267I IEF273I IEF370I IEF454I	Last concatenated DD card unnecessary or invalid for this data set. Cannot suballocate from a split cylinder data set. Space request must be ABSTR for BOS volume. Duplicate data set names within split cylinder request. Too many DD cards. Insufficient space. Job canceled. DA space unavailable. Duplicate name on volume. No space in VTOC. Permanent I/O error. Absolute track unavailable. Space requested unavailable. Invalid record length. Wrong DSORG or DISP. No prime area request for ISAM data set. Prime area must be requested before OVFLOW. Space request wrong; must be on cylinder boundry. Duplication of the DSNAME element not allowed; same area requested twice. Number of tracks requested too large for split request. Invalid JFCB or model DSCB pointer. Invalid directory request. Invalid SUBALLOC/SPLIT request. Space request must be ABSTR. Invalid directory request. Invalid user label request. Insufficient storage for mount processing. DOS VTOC cannot be converted.
N	IEF236I IEF237I IEF238A IEF248I IEF252I IEF546I	Identifies the job step. Identifies devices allocated to data sets. Cannot allocate device or space requested. Error allocating reserved volume. DA space unavailable. No storage volumes.
Q	IEF491I	Excess DD statements generated.
R	IEF308I	Insufficient main storage.

* If job is remote, the message is also written to the remote terminal by indicated modules.

Figure 3-31. Termination Interconnection Module Diagram

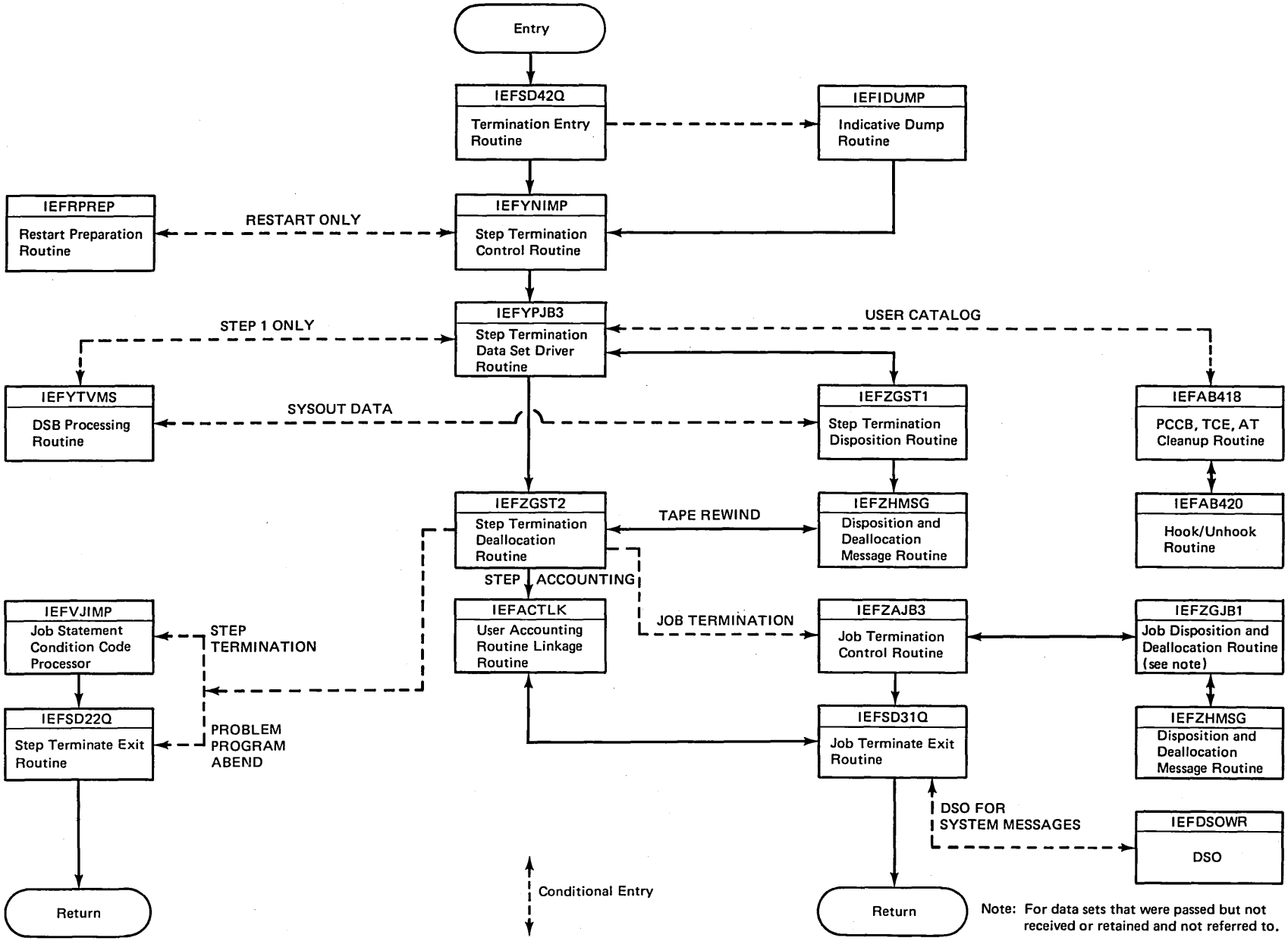


Figure 3.32. Step Termination (Part 1 of 2)

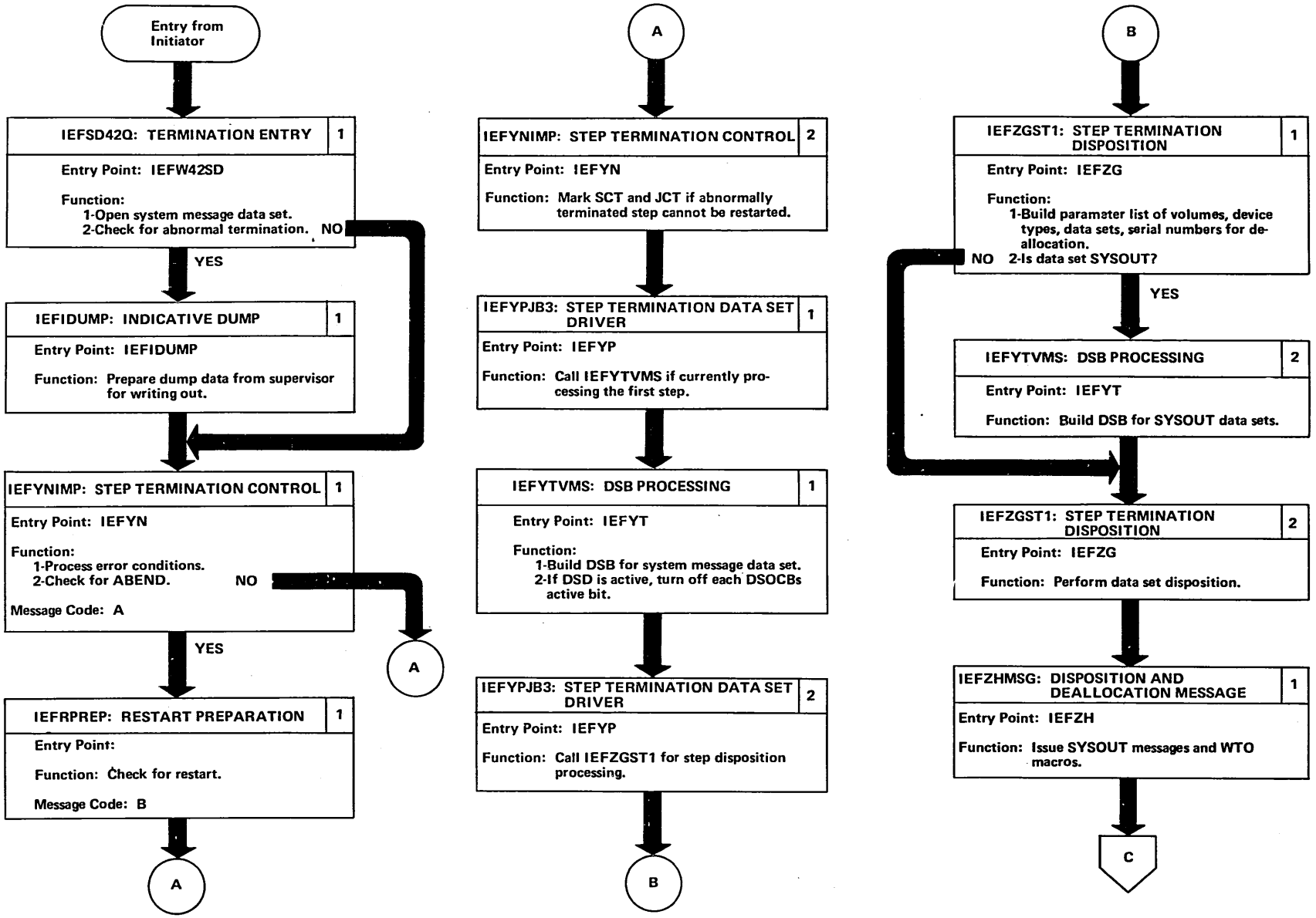
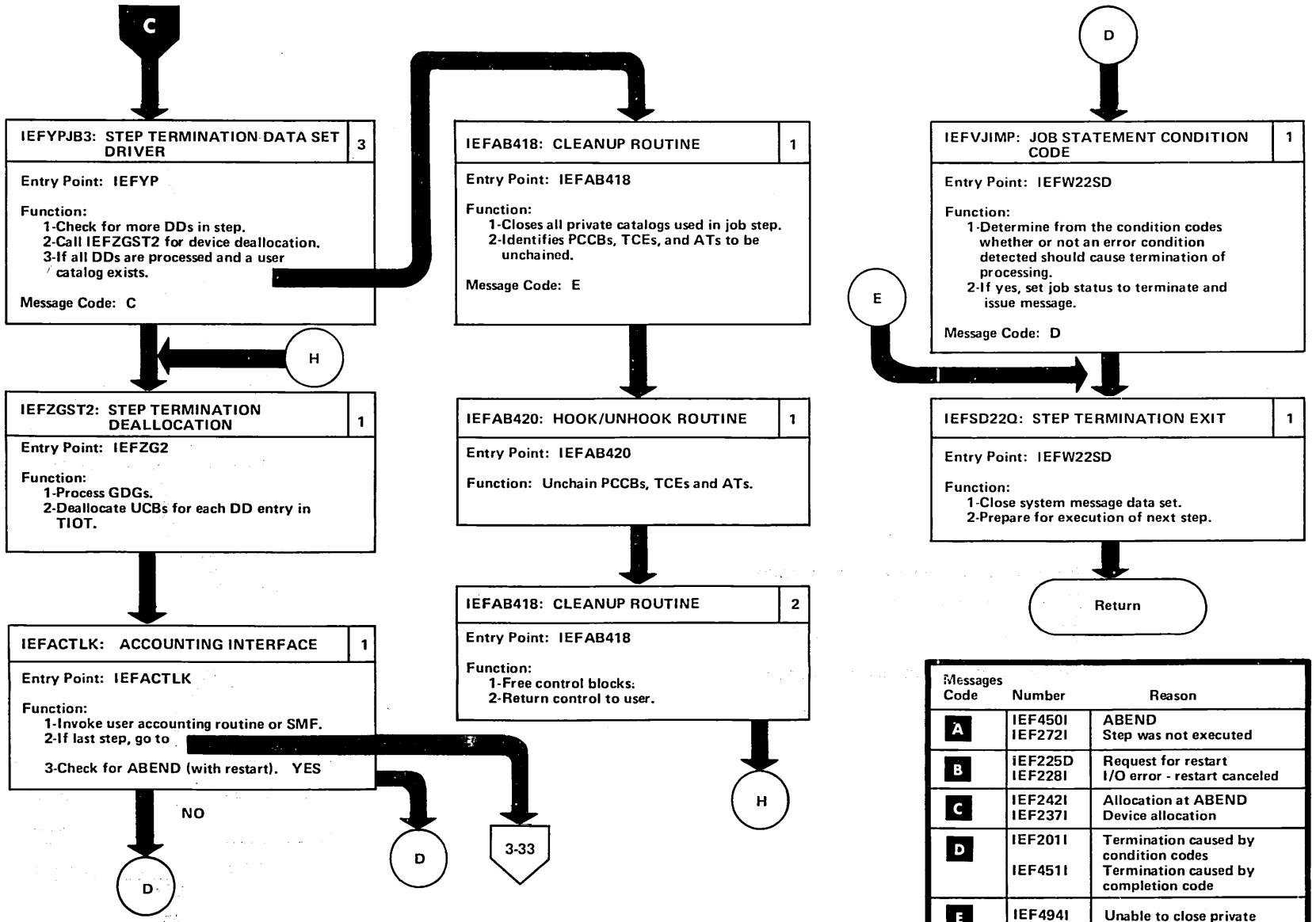
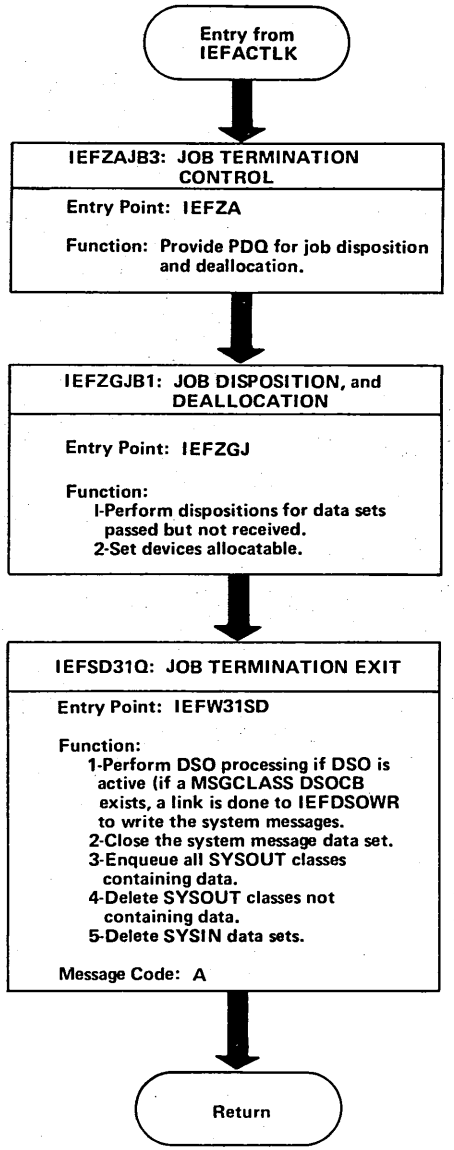


Figure 3-32. Step Termination (Part 2 of 2)



Messages Code	Number	Reason
A	IEF450I IEF272I	ABEND Step was not executed
B	IEF225D IEF228I	Request for restart I/O error - restart canceled
C	IEF242I IEF237I	Allocation at ABEND Device allocation
D	IEF201I IEF451I	Termination caused by condition codes Termination caused by completion code
E	IEF494I	Unable to close private catalog

Figure 3-33. Job Termination



Message Code	Number	Reason
A	IEF298I	SYSOUT classes

Figure 3-34. System Restart Interconnection Module Diagram

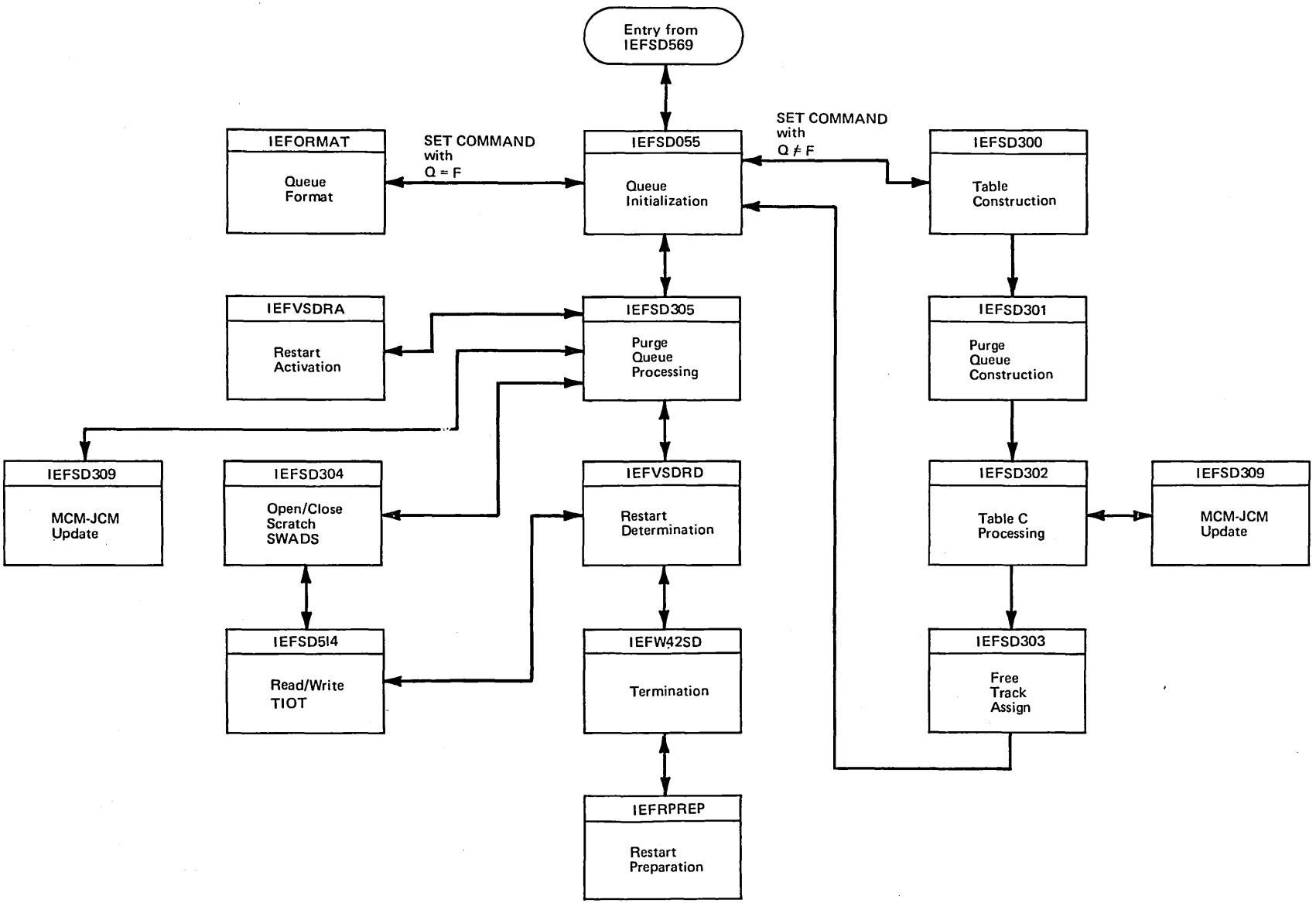


Figure 3-35. System Restart (Part 1 of 5)

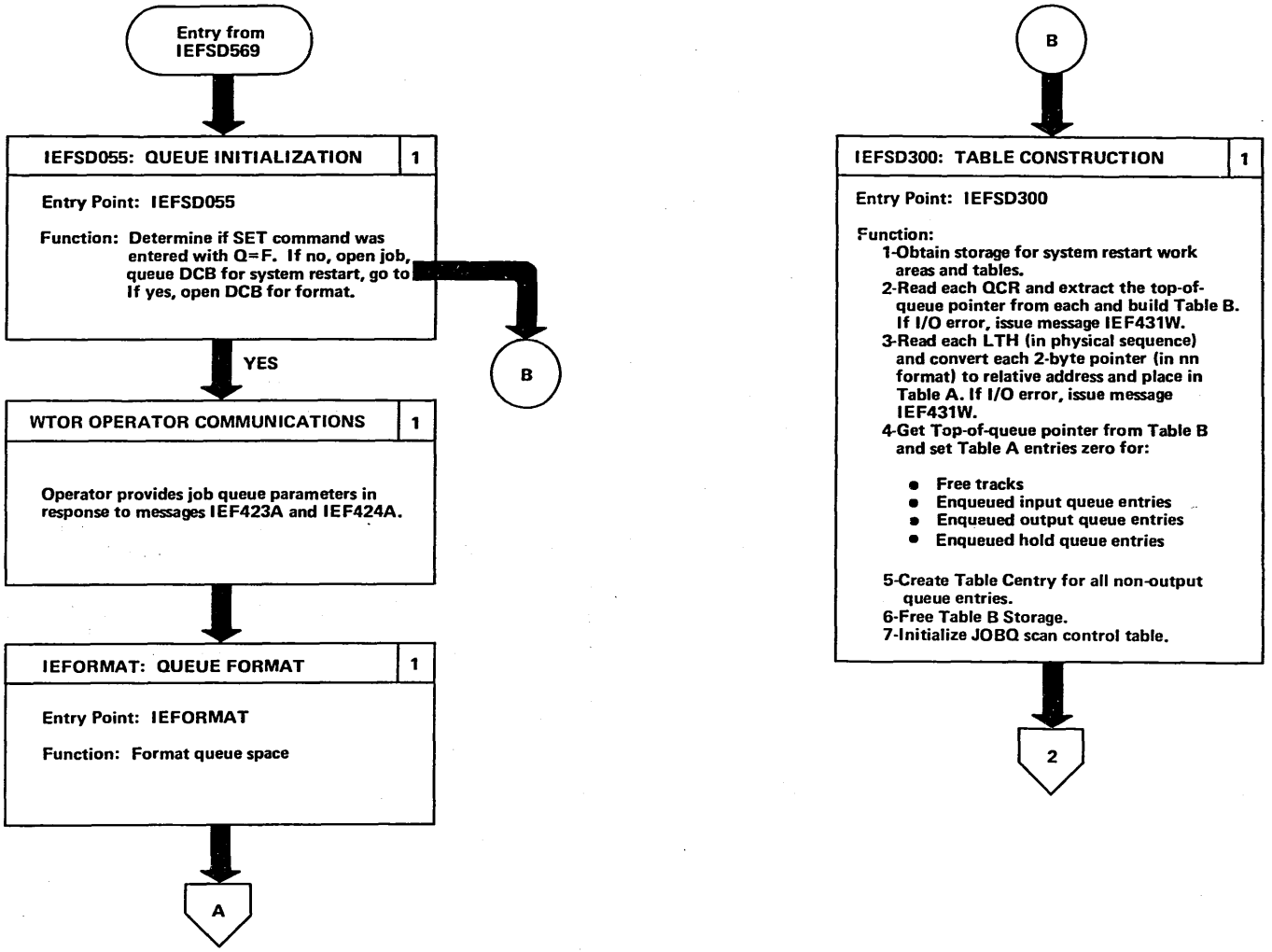


Figure 3-35. System Restart (Part 2 of 5)

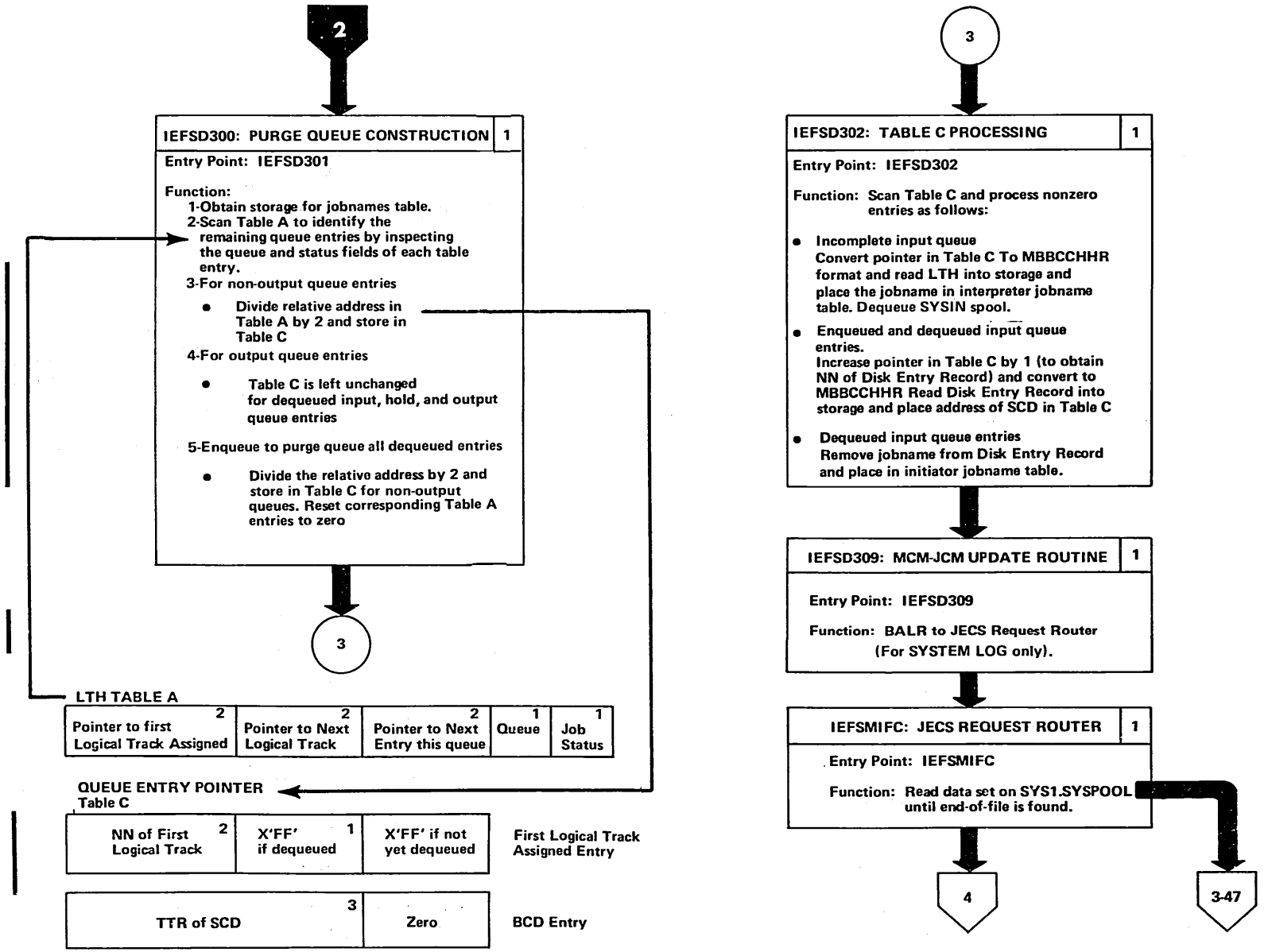


Figure 3-35. System Restart (Part 3 of 5)

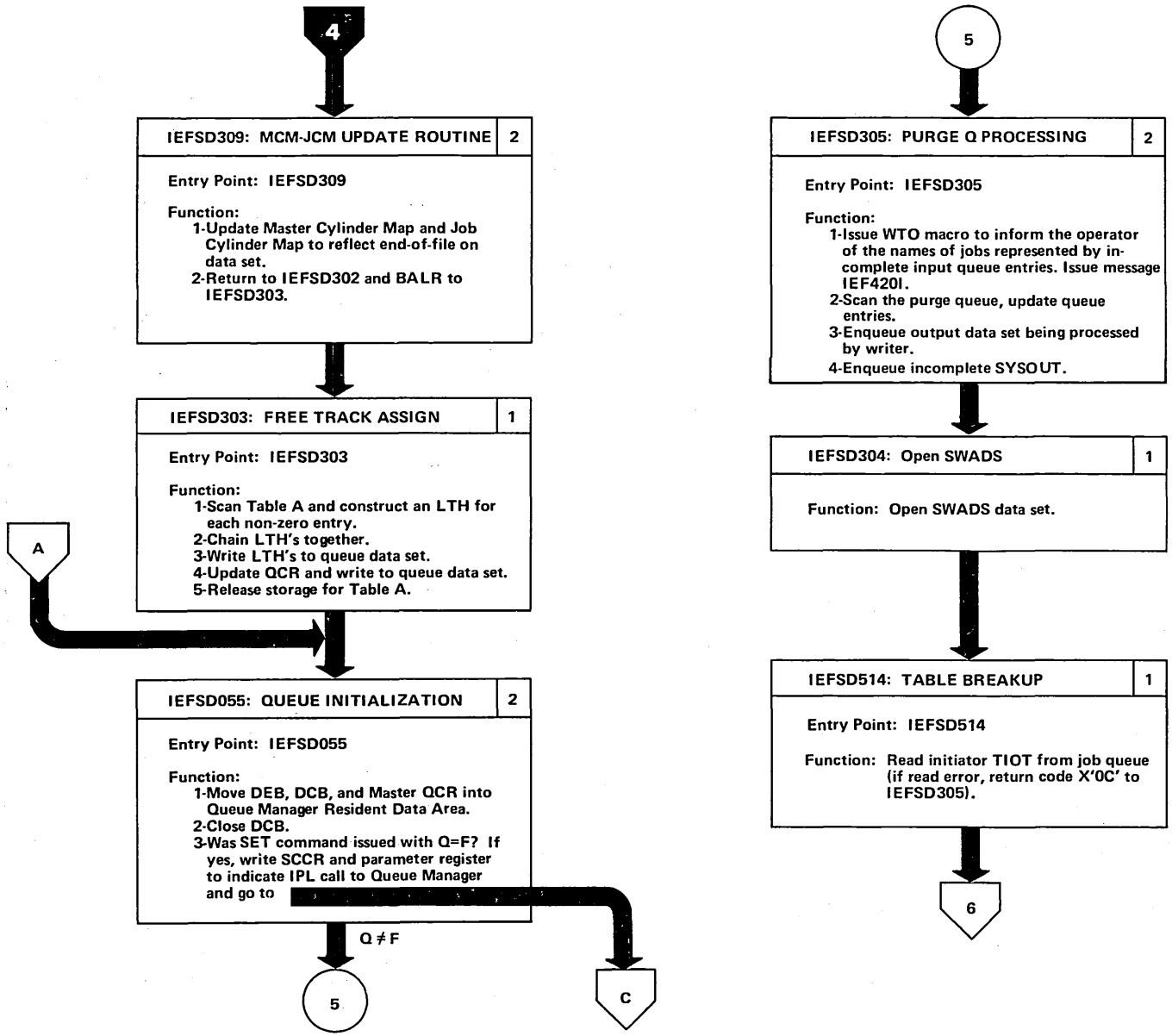


Figure 3-35. System Restart (Part 4 of 5)

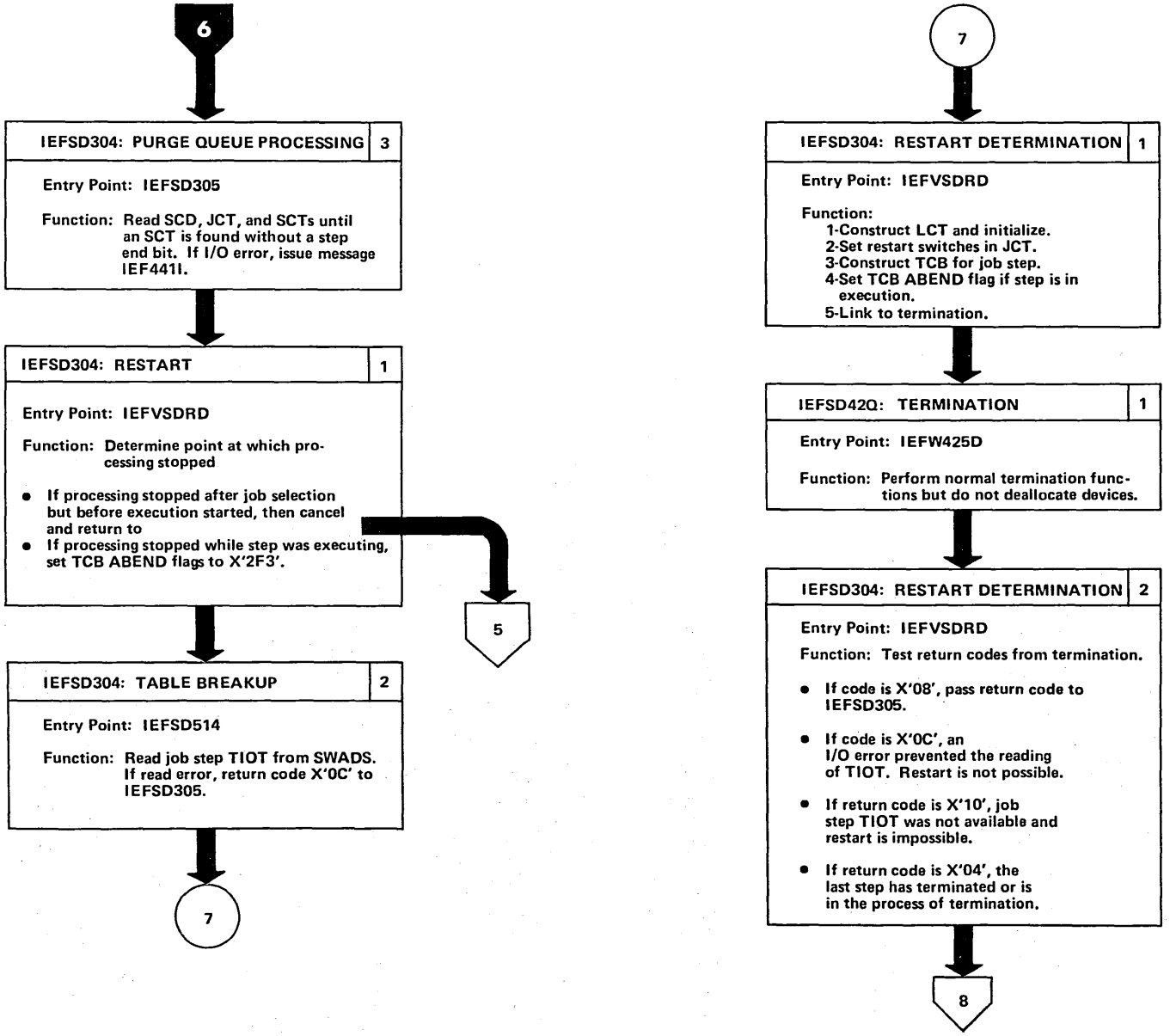


Figure 3-35. System Restart (Part 5 of 5)

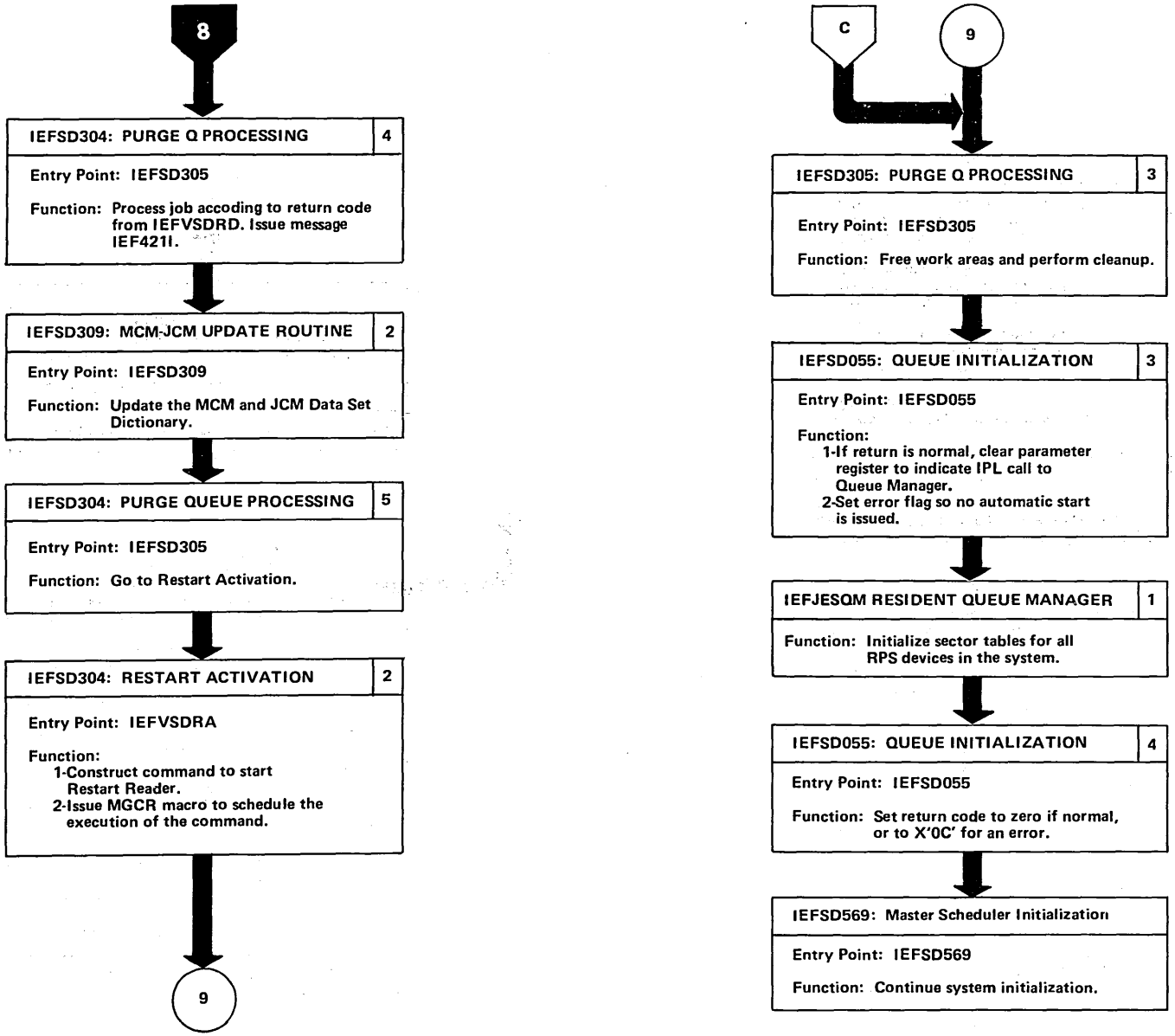


Figure 3-36. Restart Reader Interconnection Module Diagram

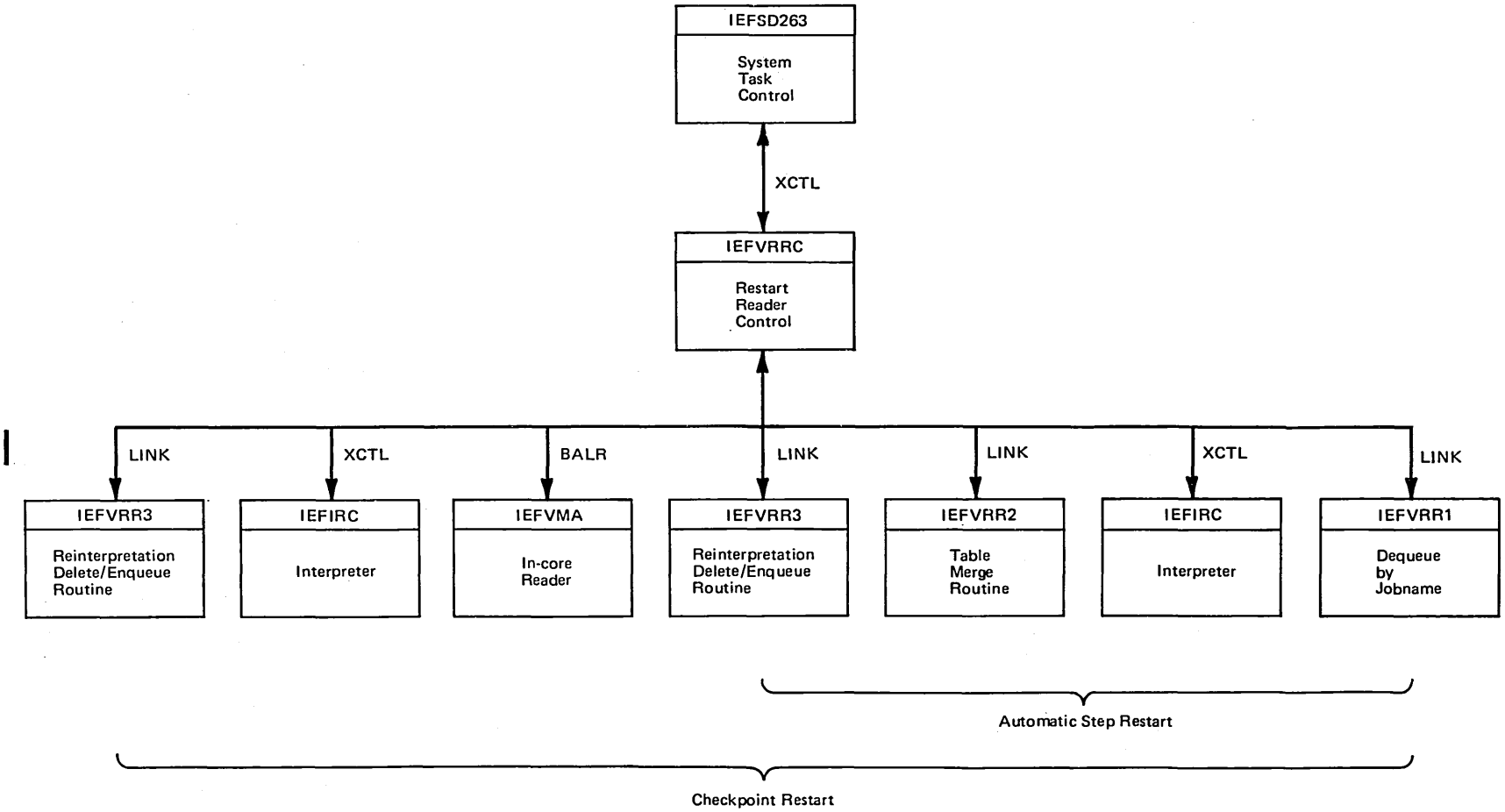


Figure 3-37. Restart Reader (Part 1 of 4)

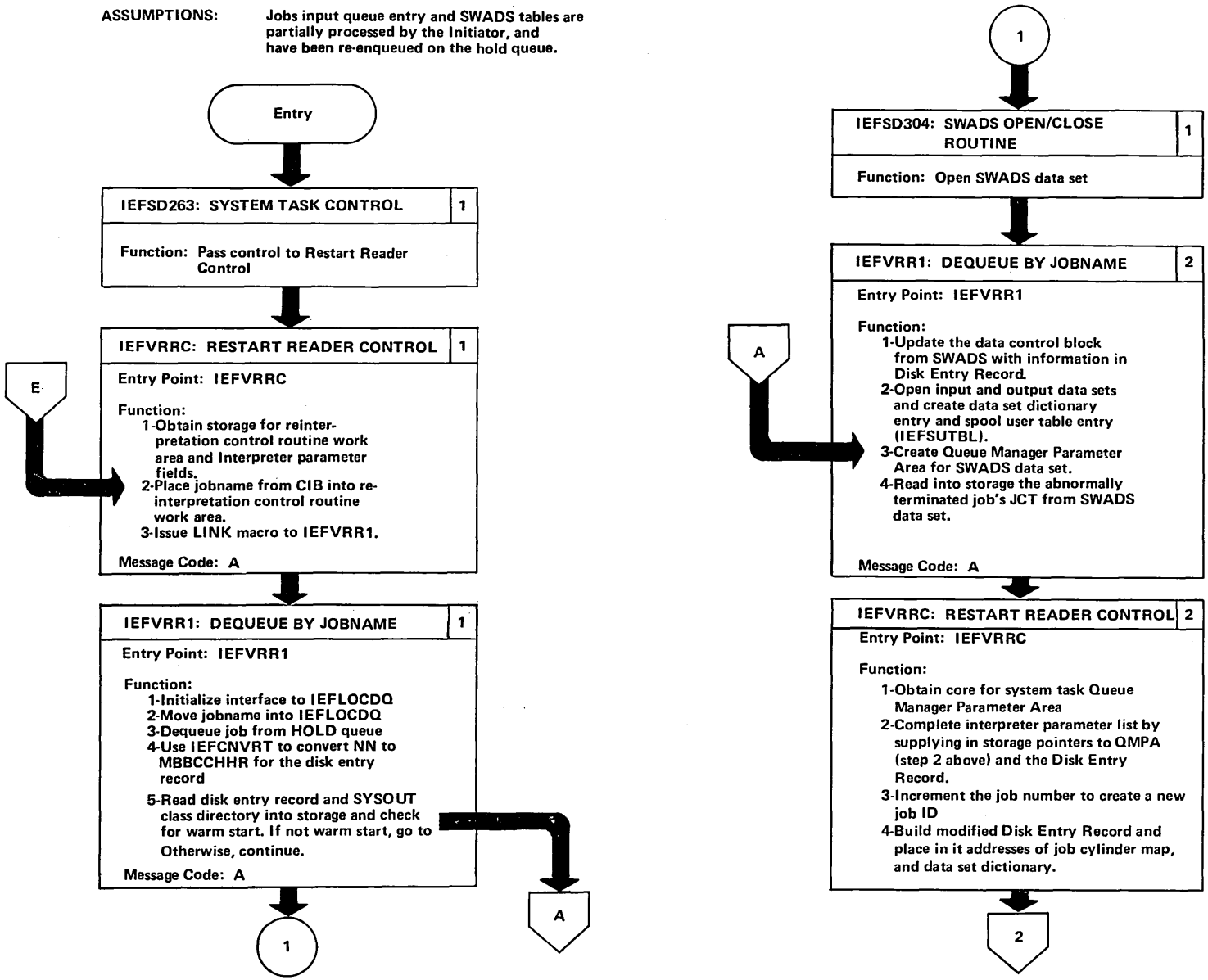


Figure 3-37. Restart Reader (Part 2 of 4)

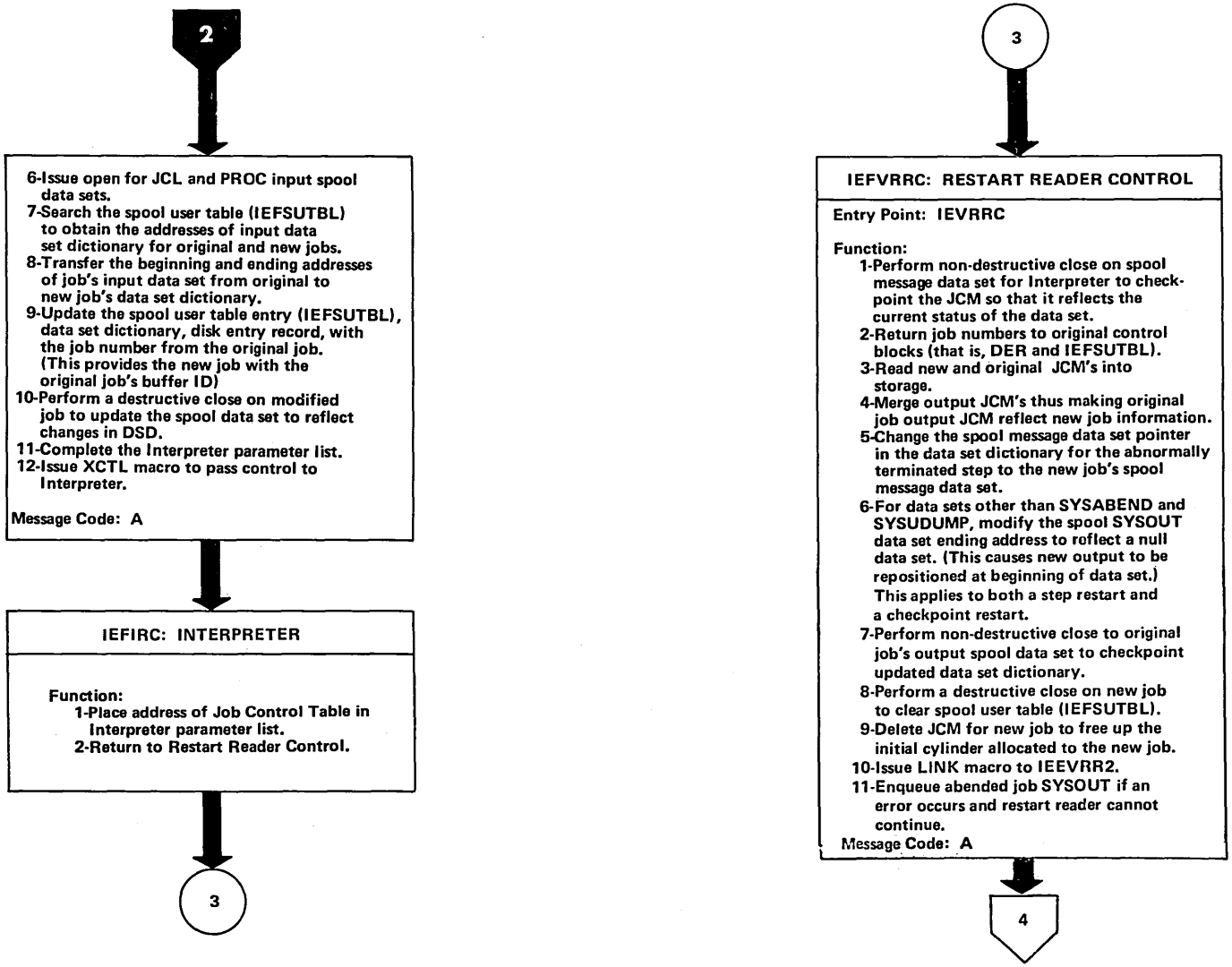


Figure 3-37. Restart Reader (Part 3 of 4)

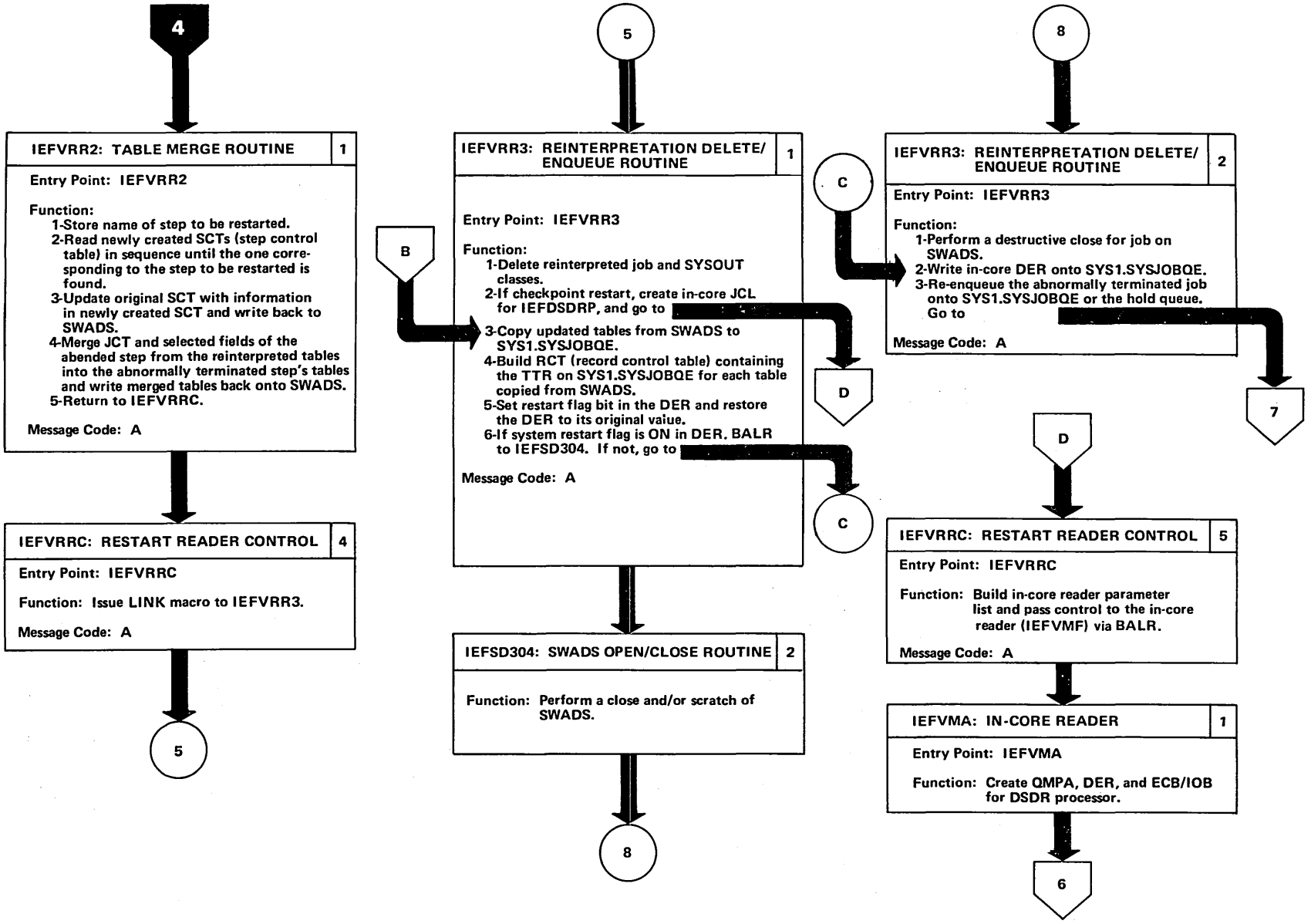
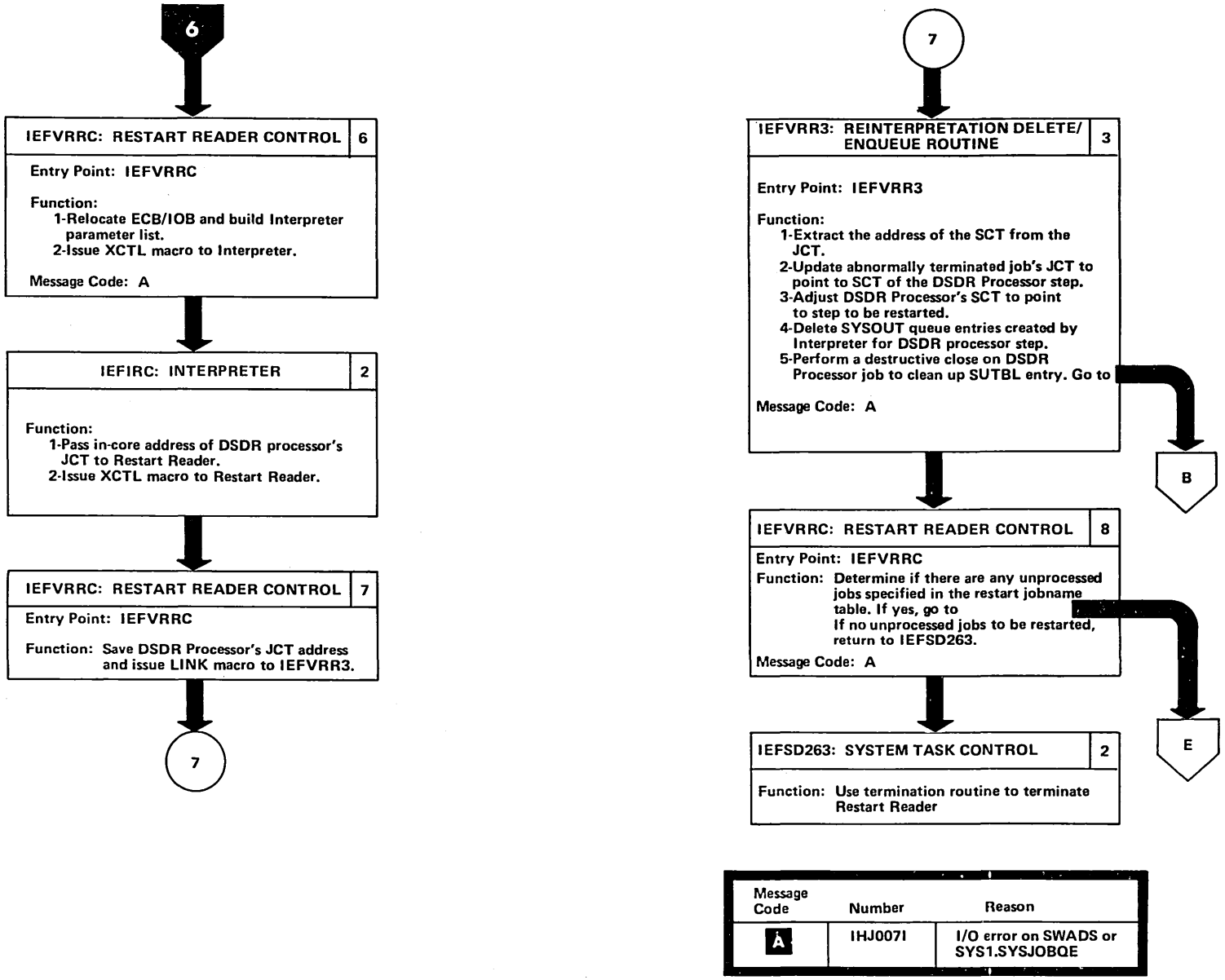


Figure 3-37. Restart Reader (Part 4 of 4)



Message Code	Number	Reason
A	IHJ007I	I/O error on SWADS or SYS1.SYSJOBQE

Figure 3-38. JES Reader Interconnection Module Diagram

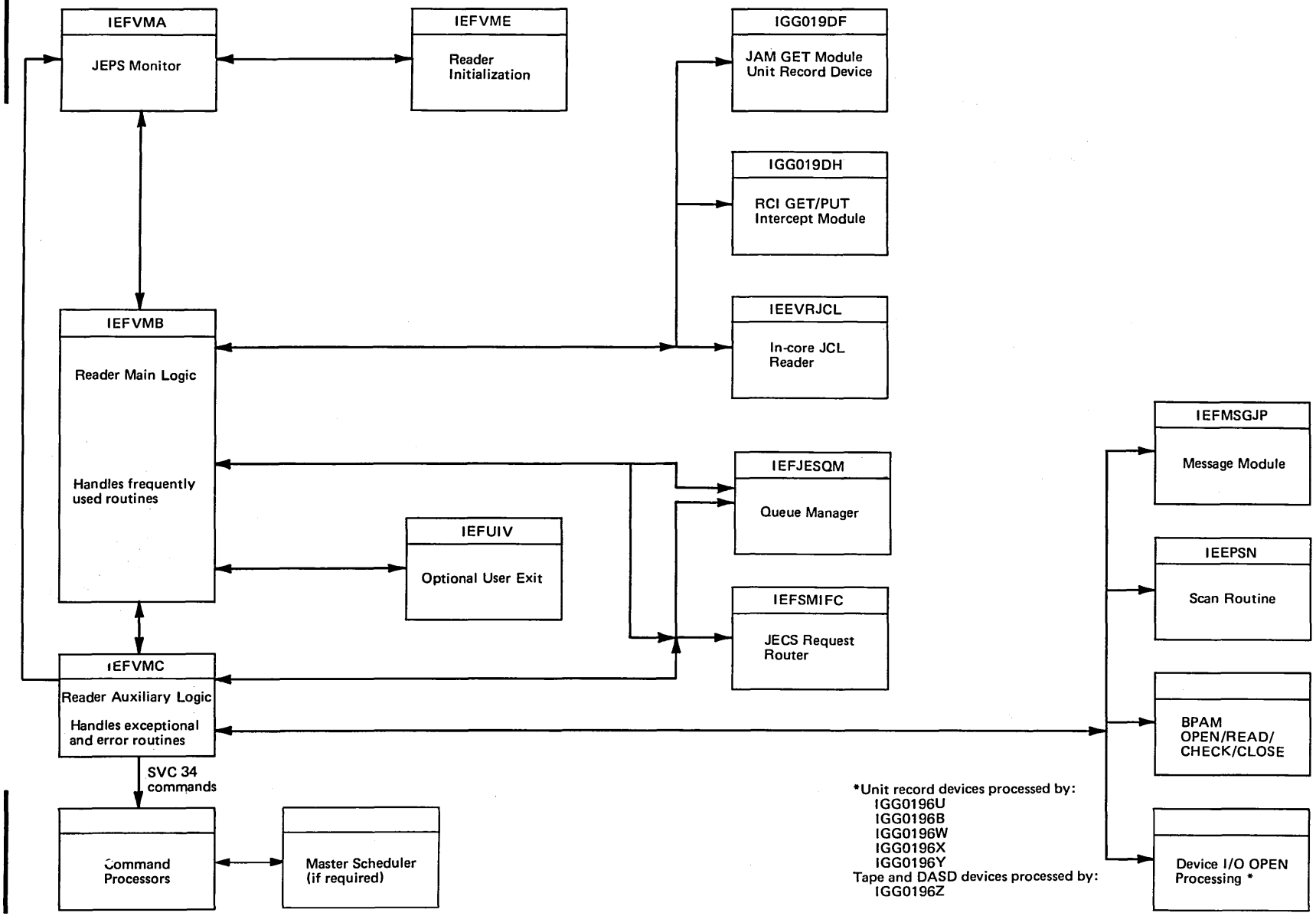


Figure 3-39. Start Reader/Writer

ASSUMPTIONS

The Problem Program Interface routine IEESD263 (Figure 3-3), has just posted the JEPS monitor following a START command for a reader or writer.

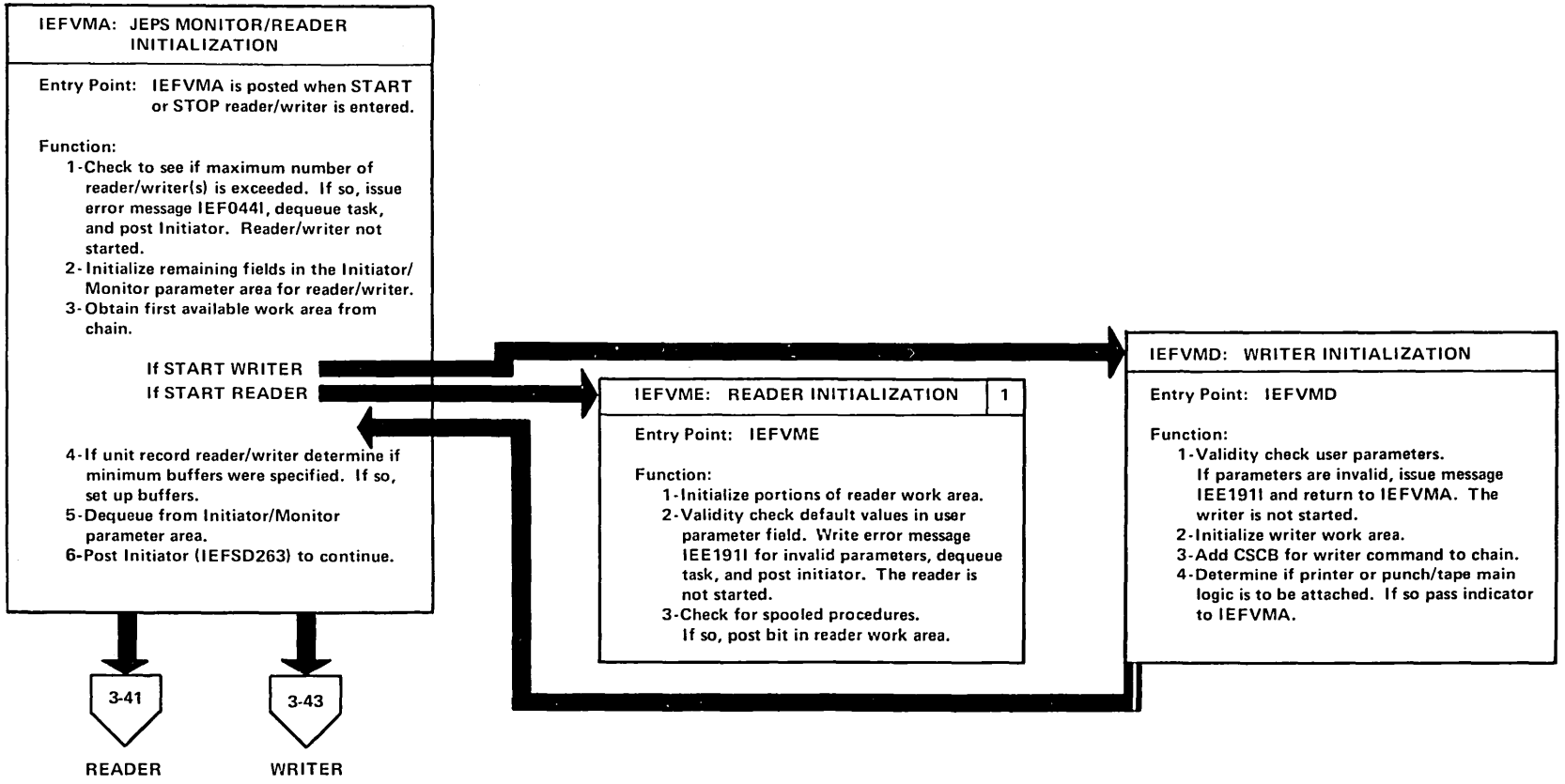


Figure 3-40. In-Core JCL Reader (START Command Processing) (Part 1 of 3)

ASSUMPTIONS

System Task Control, IEEVRCTL (Figure 3-3) has passed control to IEFVMF, the in-core reader interface module, following a START command.

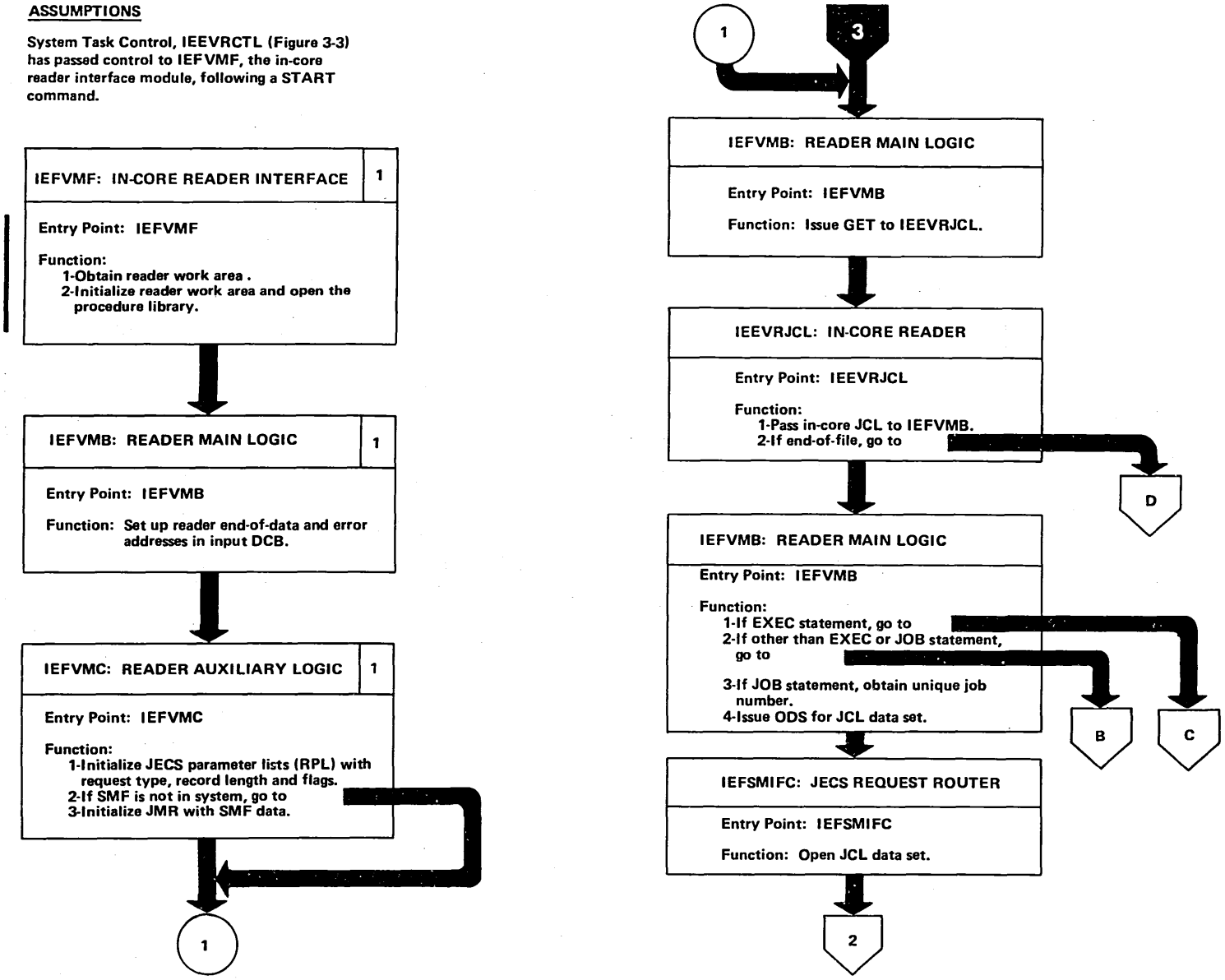


Figure 3-40. In-Core JCL Reader (START Command Processing) (Part 2 of 3)

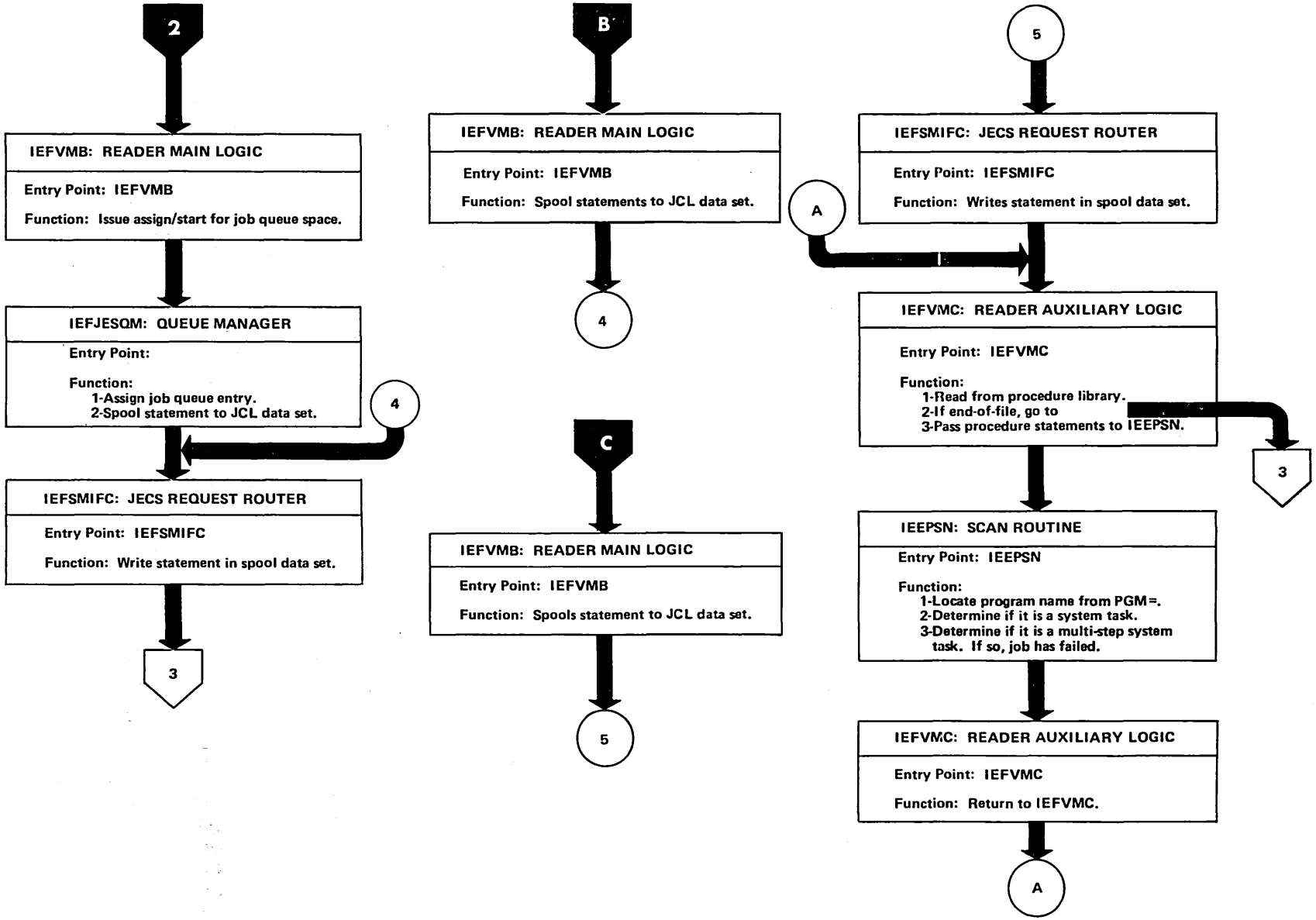


Figure 3-40. In-Core JCL Reader (START Command Processing) (Part 3 of 3)

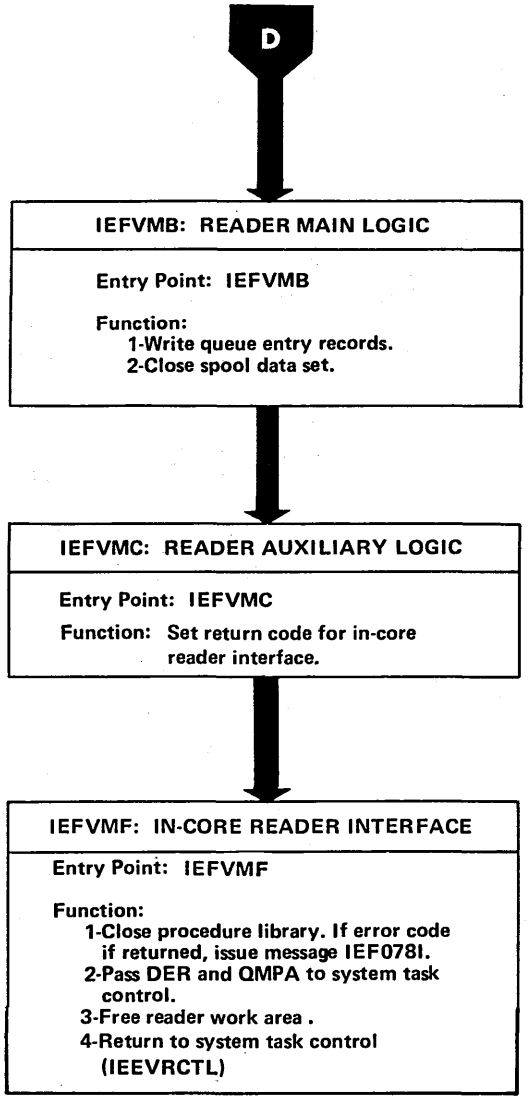


Figure 3-41. JES Reader Task (Part 1 of 6)

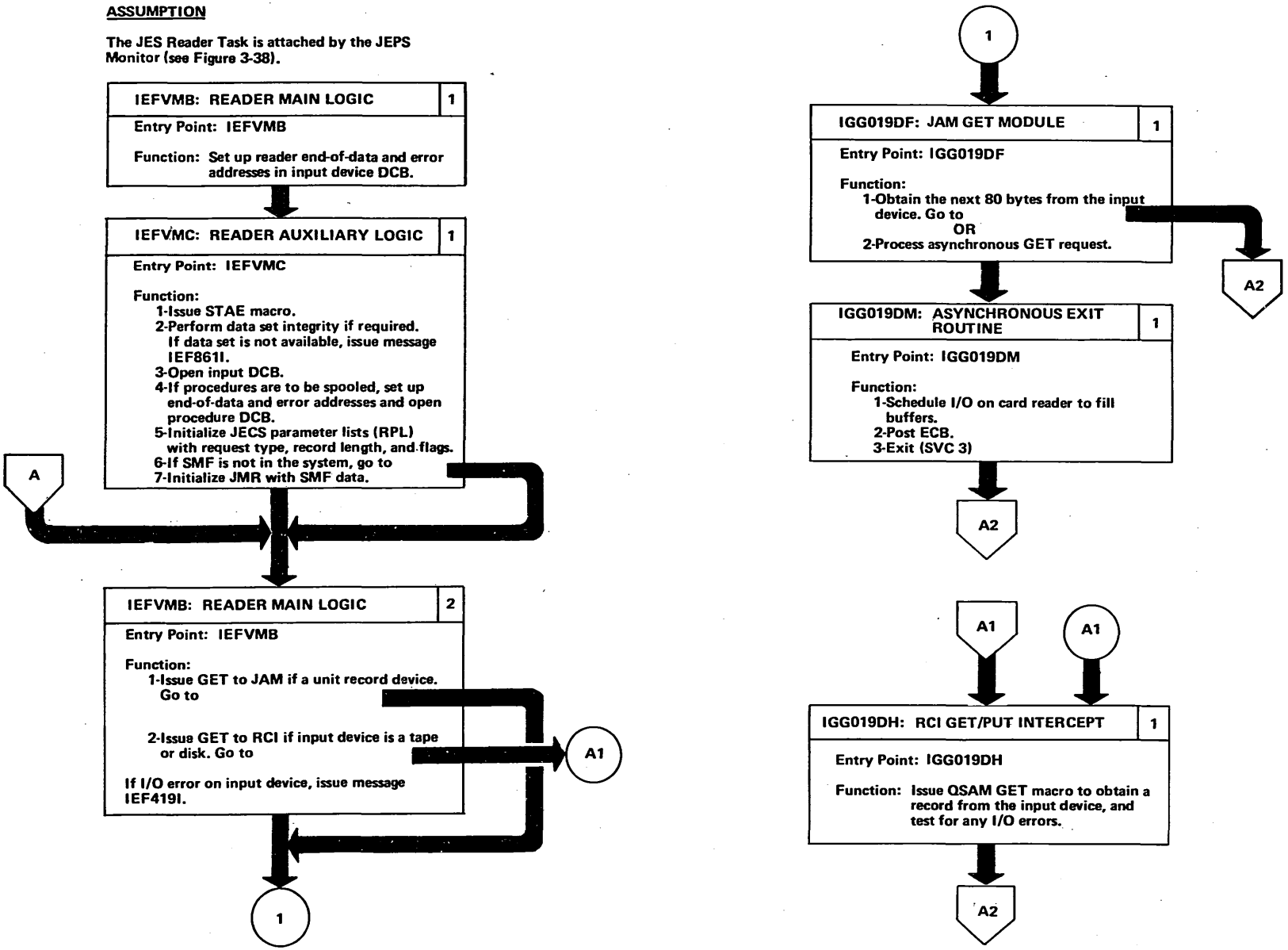


Figure 3-41. JES Reader Task (Part 2 of 6)

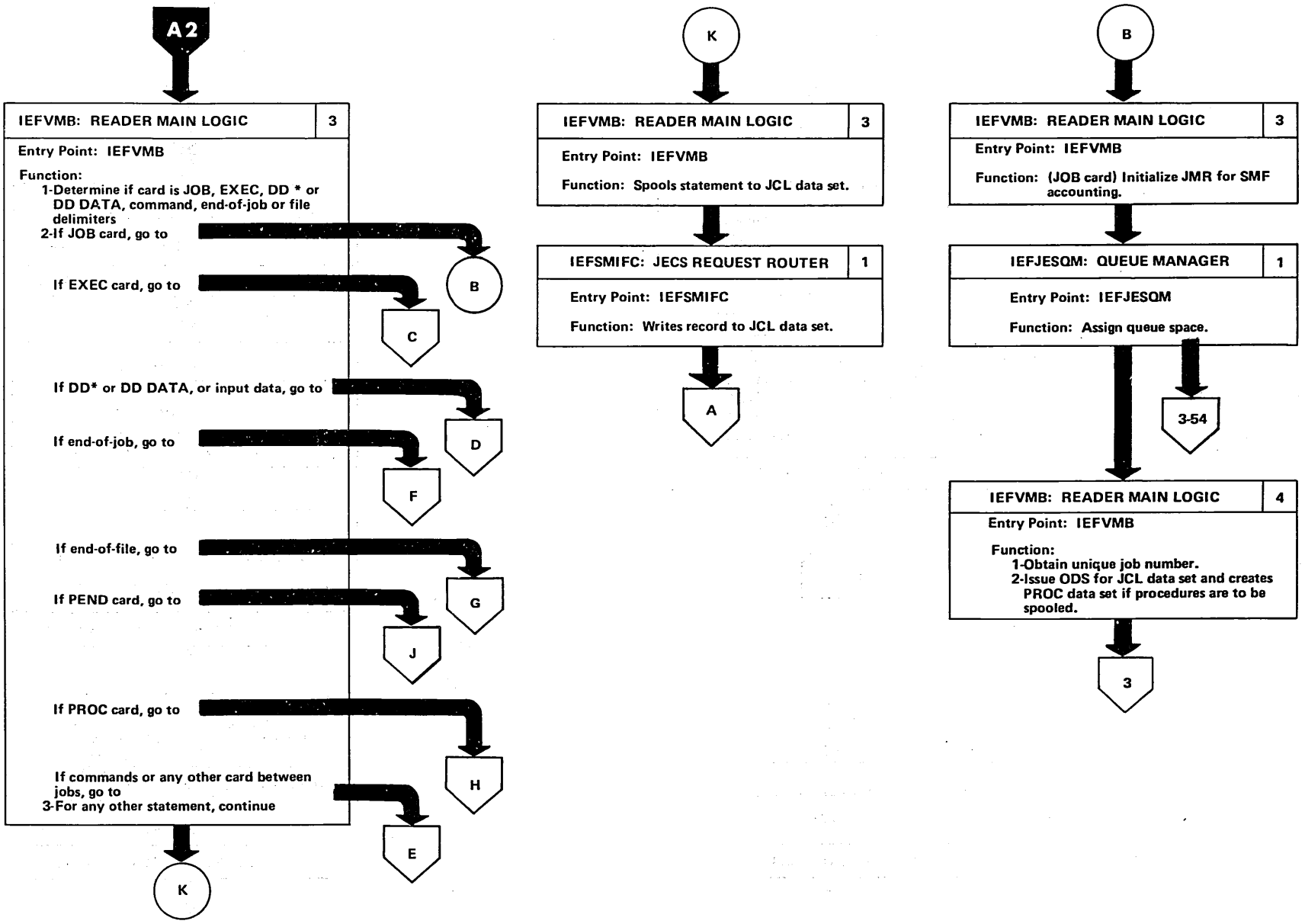


Figure 3-41. JES Reader Task (Part 3 of 6)

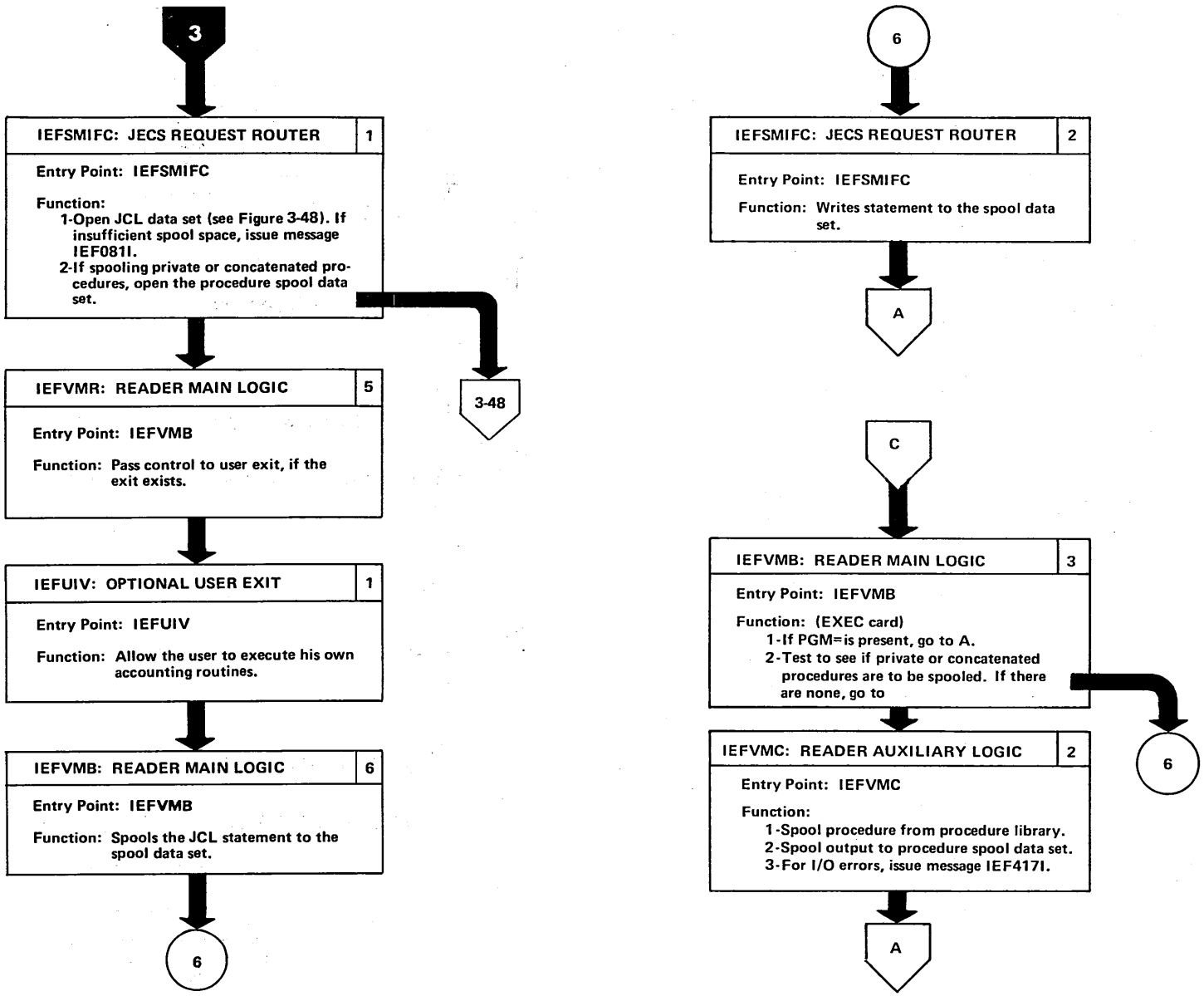


Figure 3-41. JES Reader Task (Part 4 of 6)

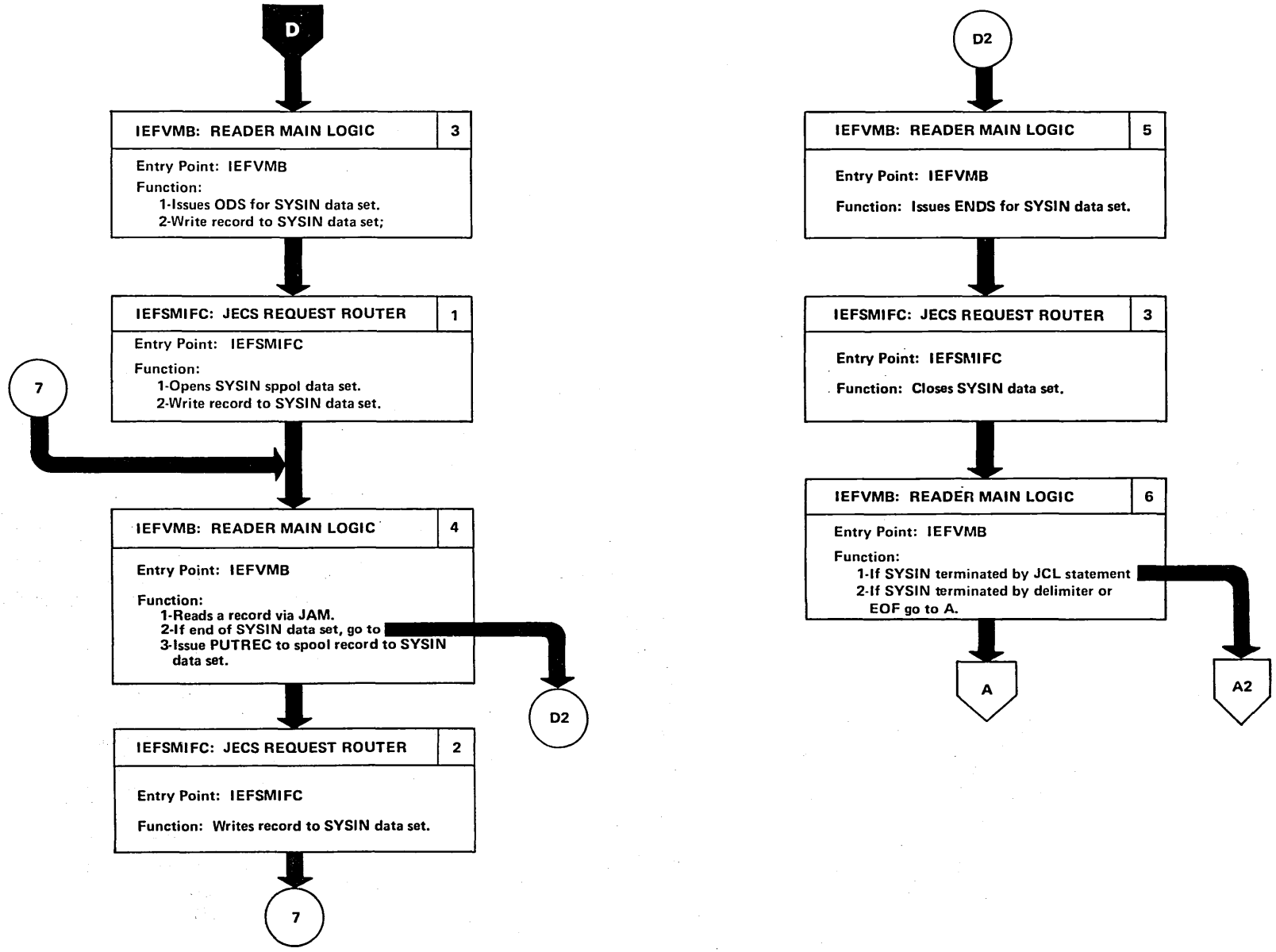


Figure 3-41. JES Reader Task (Part 5 of 6)

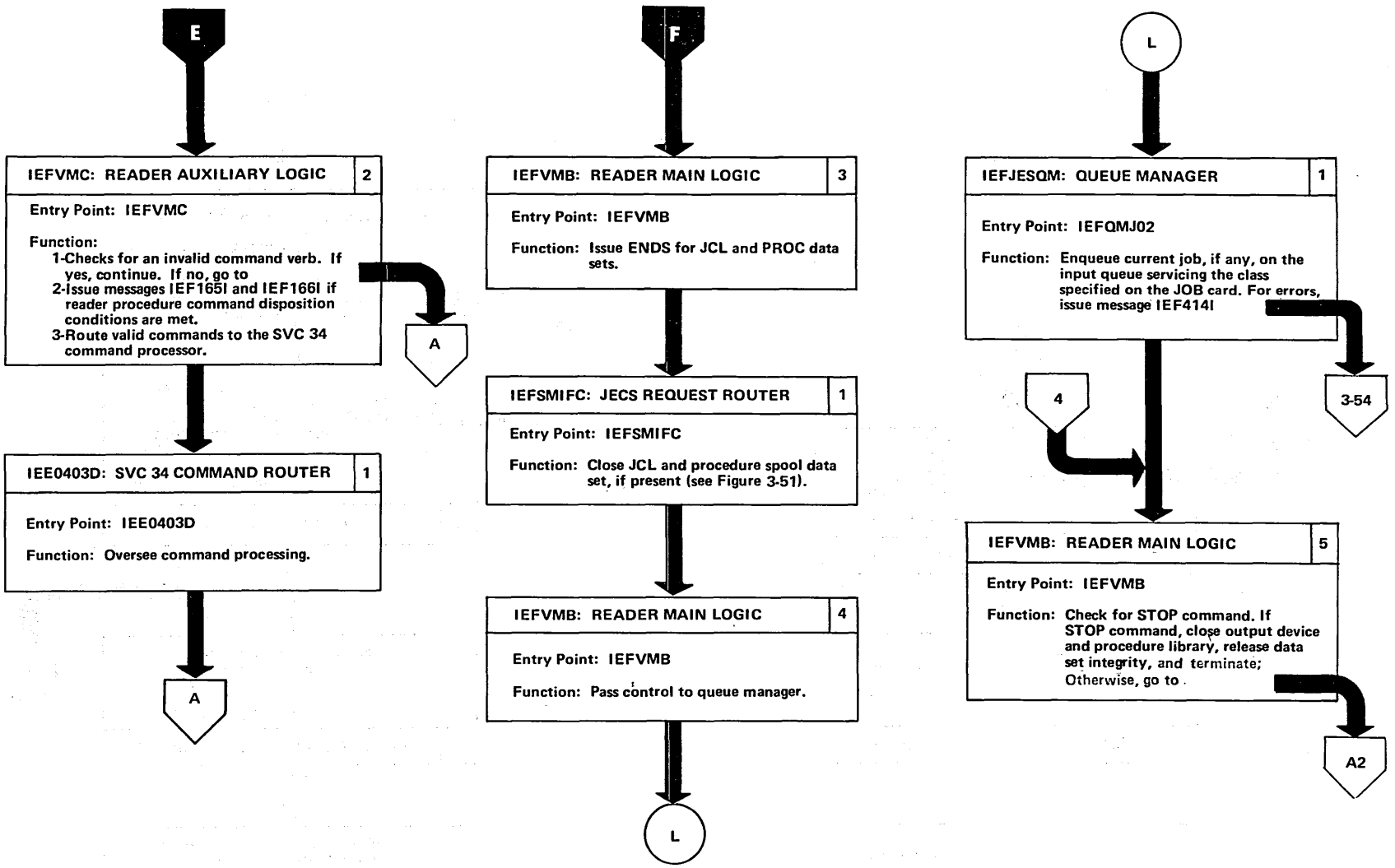


Figure 3-41. JES Reader Task (Part 6 of 6)

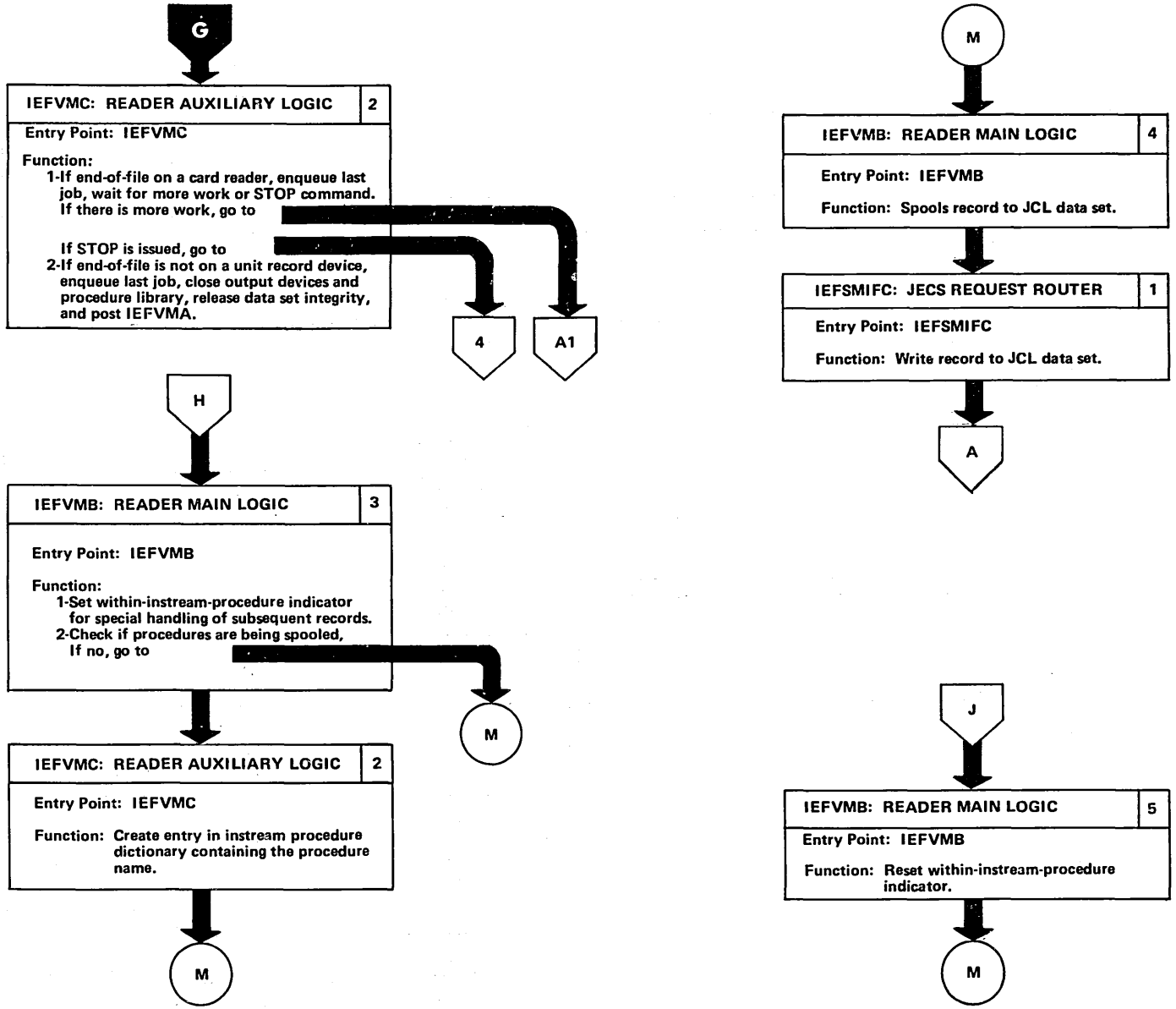


Figure 3-42. JES Writer Interconnection Module Diagram

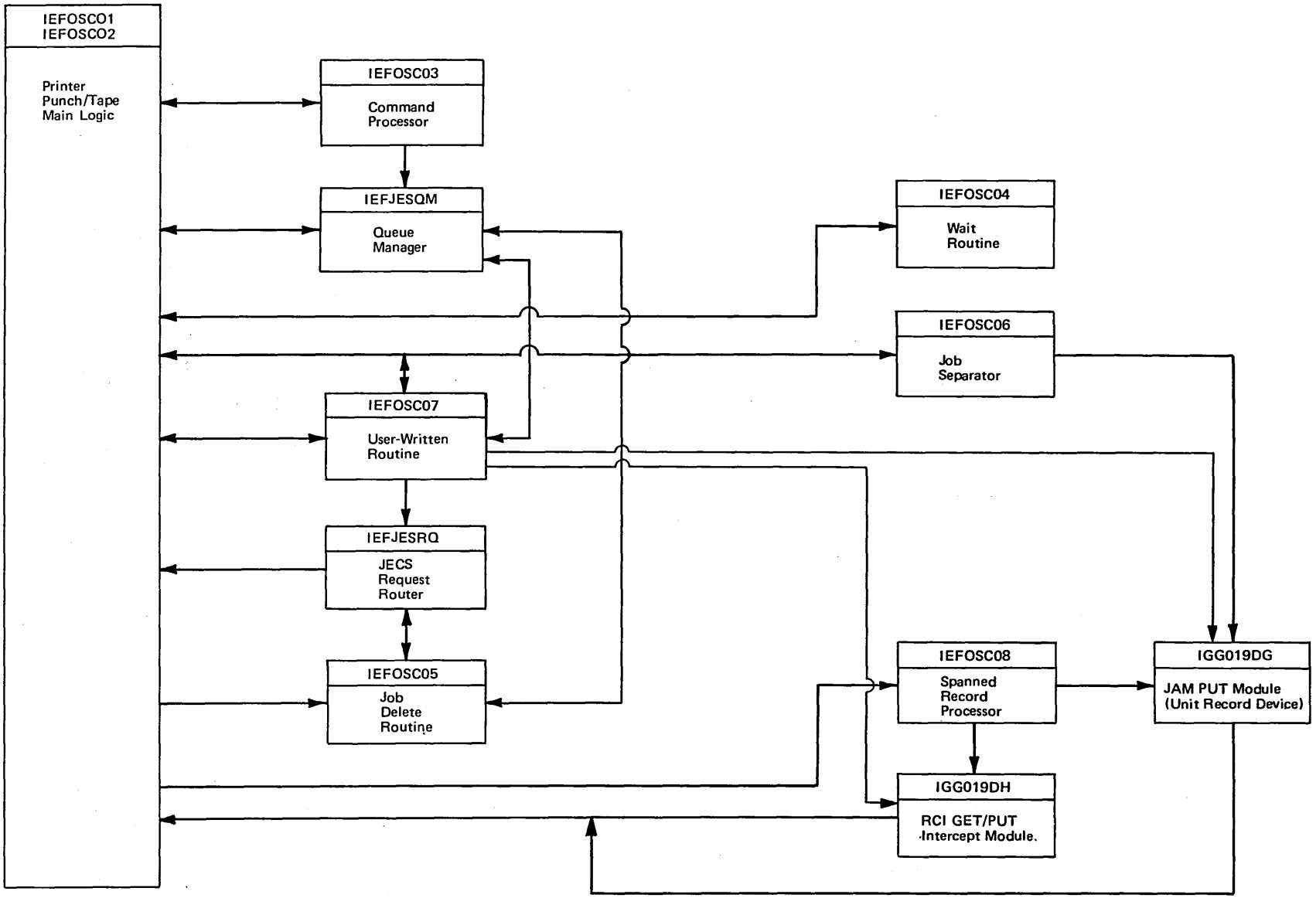


Figure 3-43. JES Writer Task (Part 1 of 7)

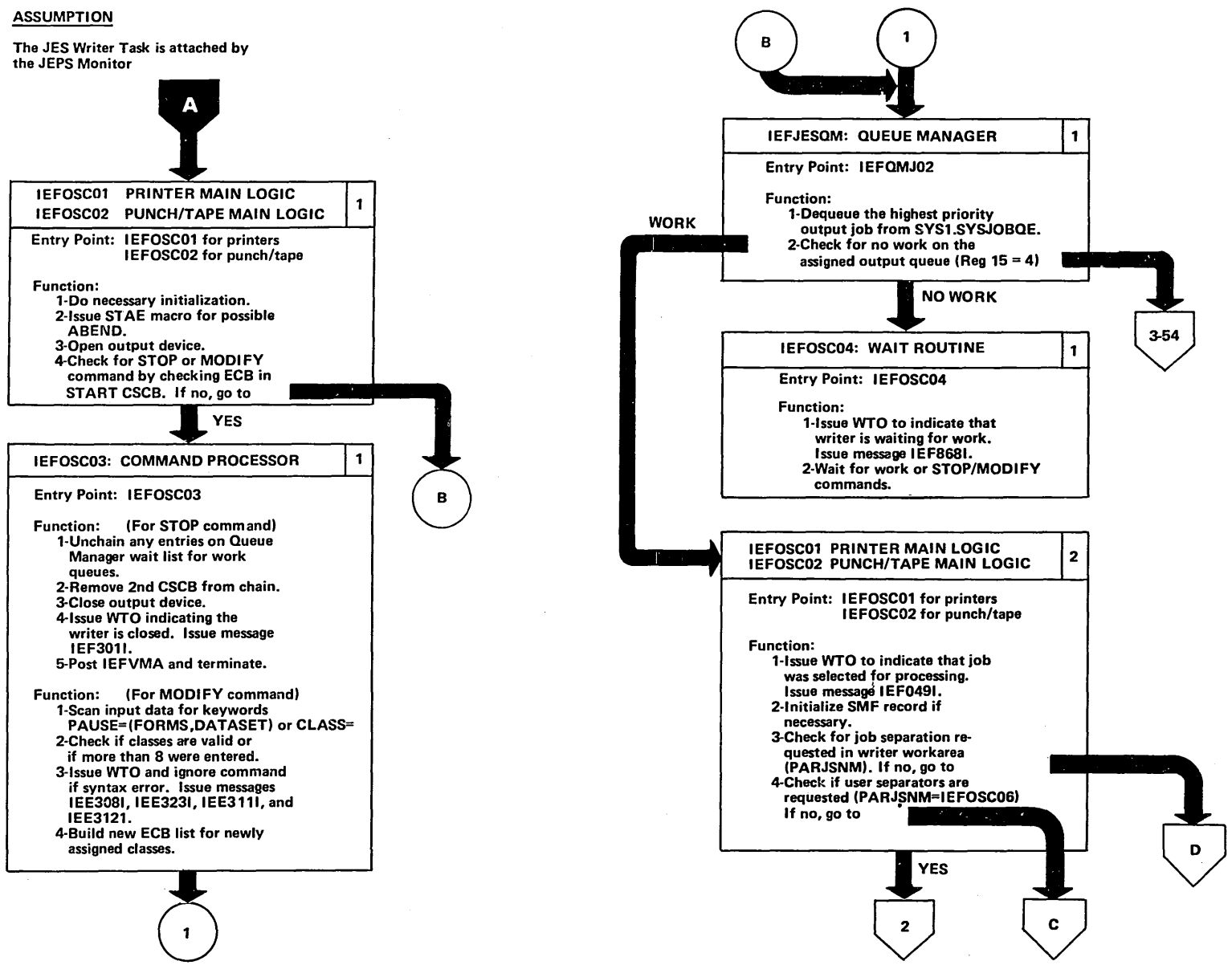


Figure 3-43. JES Writer Task (Part 2 of 7)

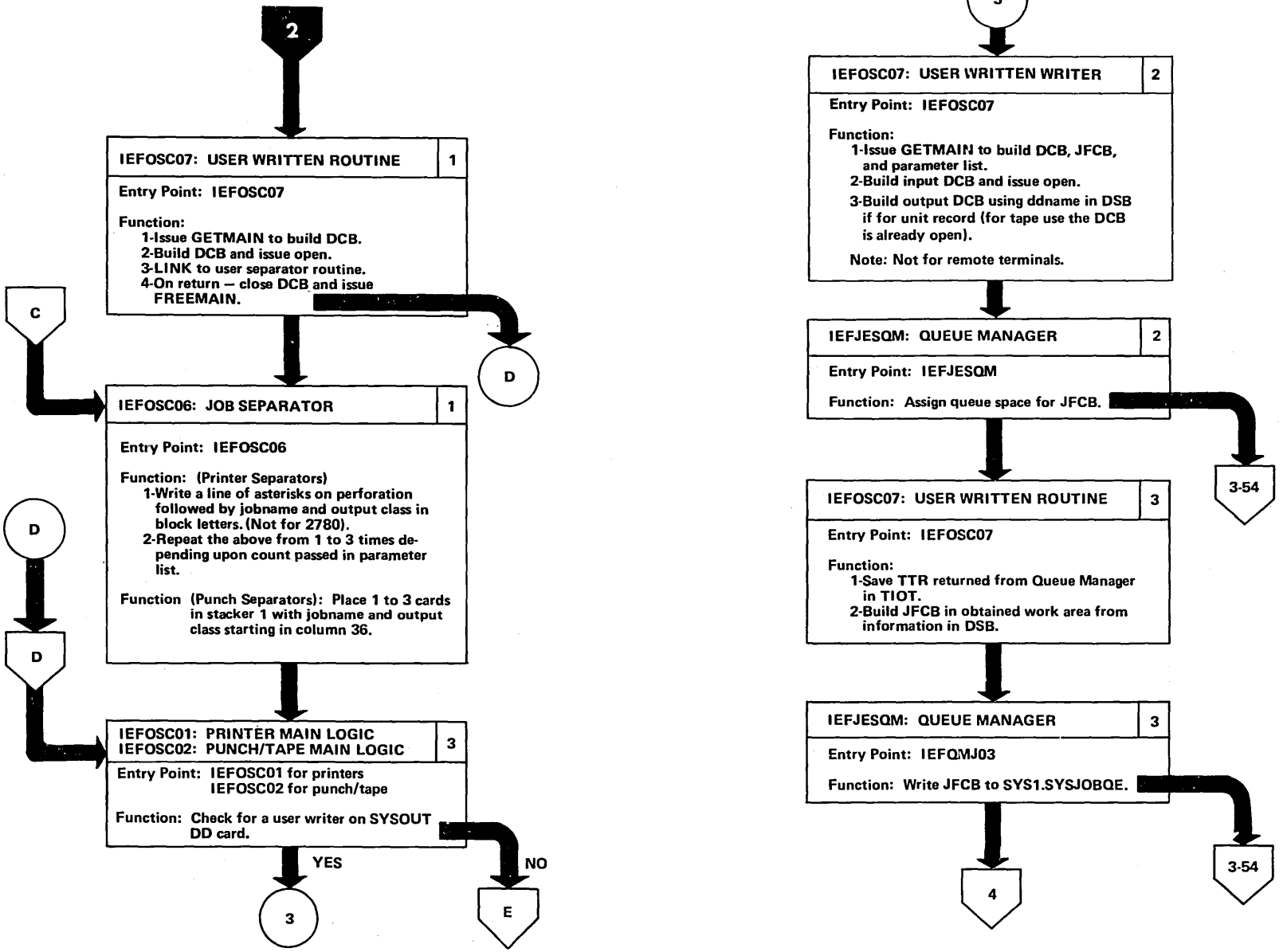


Figure 3-43. JES Writer Task (Part 3 of 7)

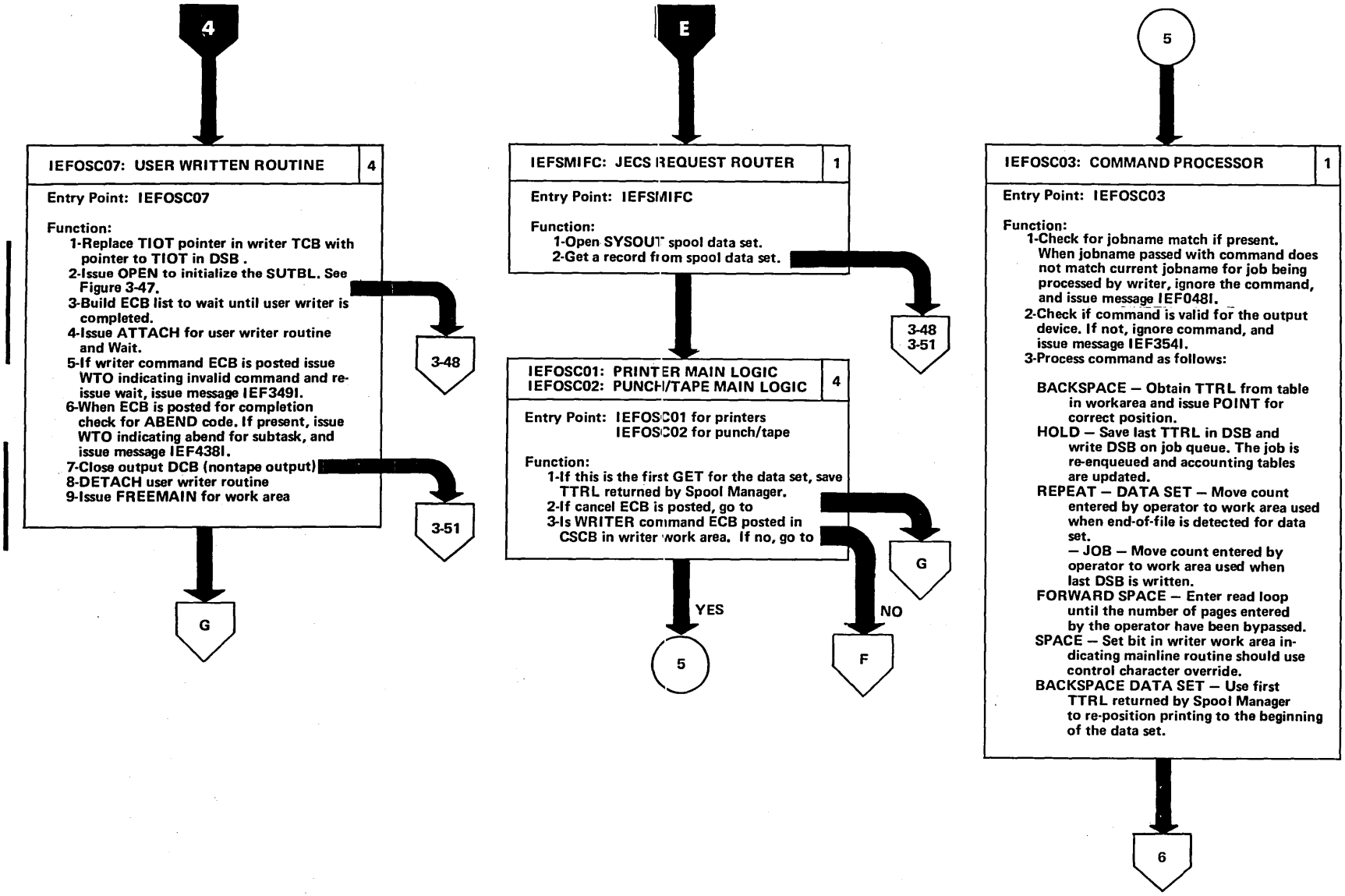


Figure 3-43. JES Writer Task (Part 4 of 7)

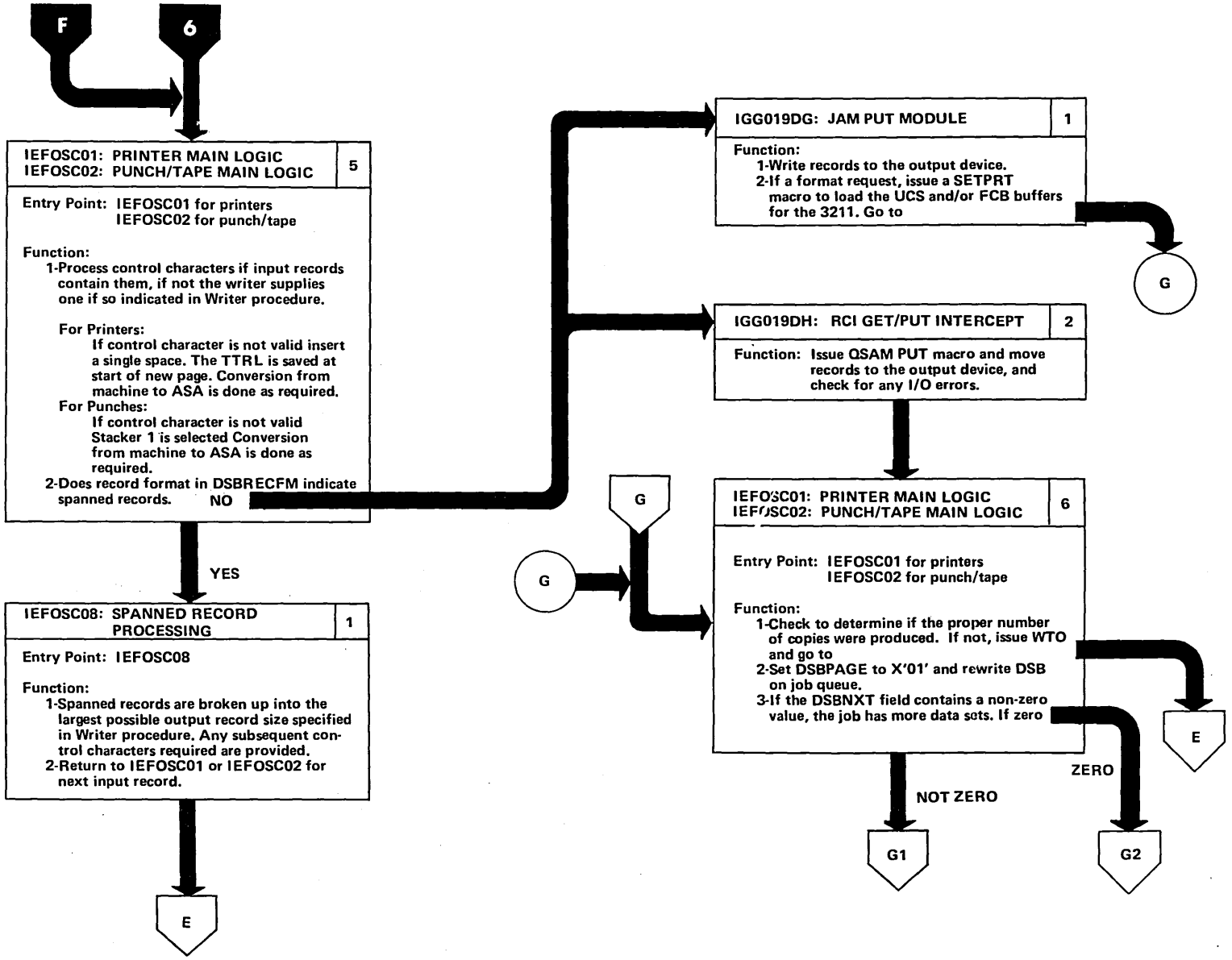


Figure 3-43. JES Writer Task (Part 5 of 7)

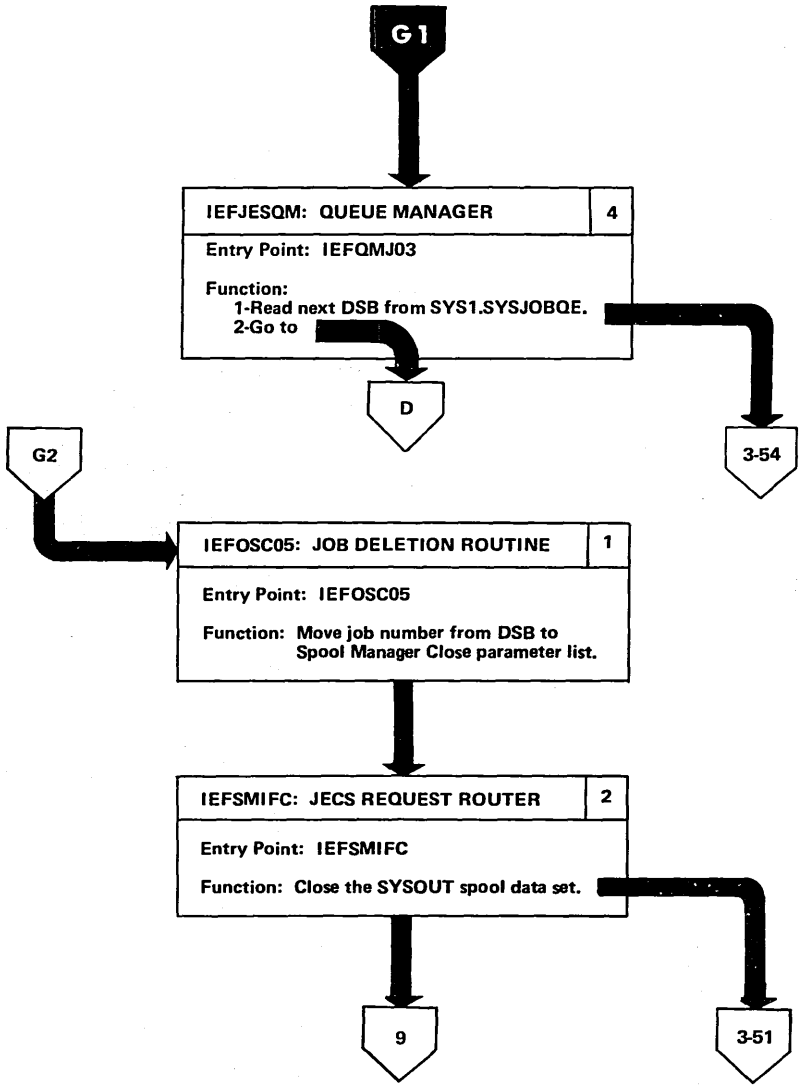


Figure 3-43. JES Writer Task (Part 6 of 7)

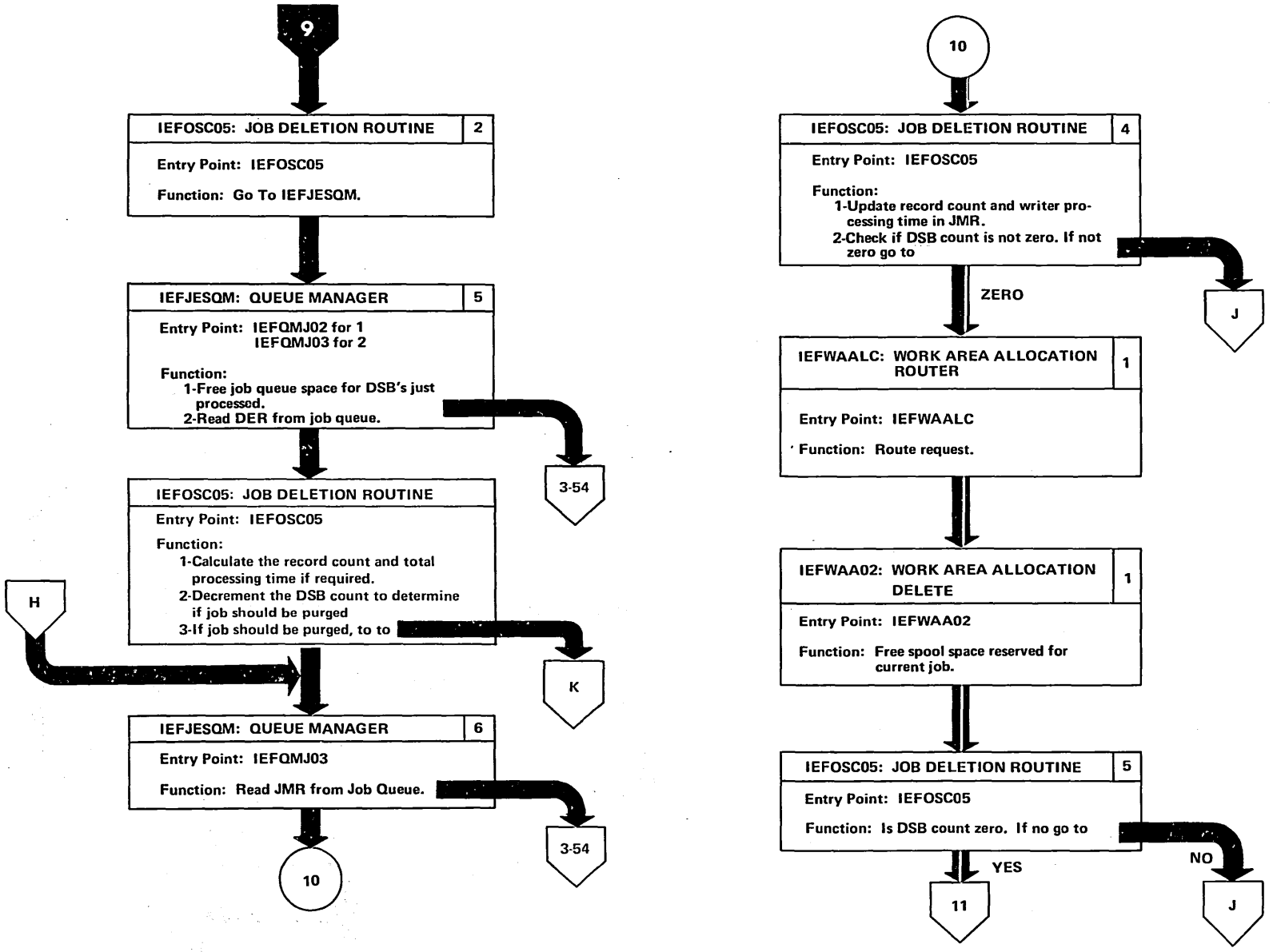


Figure 3-43. JES Writer Task (Part 7 of 7)

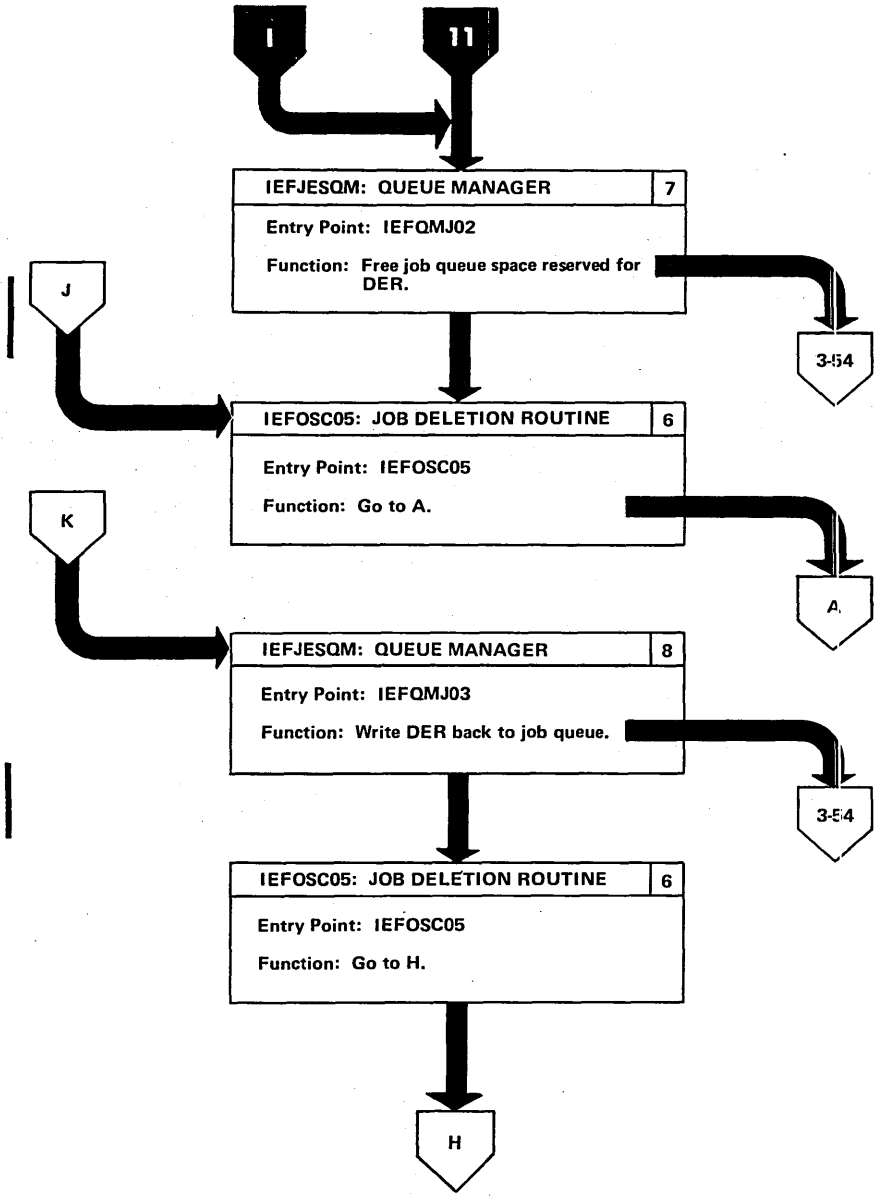


Figure 3-44. Non-Unit Record Device I/O OPEN Processing

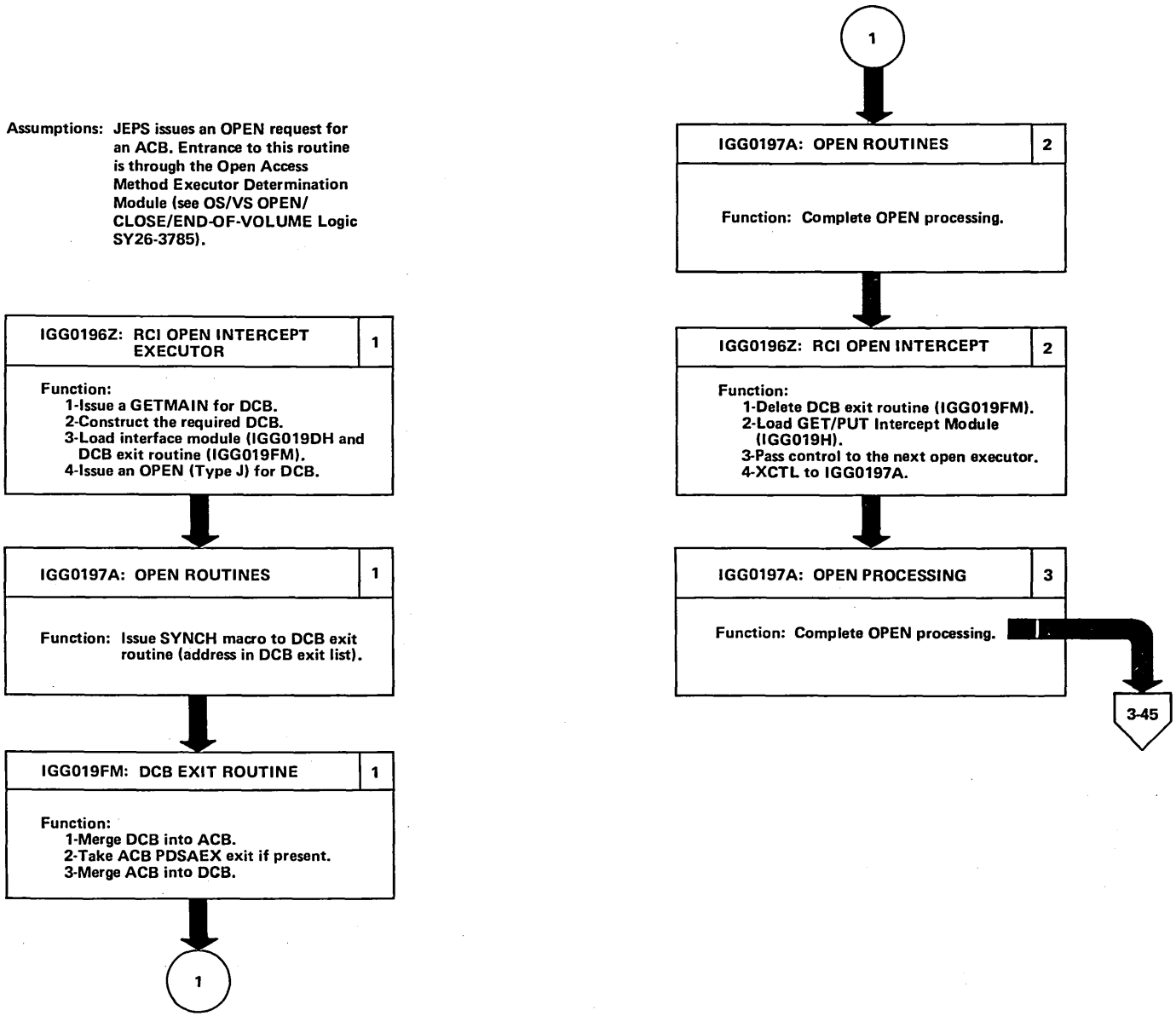


Figure 3-45. JAM OPEN Executors

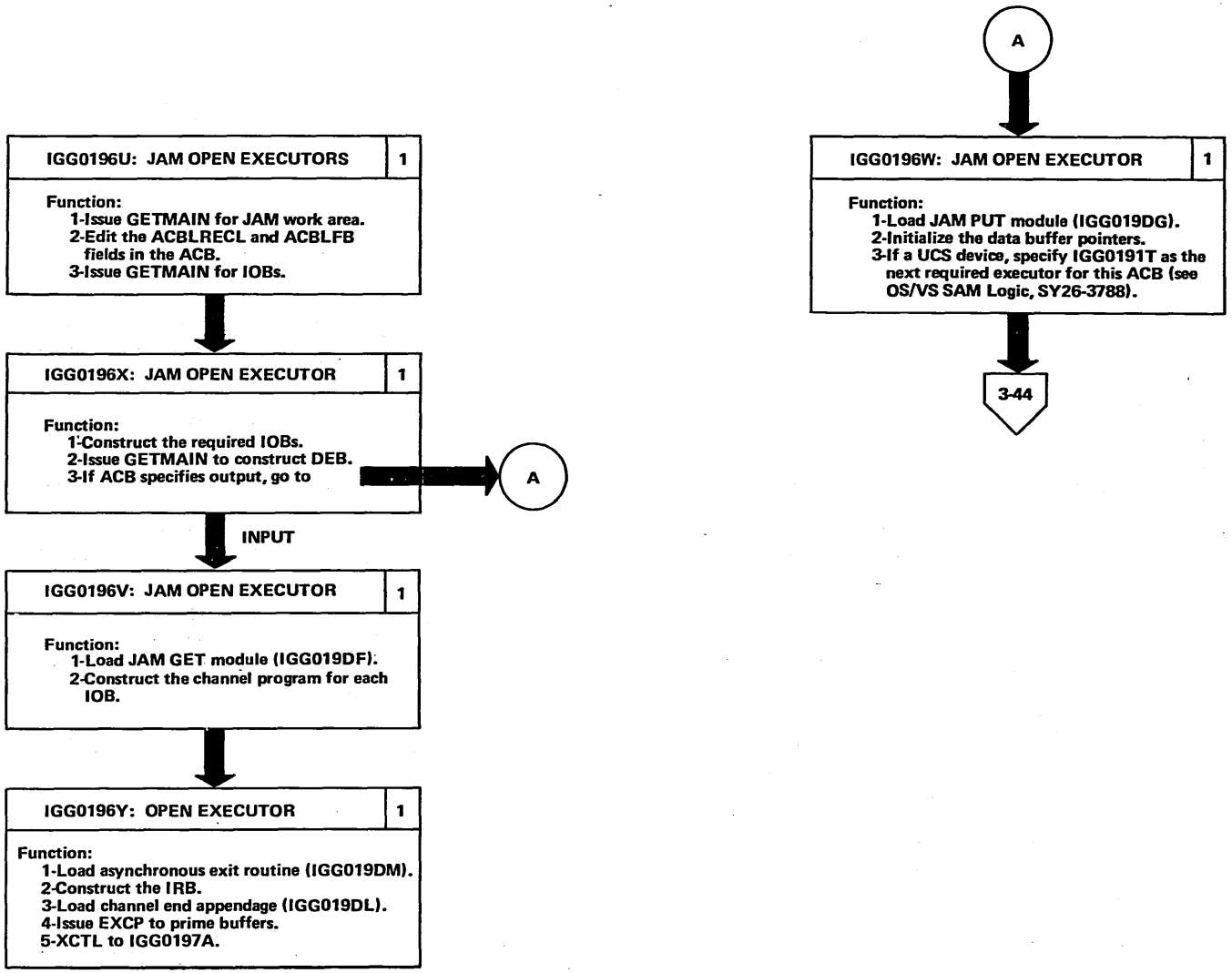


Figure 3-46. SYSIN/SYSOUT JAM CLOSE Processing

Assumptions: JEPS has issued a close for an ACB. Entrance to this routine is through the Close Access Method Executive Determination routine (see OS/VS Open/Close/End-of-Volume Logic, SY26-3785).

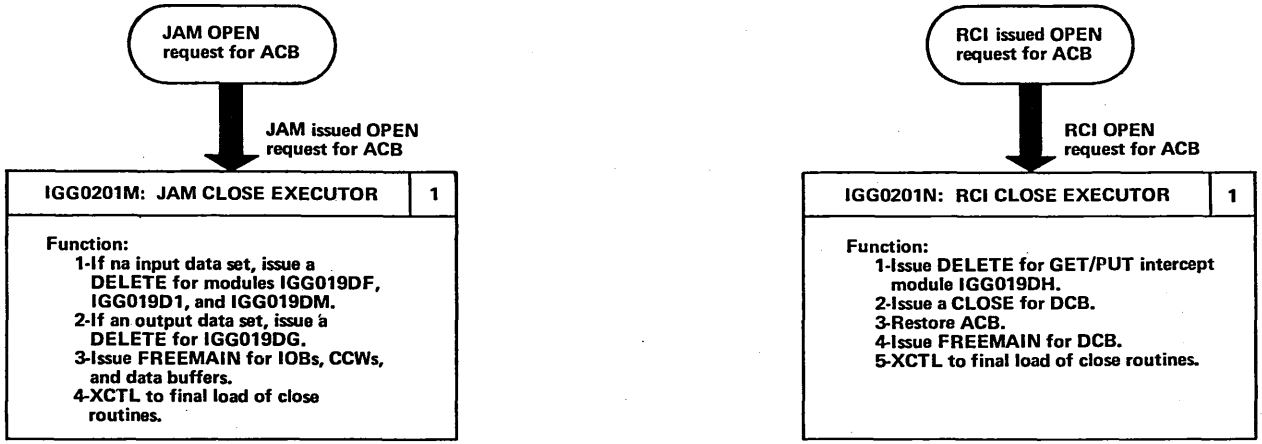


Figure 3-47. JECS Request Router Interconnection Module Diagram (Part 1 of 4)

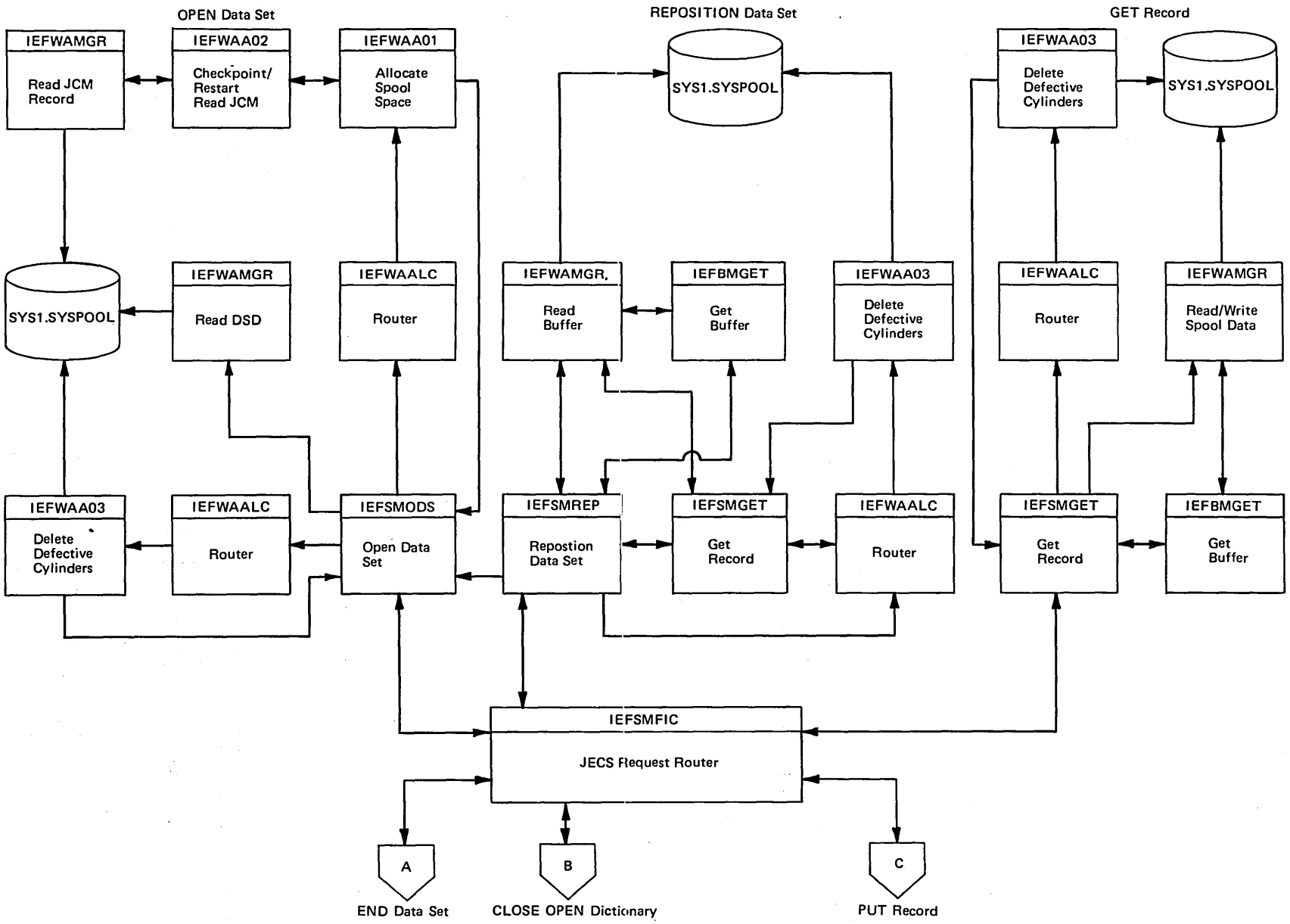
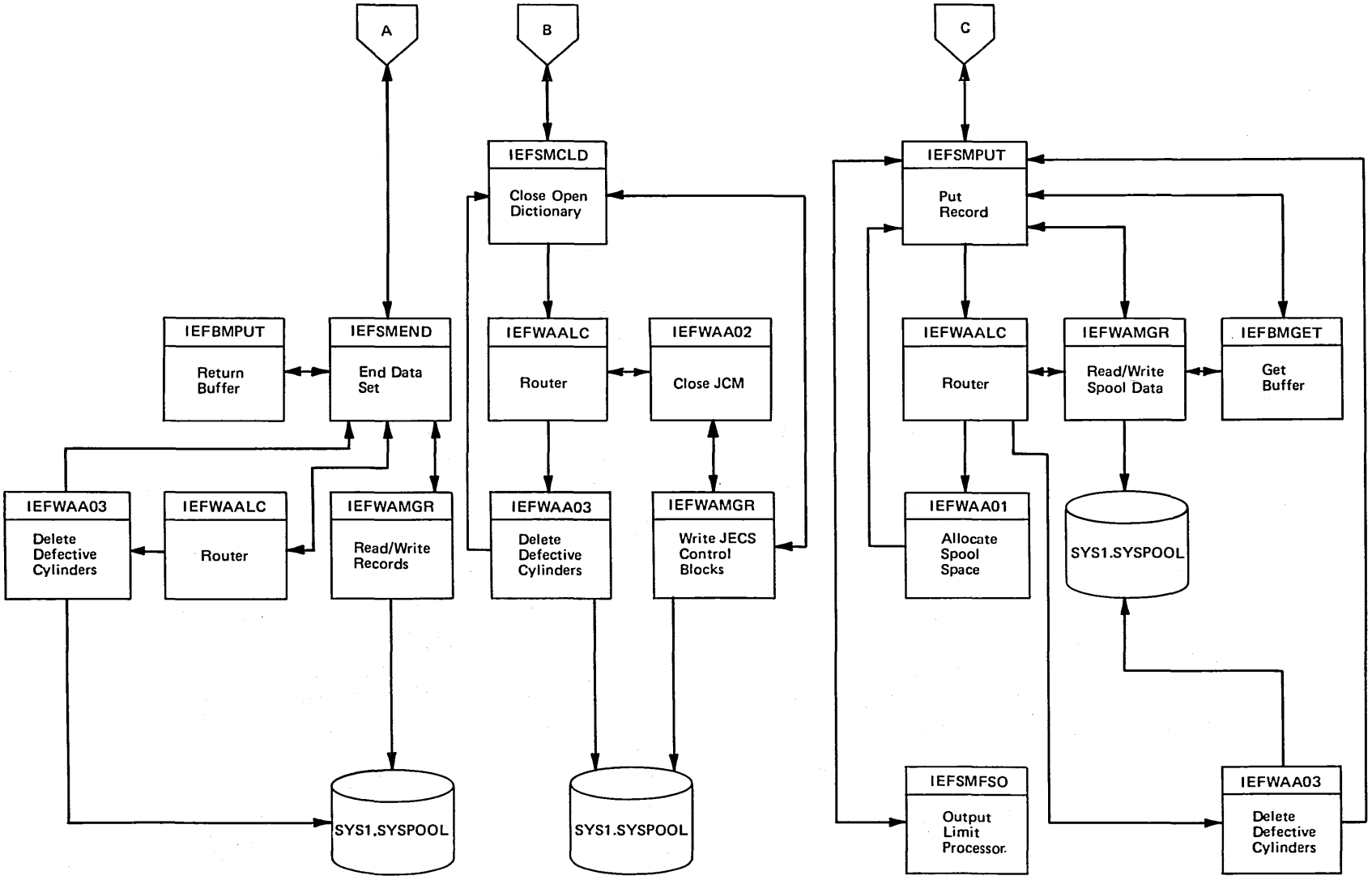


Figure 3-47. JECS Request Router Interconnection Module Diagram (Part 2 of 4)



END Data Set

CLOSE OPEN Dictionary

PUT Record

Figure 3-47. JECS Request Router Interconnection Module Diagram (Part 3 of 4)

Modules outside of JECS that interface directly with modules within JECS for the indicated function:

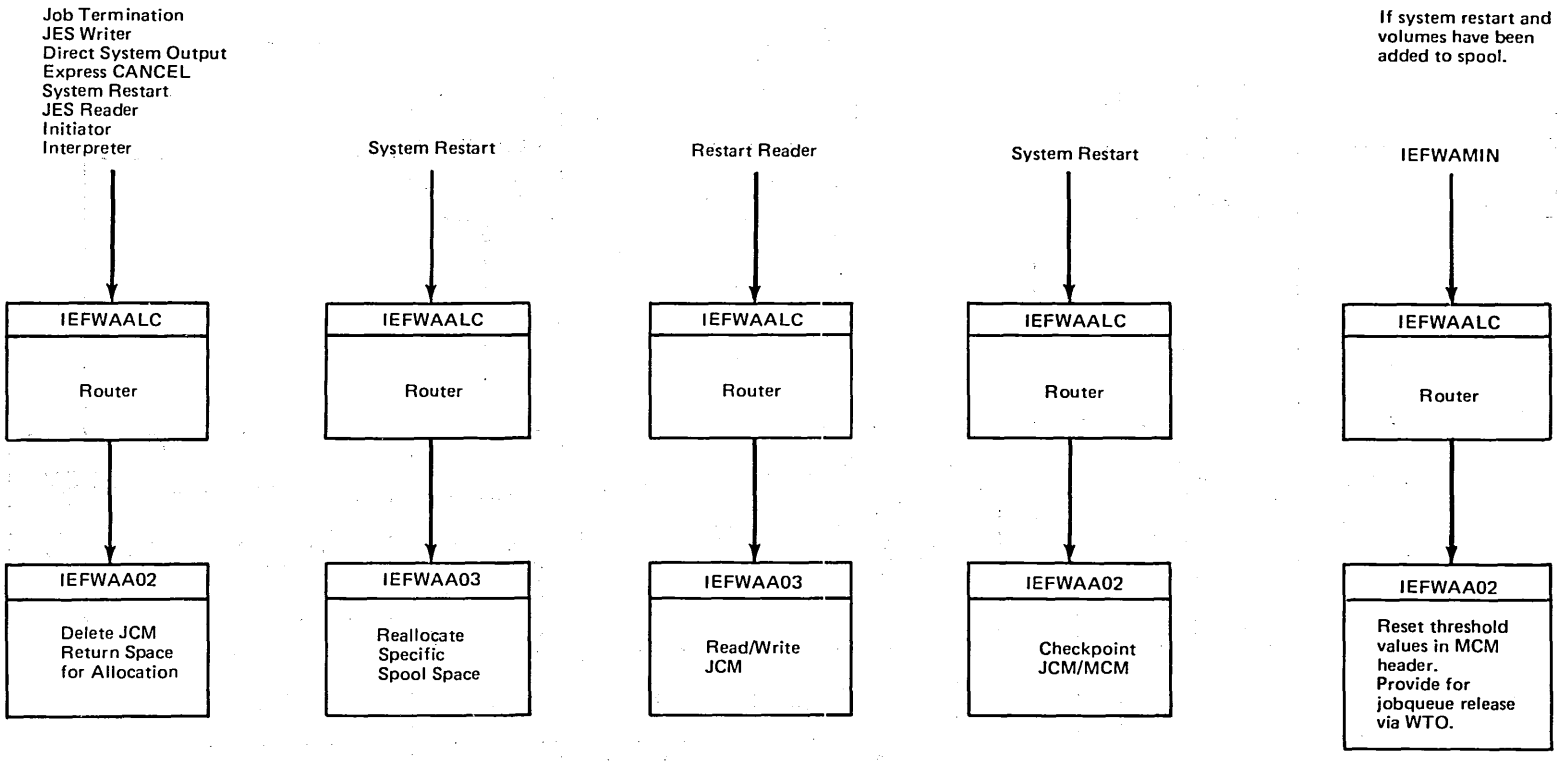


Figure 3-47. JECS Request Router Interconnection Module Diagram (Part 4 of 4)

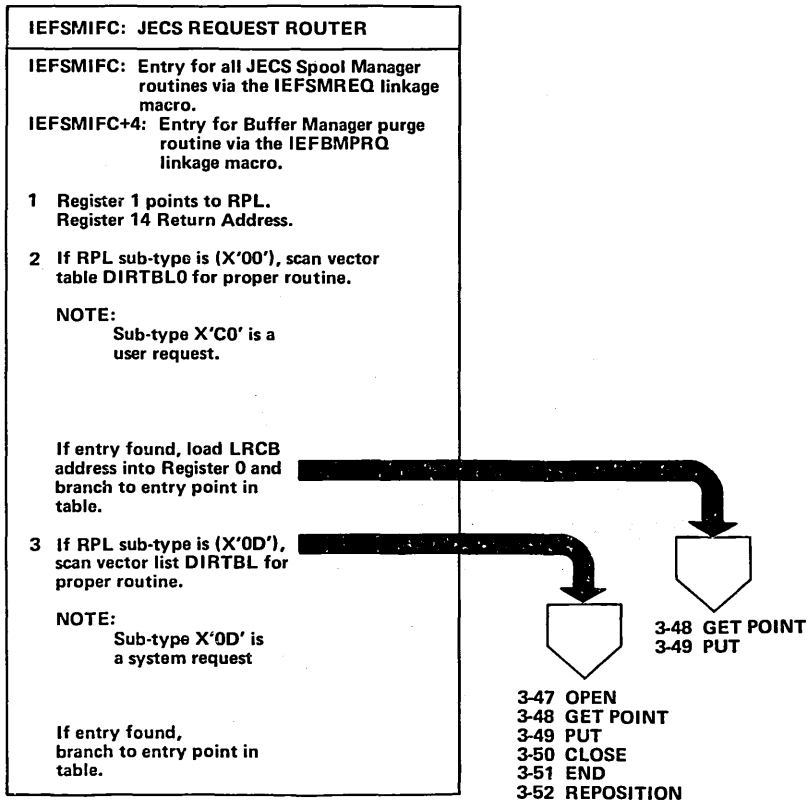


Figure 3-48. OPEN Data Set (Part 1 of 2)

ASSUMPTION

Normal entrance is through the macro IEFSMREQ which branches to IEFSMIFC (Request Router) which in turn routes the request to IEFSMODS. Certain special functions such as repositioning of a data set (IEFSMREP) enter IEFSMODS directly. This functional flow diagram assumes that there are no error conditions present.

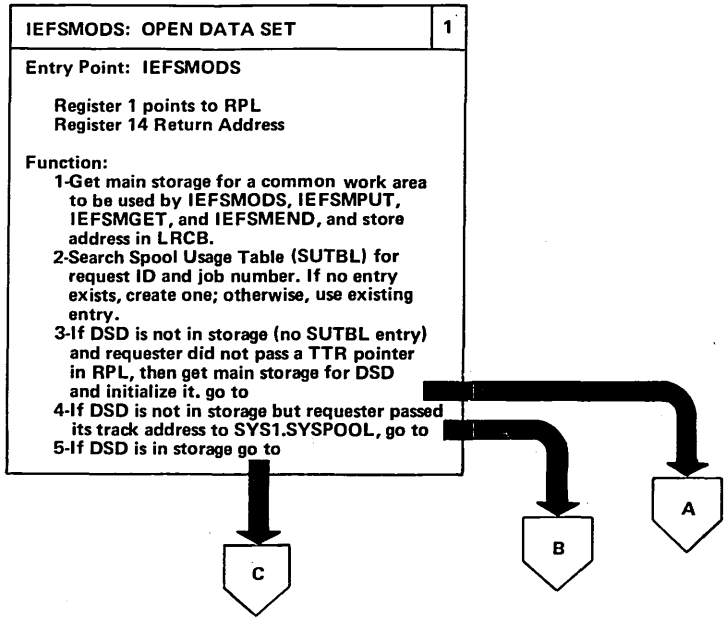


Figure 3-48. OPEN Data Set (Part 2 of 2)

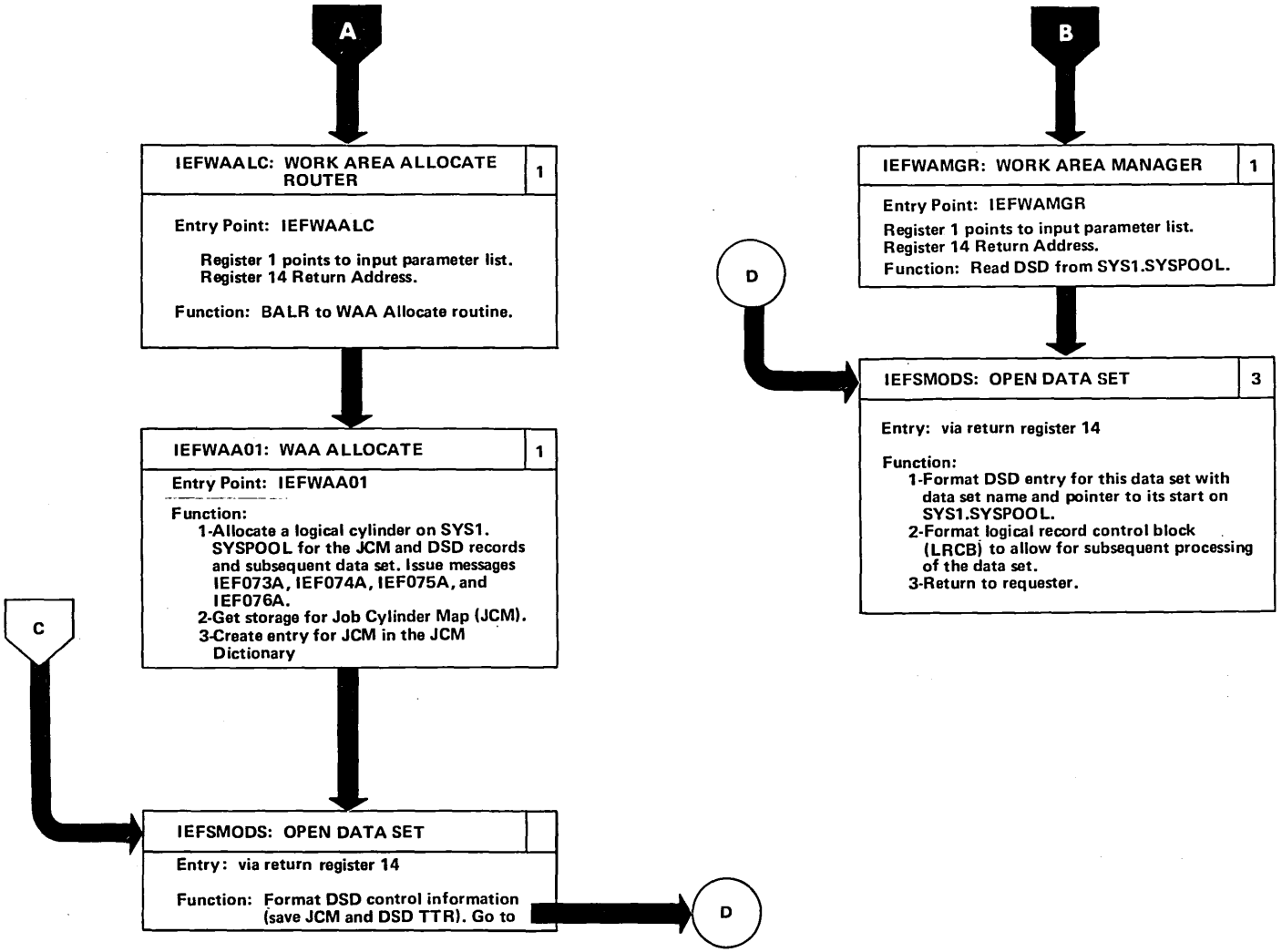


Figure 3-49. GET Record and POINT

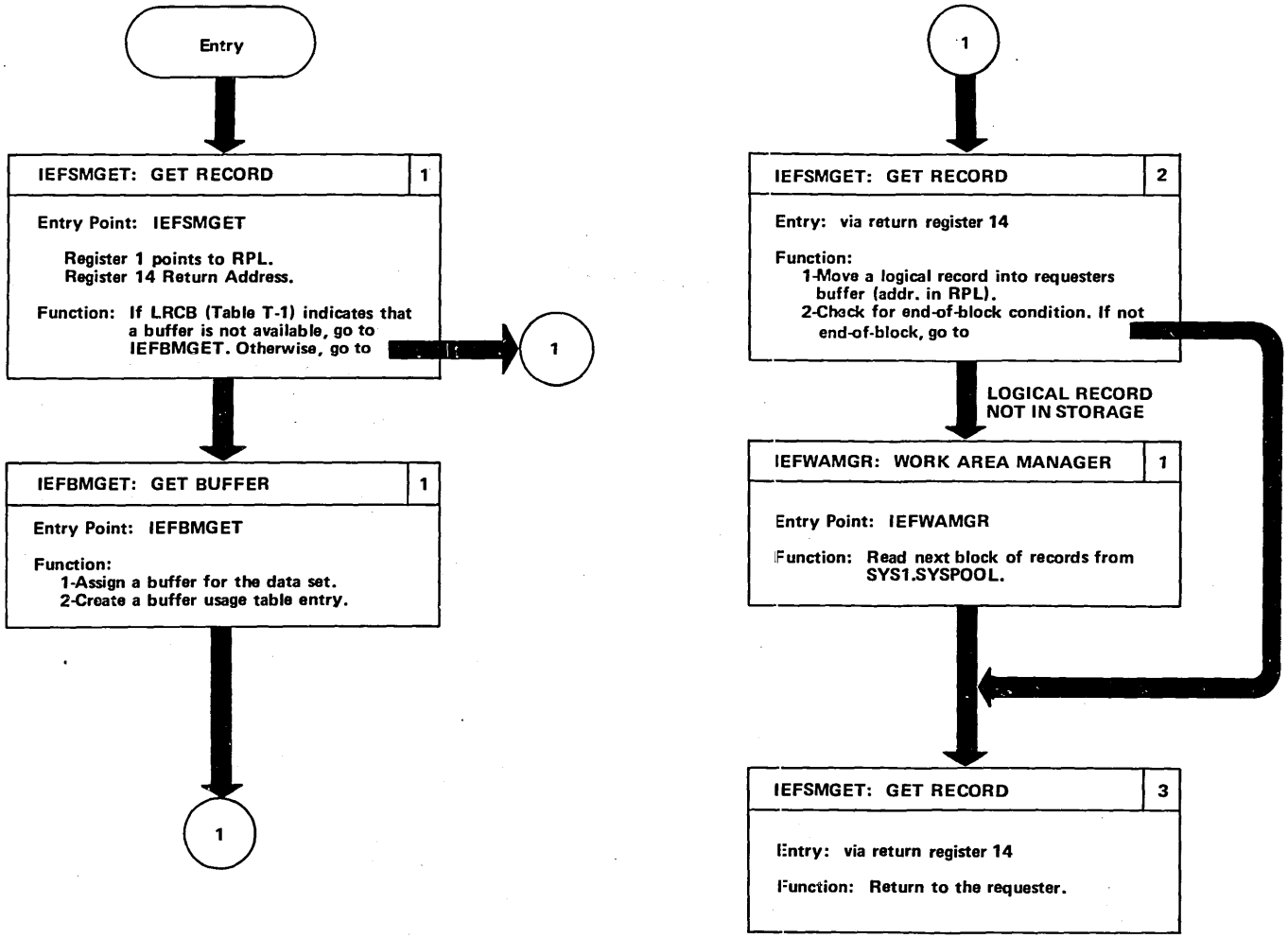


Figure 3-50. PUT Record (Part 1 of 2)

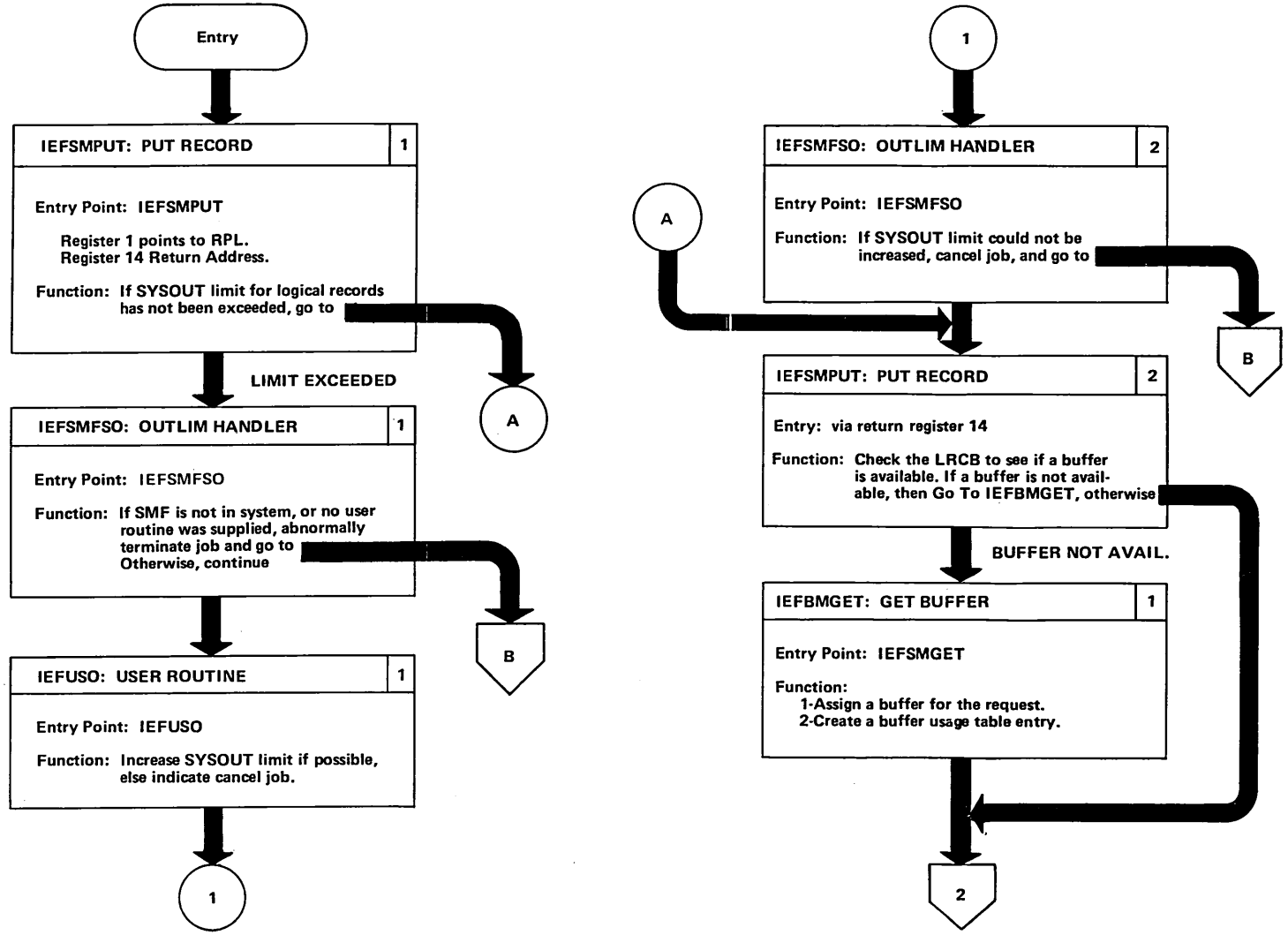


Figure 3-50. PUT Record (Part 2 of 2)

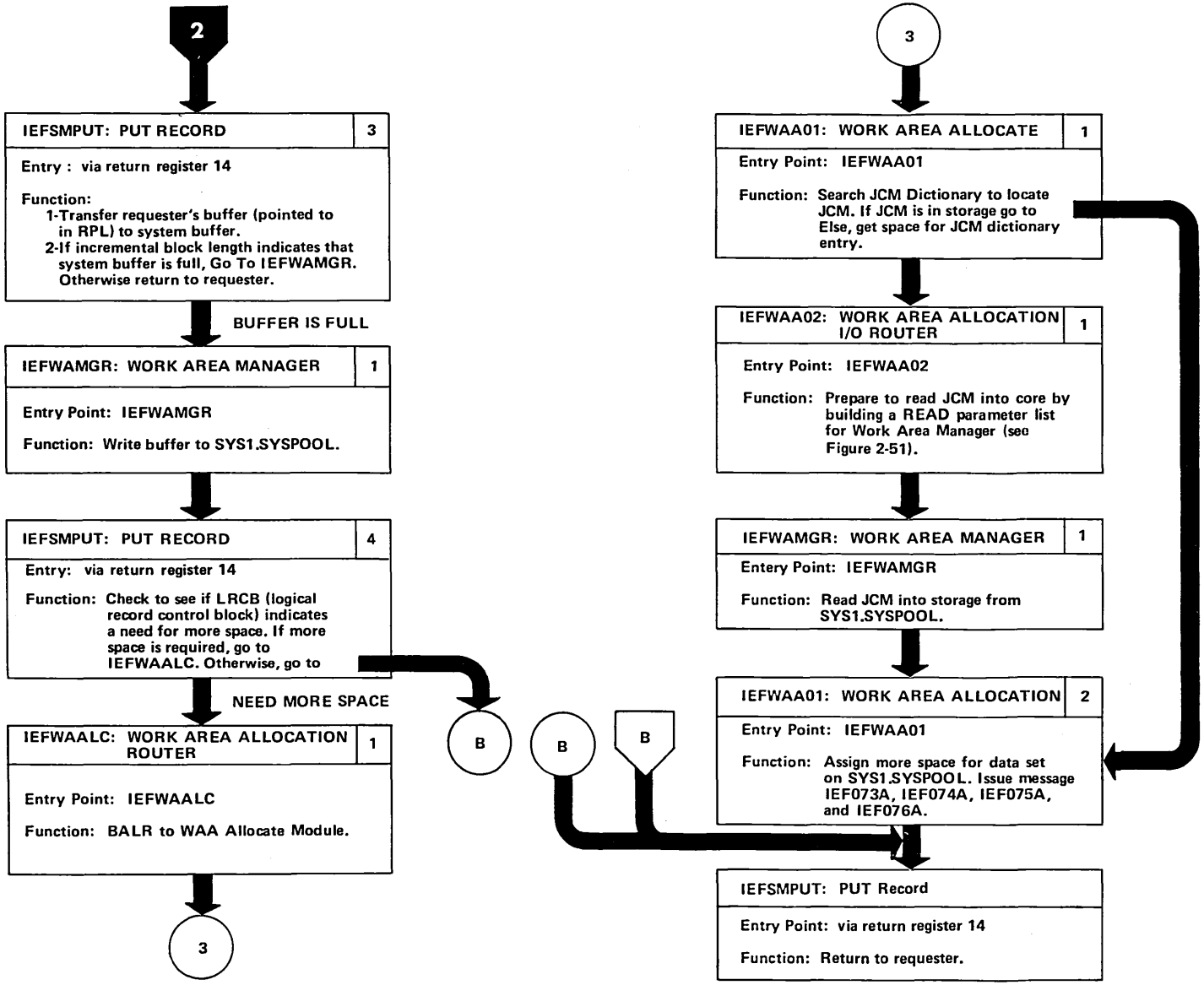


Figure 3-51. CLOSE OPEN Dictionary Processing (Part 1 of 2)

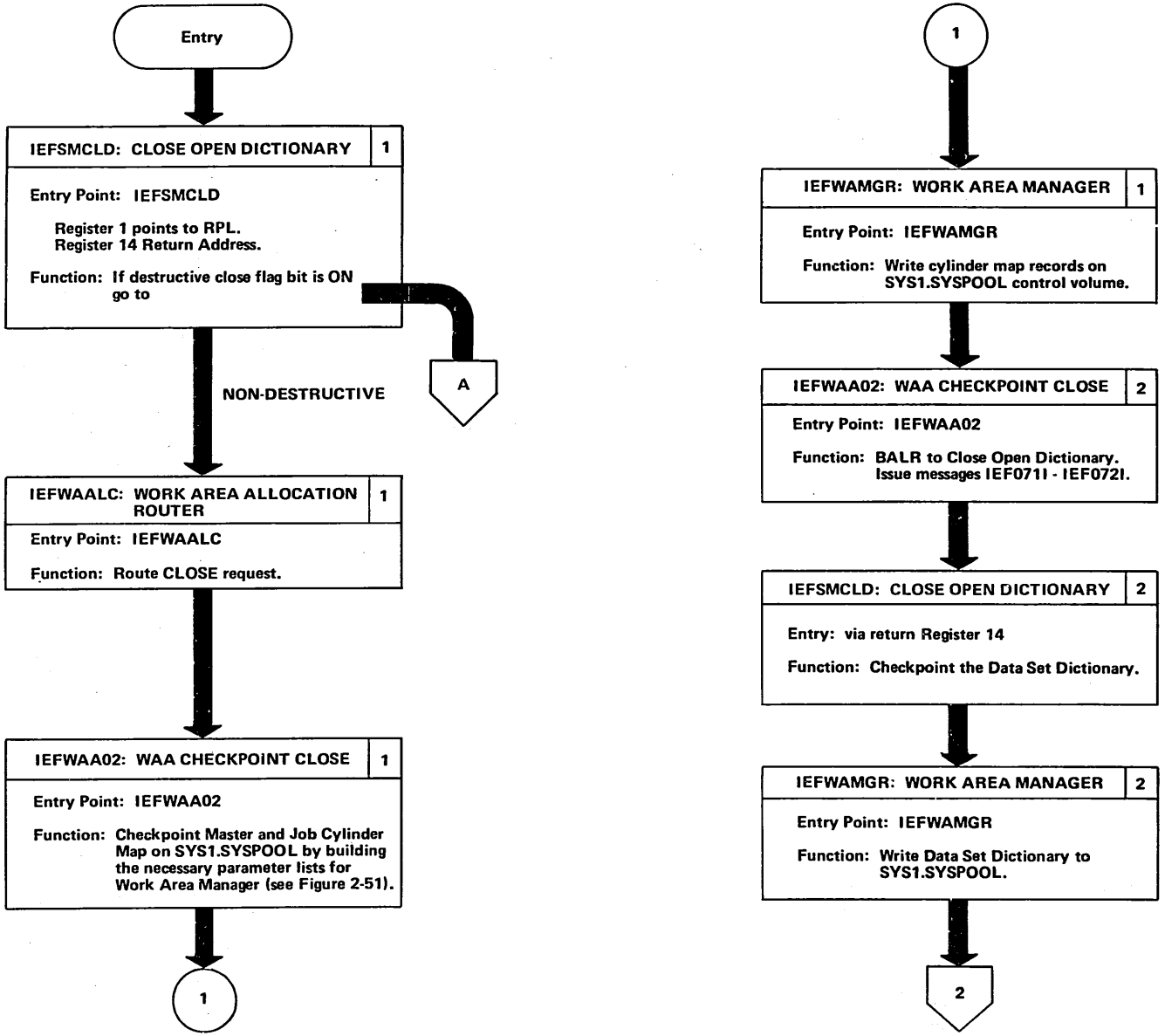


Figure 3-51. CLOSE OPEN Dictionary Processing (Part 2 of 2)

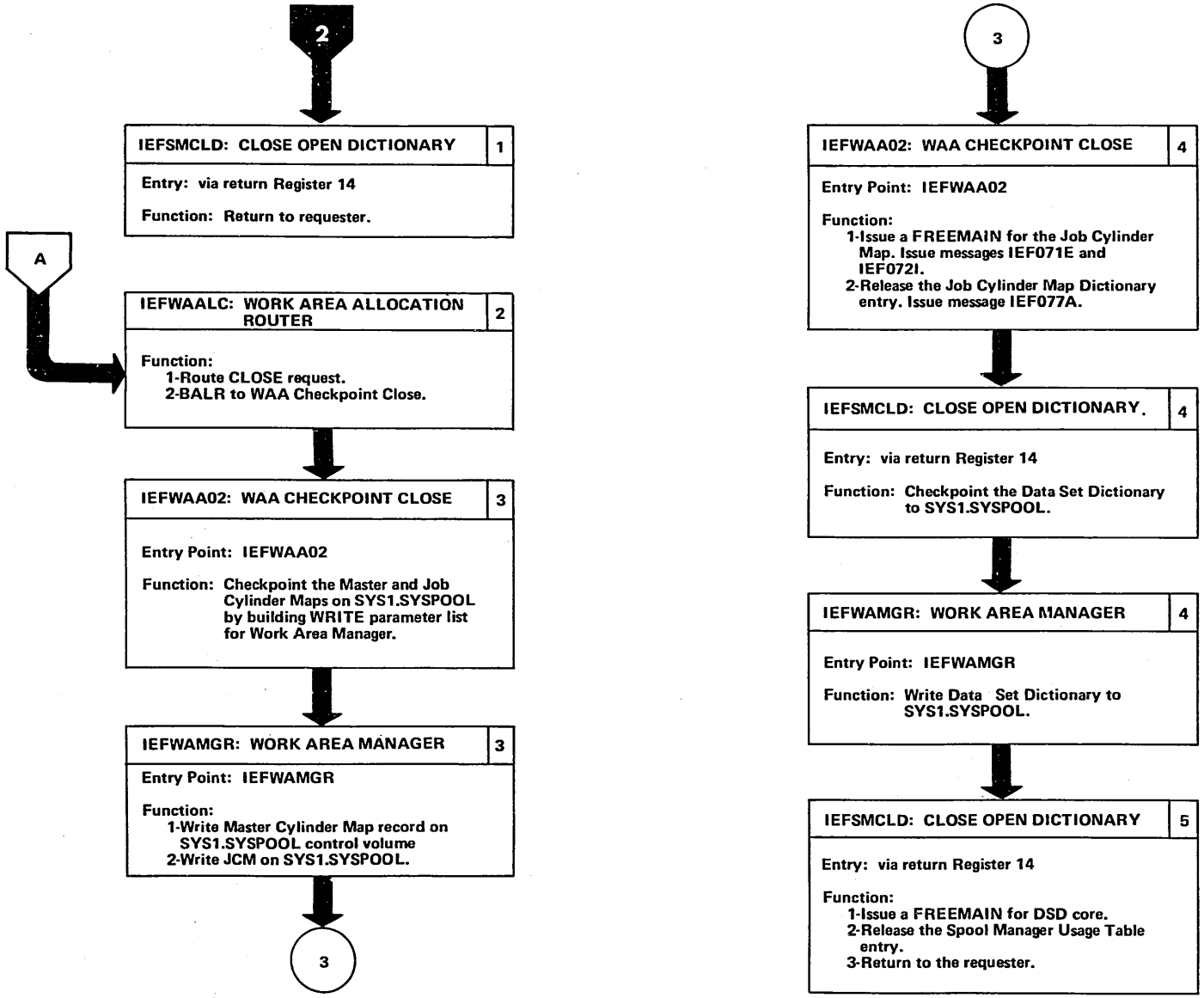


Figure 3-52. END Data Set

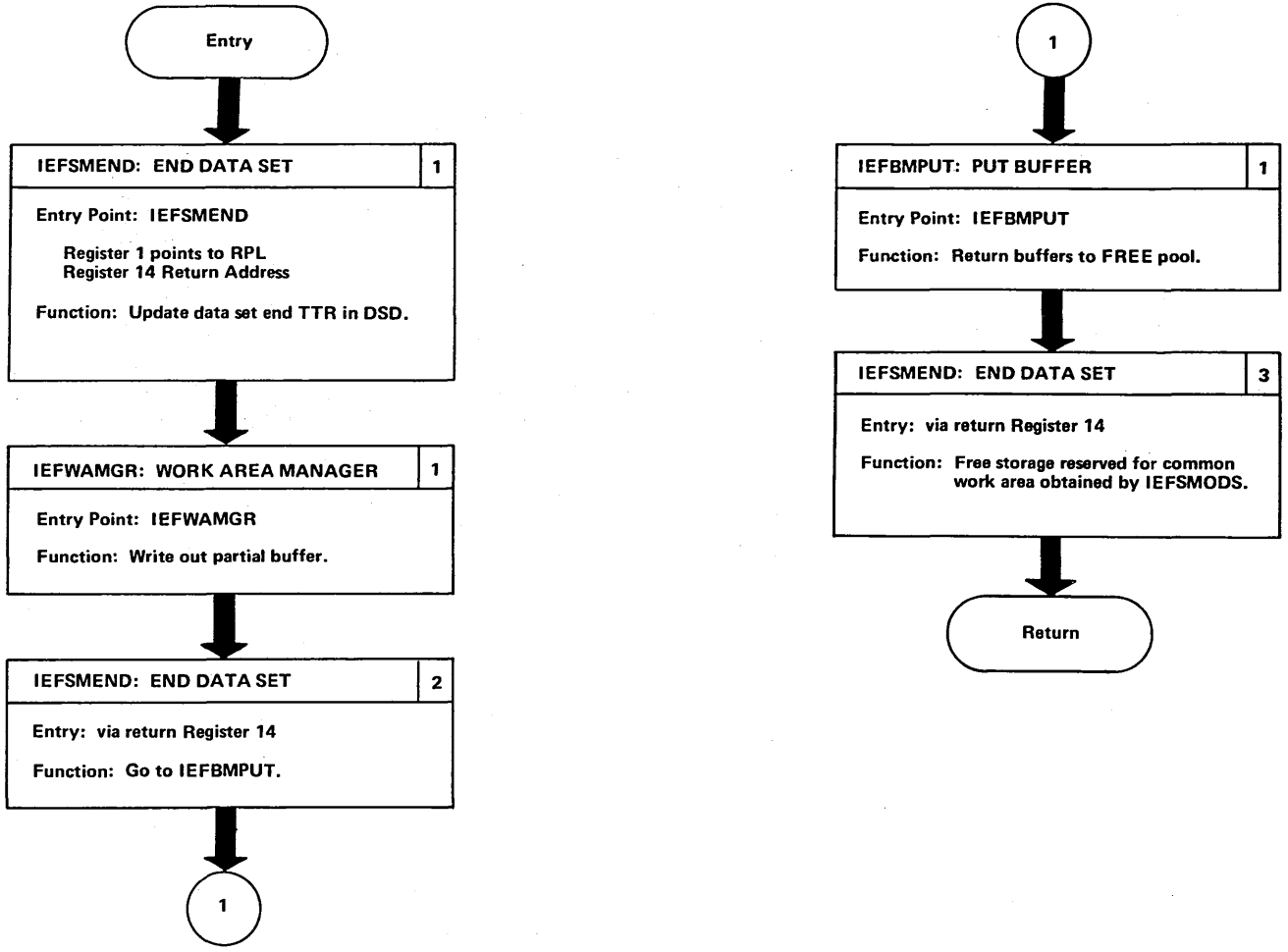
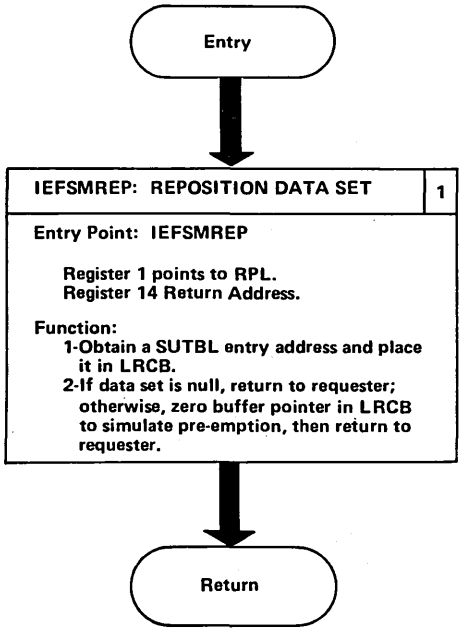


Figure 3-53. REPOSITION a Data Set (Part 1 of 3)

ASSUMPTION

For automatic SYSIN restart only



ASSUMPTION

For automatic SYSOUT restart only

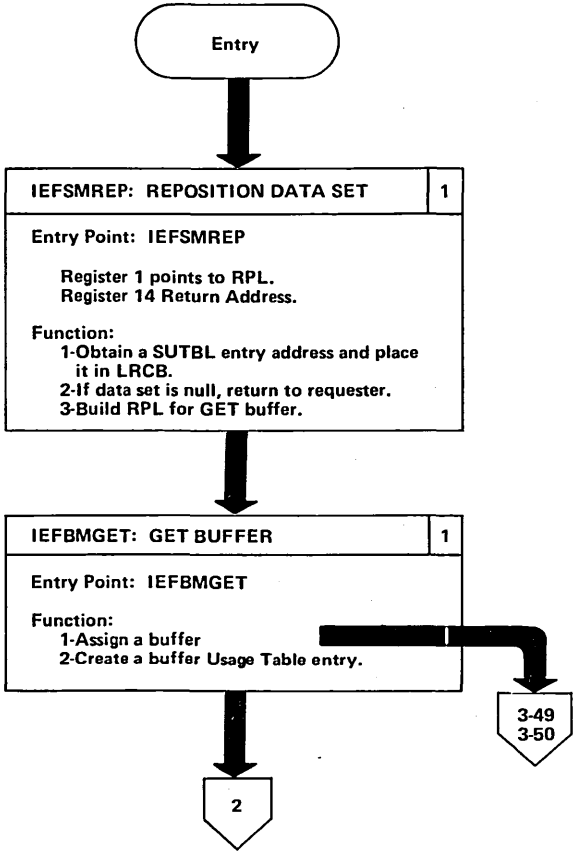


Figure 3-53. REPOSITION a Data Set (Part 2 of 3)

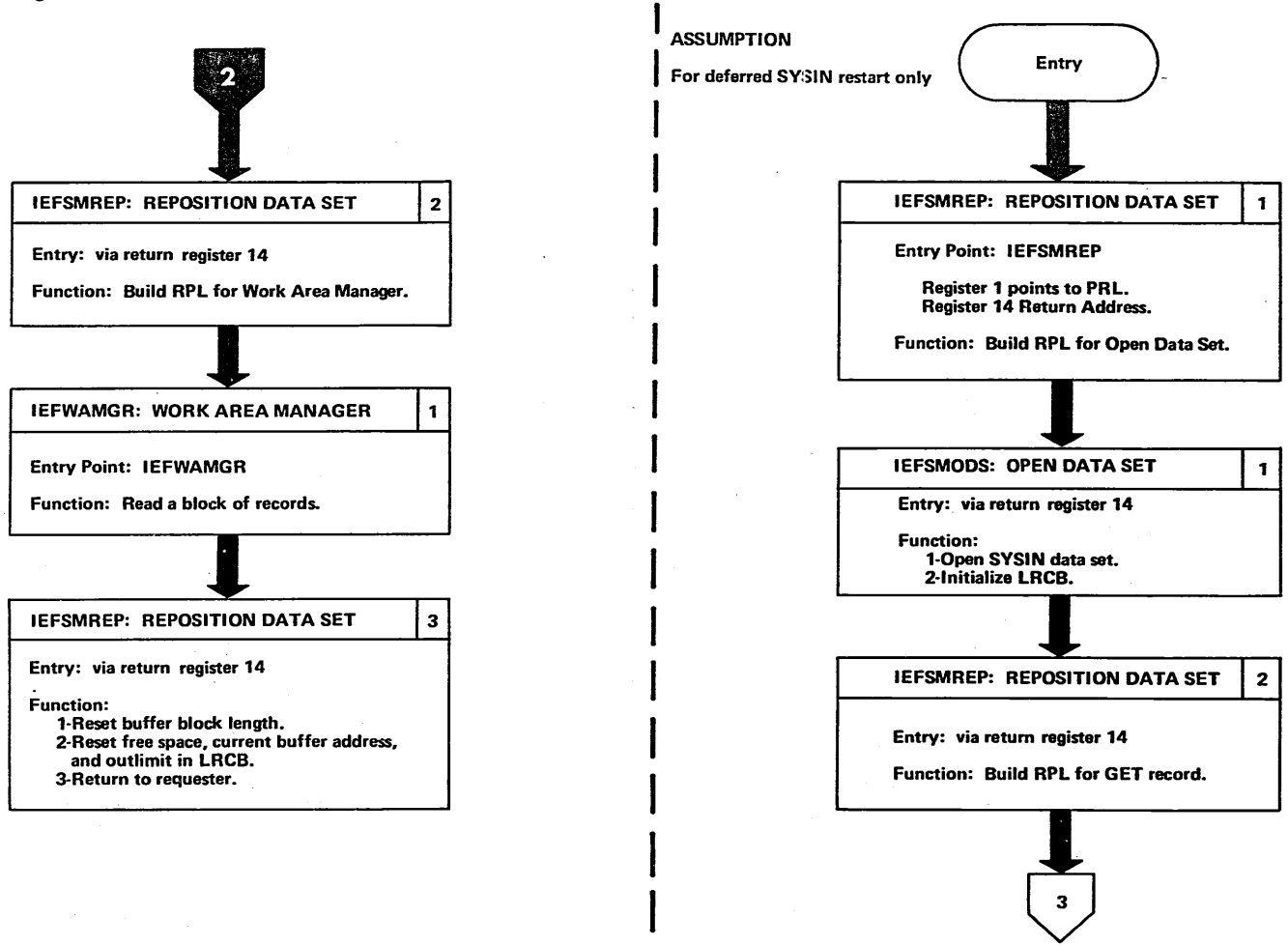
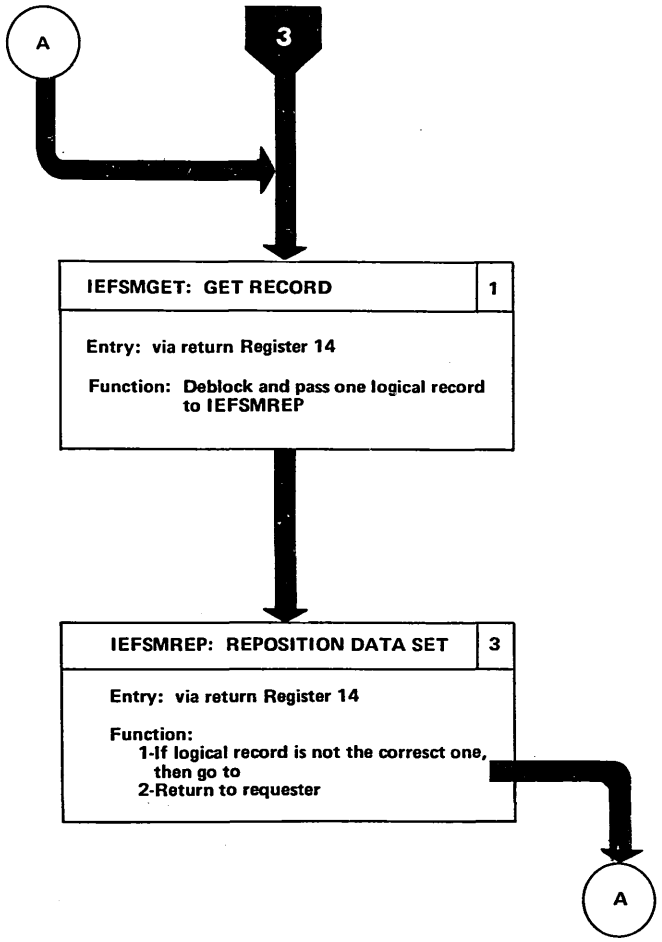


Figure 3-53. REPOSITION a Data Set (Part 3 of 3)



ASSUMPTION

For deferred SYSOUT restart only

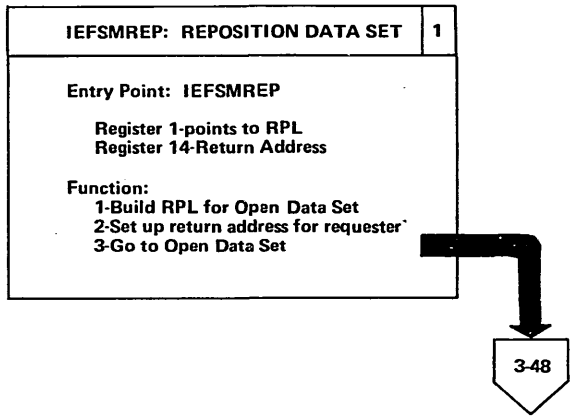


Figure 3-54. Queue Manager (Part 1 of 5)

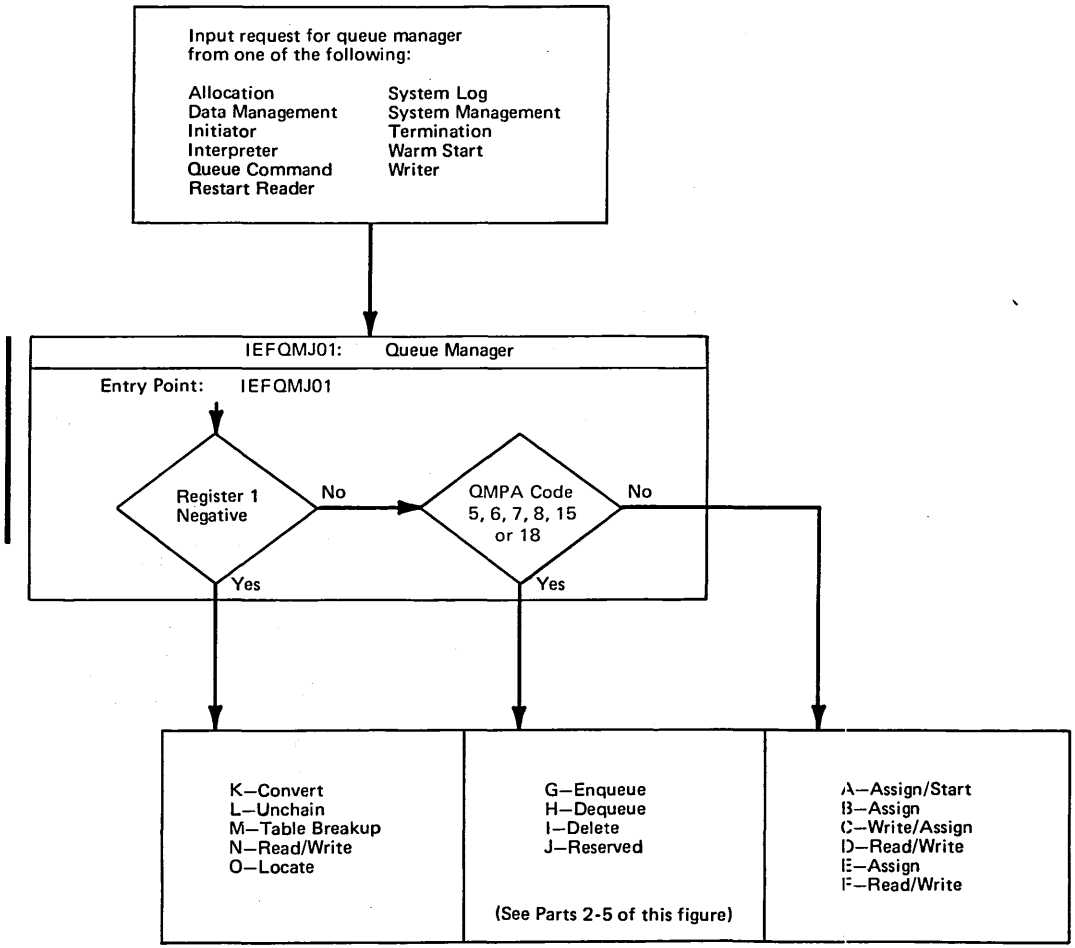


Figure 3-54. Queue Manager (Part 2 of 5)

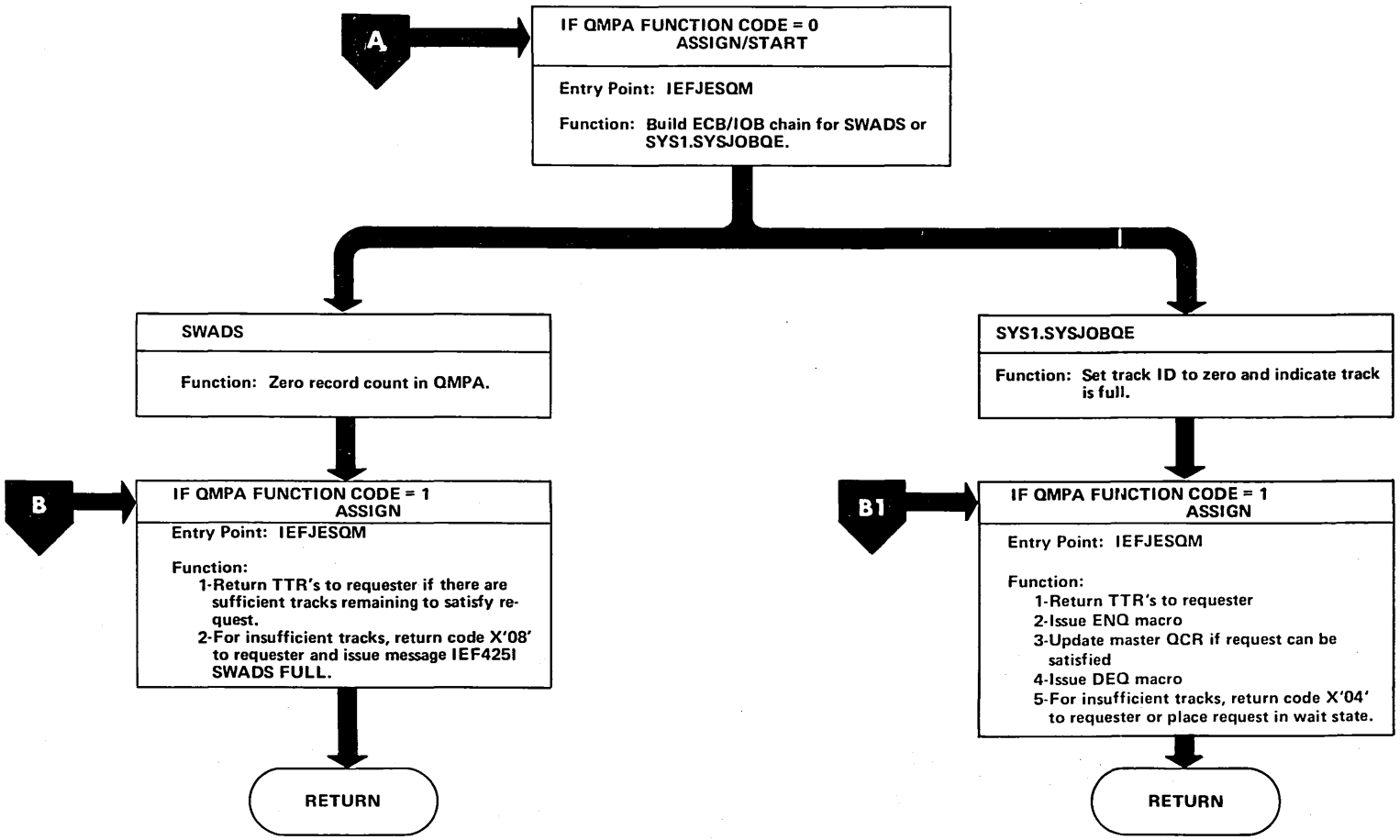


Figure 3-54. Queue Manager (Part 3 of 5)

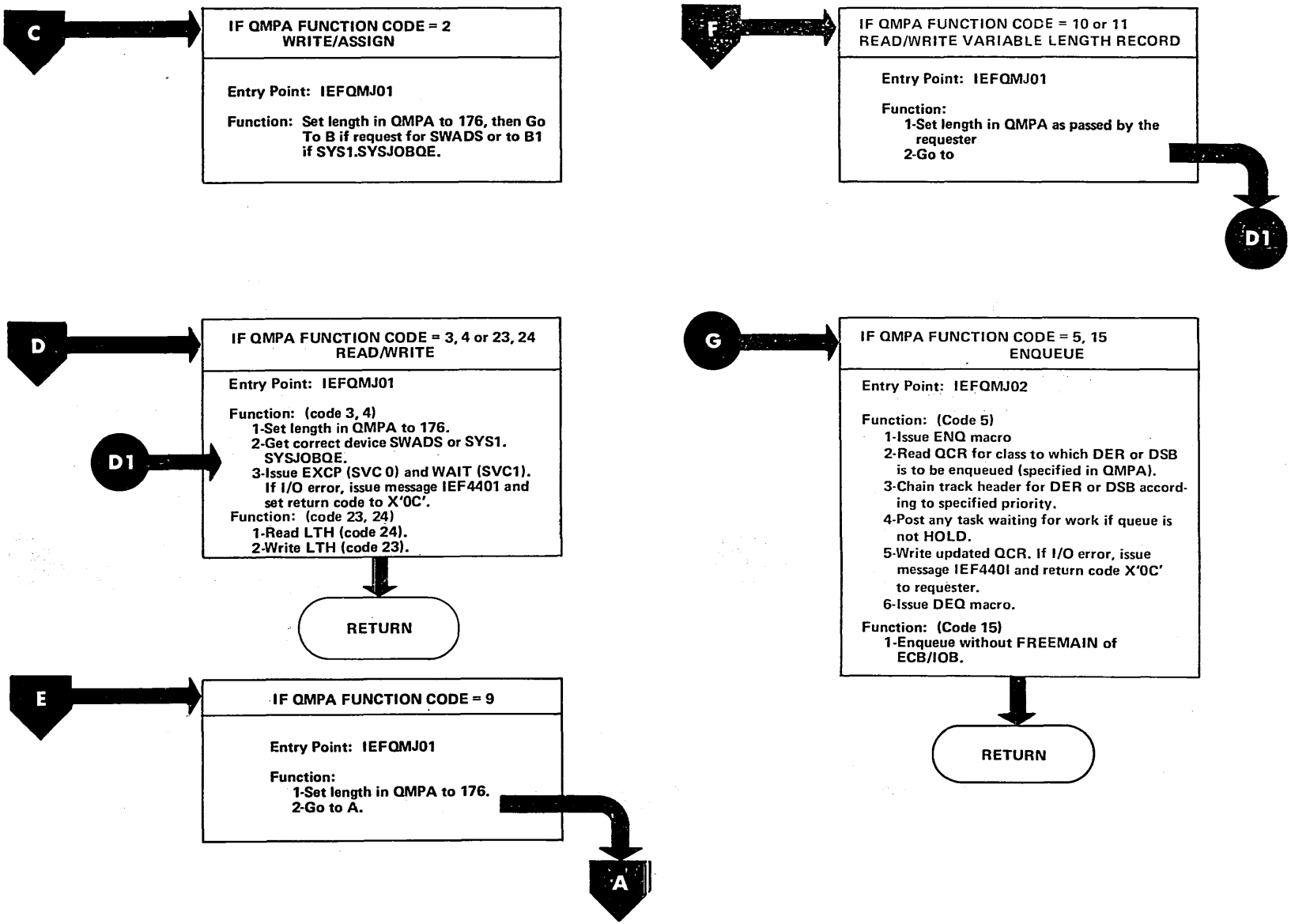


Figure 3-54. Queue Manager (Part 4 of 5)

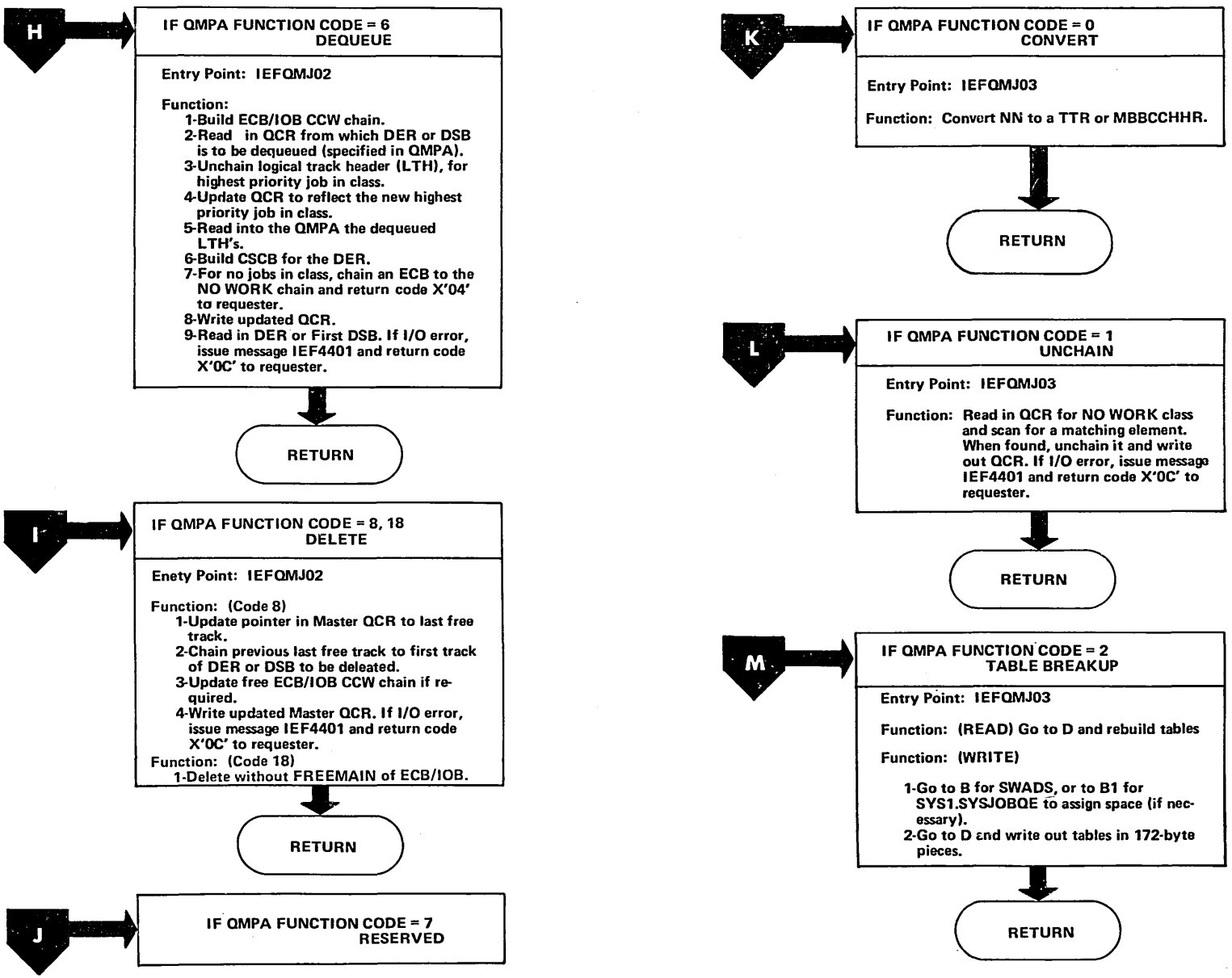


Figure 3-54. Queue Manager (Part 5 of 5)

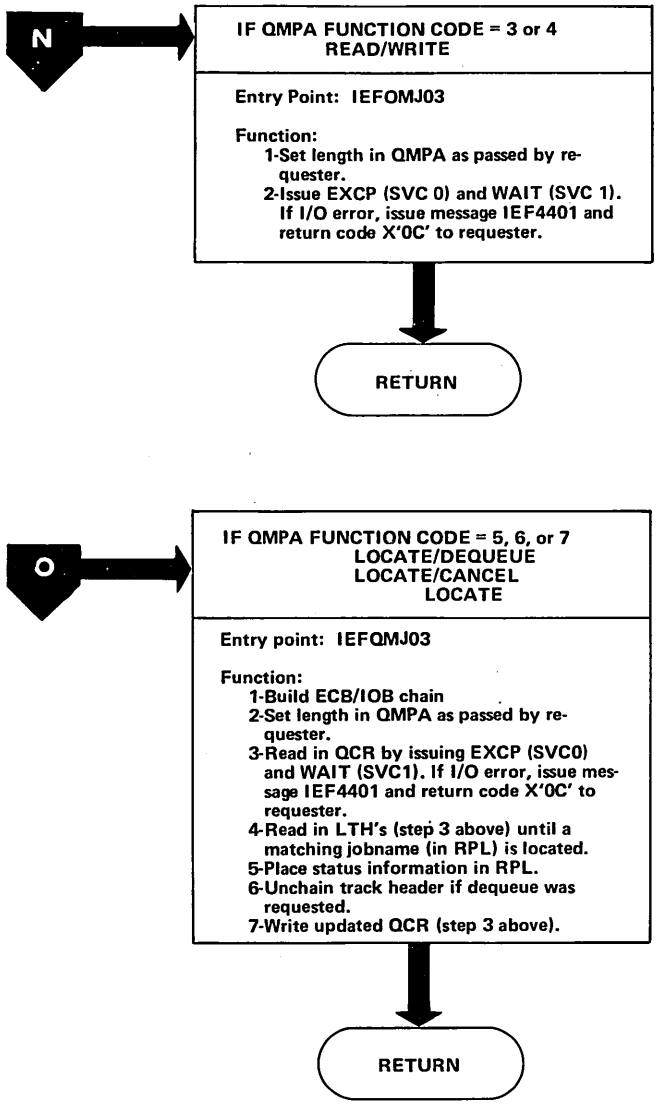


Figure 3-55. Master Scheduler Initialization Interconnection Module Diagram (Part 1 of 2)

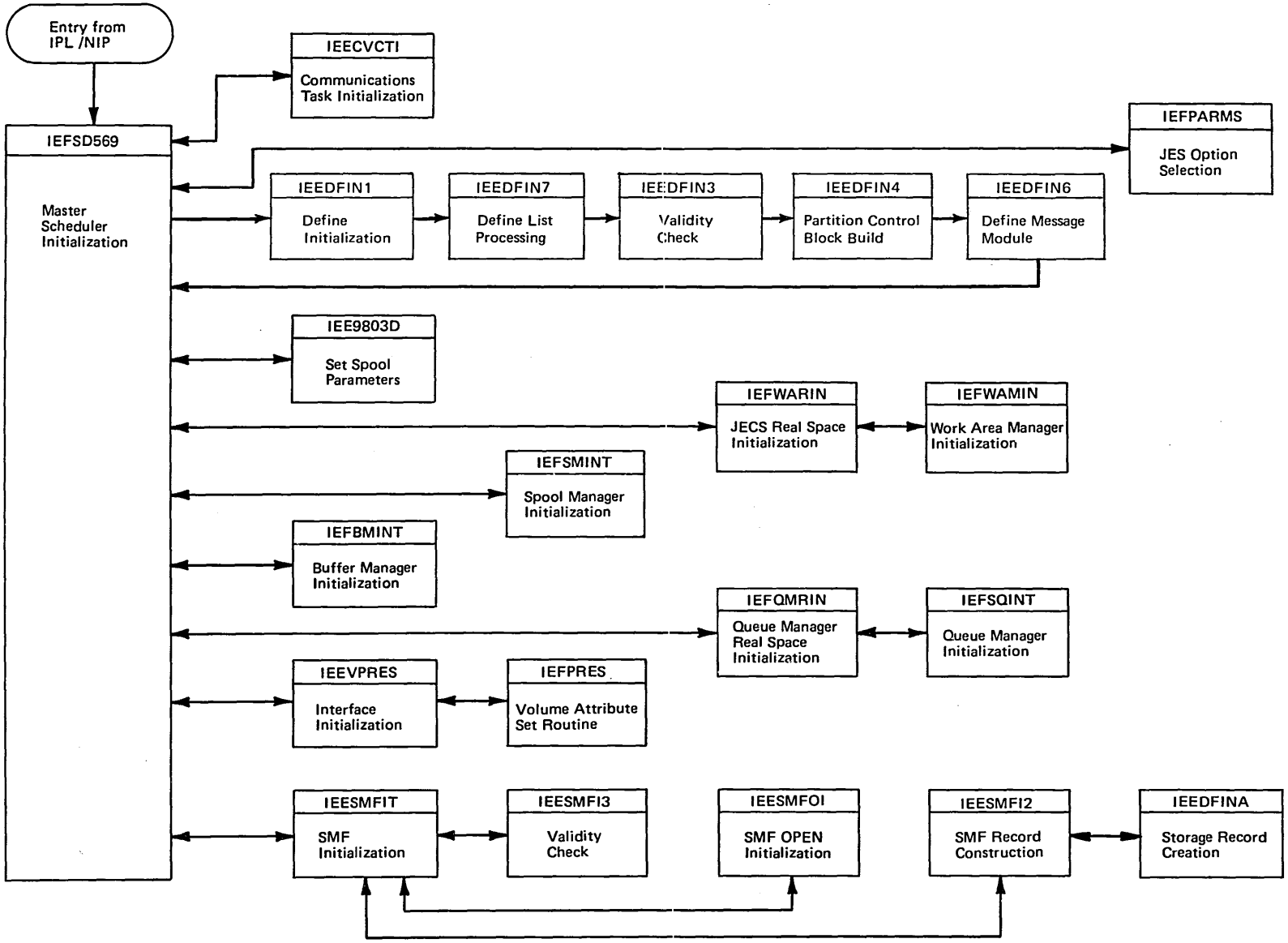


Figure 3-55. Master Scheduler Initialization Interconnection Module Diagram (Part 2 of 2).

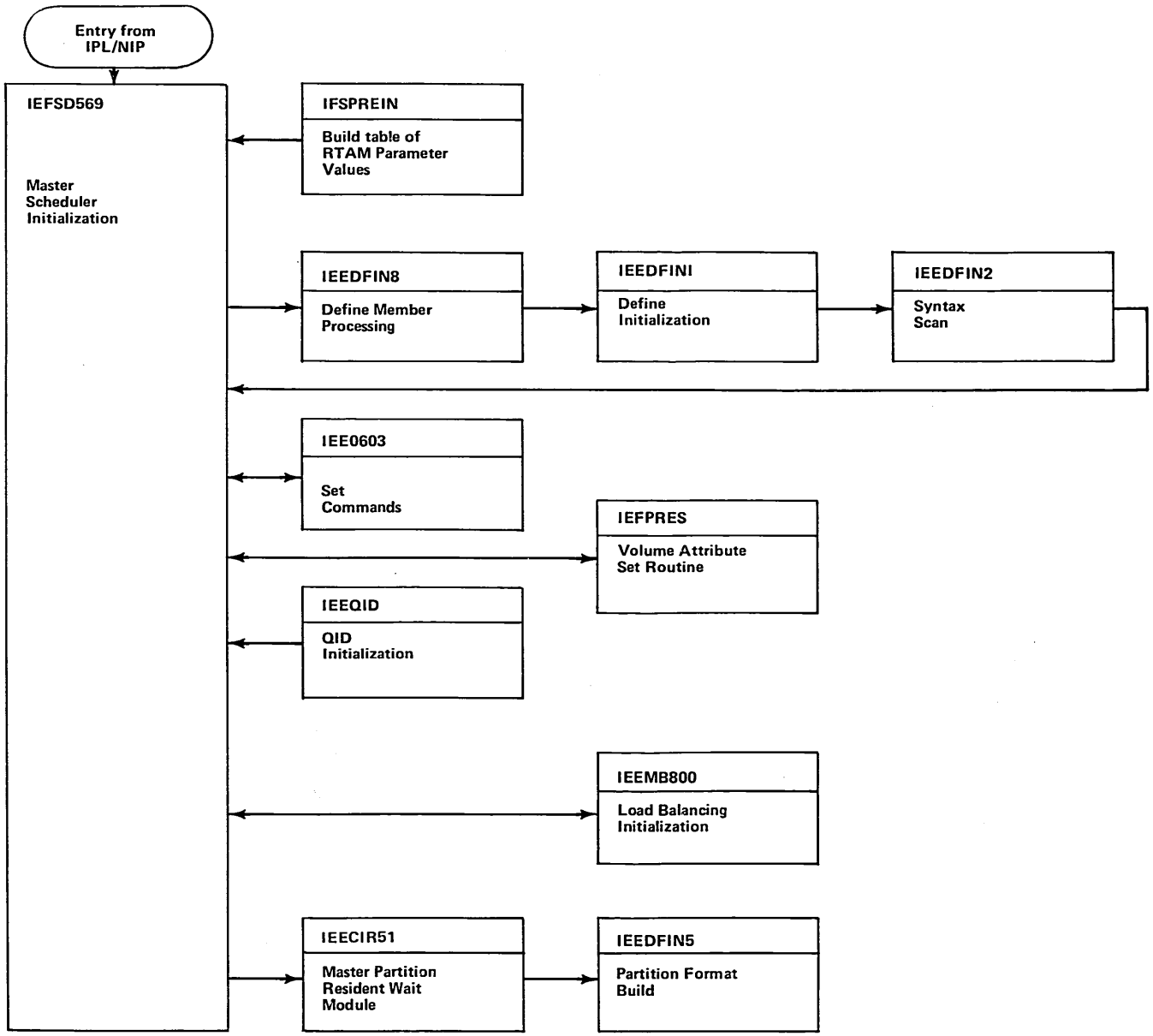
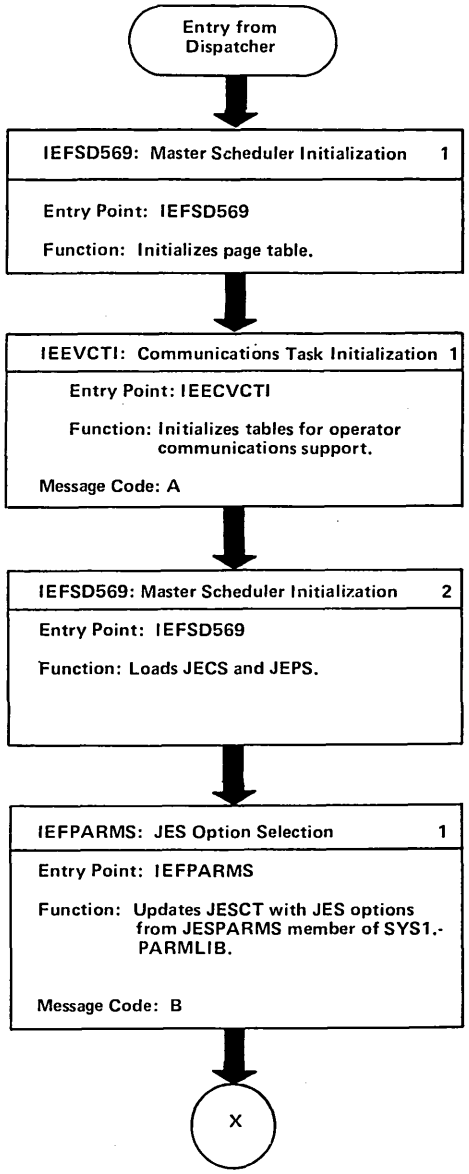
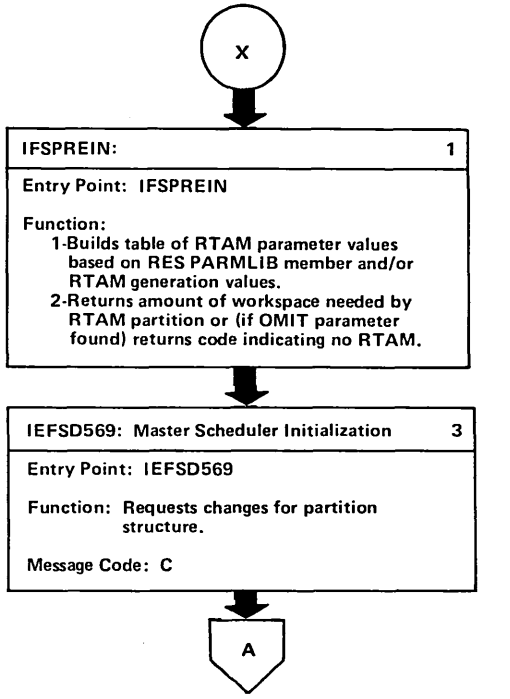


Figure 3-56. Master Scheduler Initialization (Part 1 of 11)





Messages Code	Number	Reason
A	IEE140I	Console status
B	IEF018I	Parameter incomplete
	IEF019I	Value too large
	IEF020I	Value not within limits
	IEF021I	Missing parenthesis
	IEF022I	Invalid number
	IEF023I	Invalid field
	IEF024I	Comma missing
	IEF025I	Invalid parameter
	IEF026I	Invalid volume ID
	IEF027I	Duplicate field
	IEF051I	JESPARMS error record
	IEF028I	No value specified
	IEF029I	Invalid first parameter
	IEF030I	I/O error
	IEF031I	SYSGEN values used
	IEF032I	SYS1.PARMLIB value used
C	IEE801D	Change partitions request
	IEE807A	Parameter error
	IEE015W	JES space exceeded--re-IPL

Figure 3-56. Master Scheduler Initialization (Part 2 of 11)

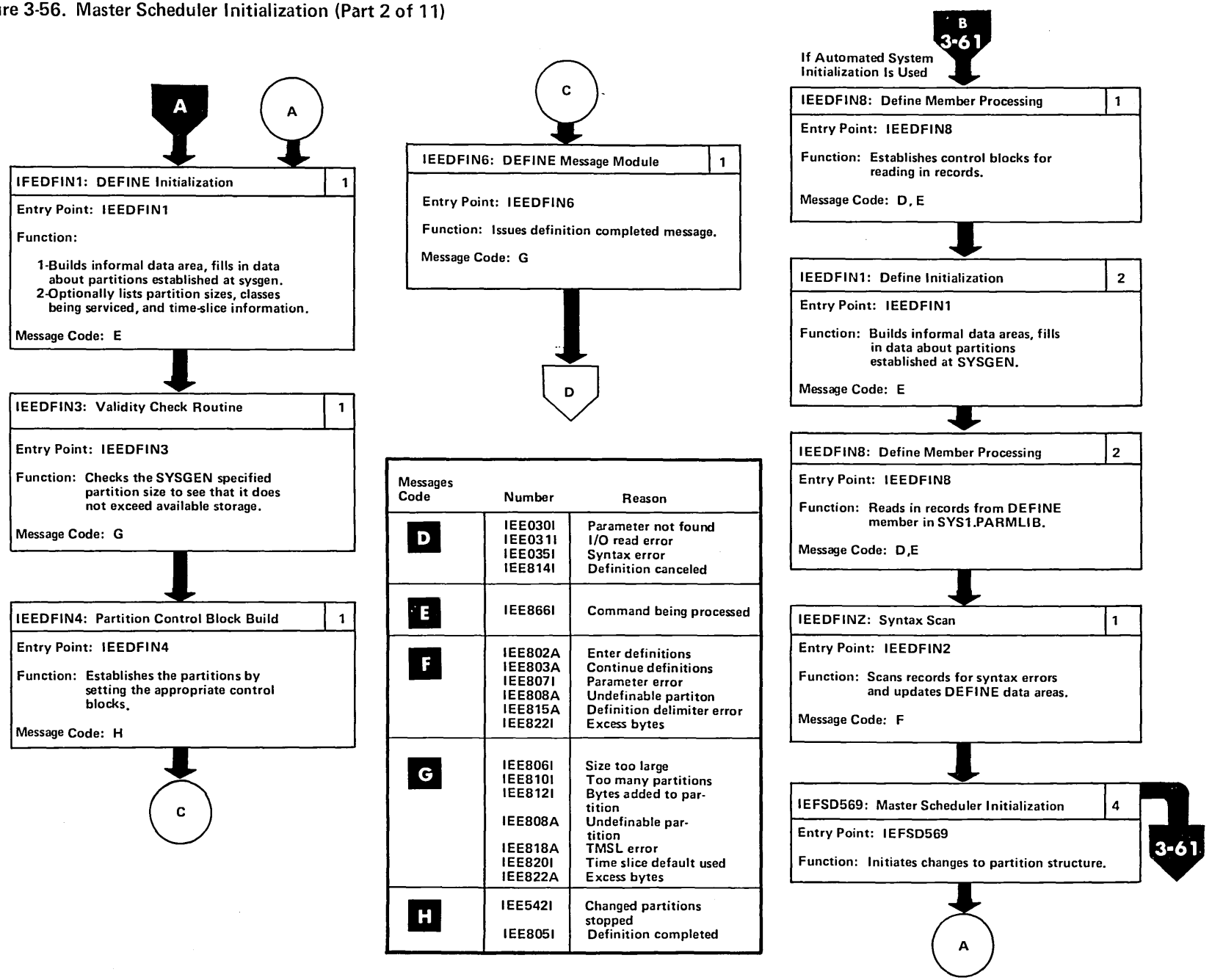
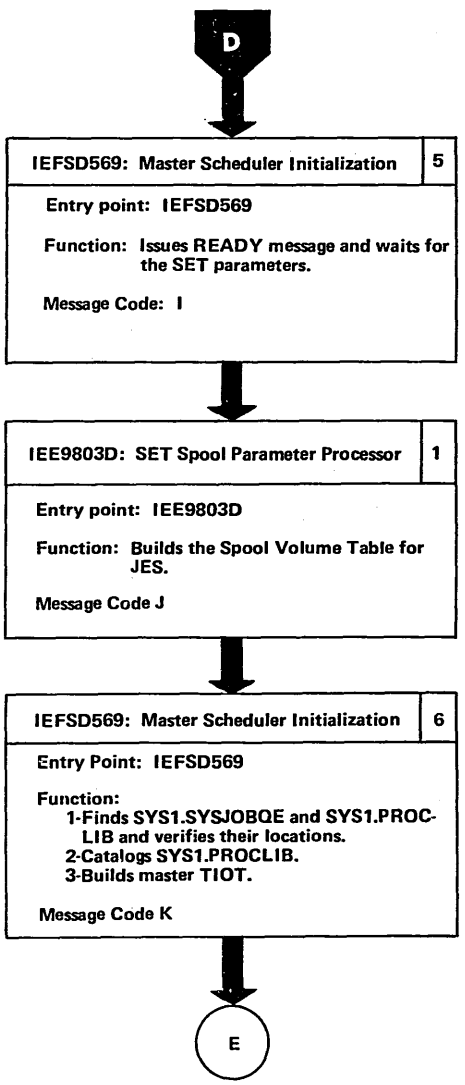
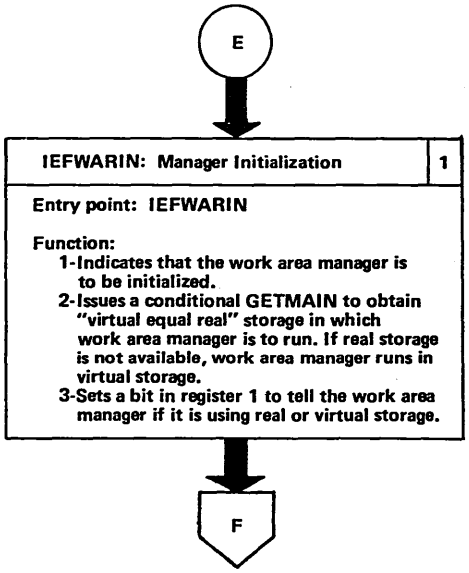


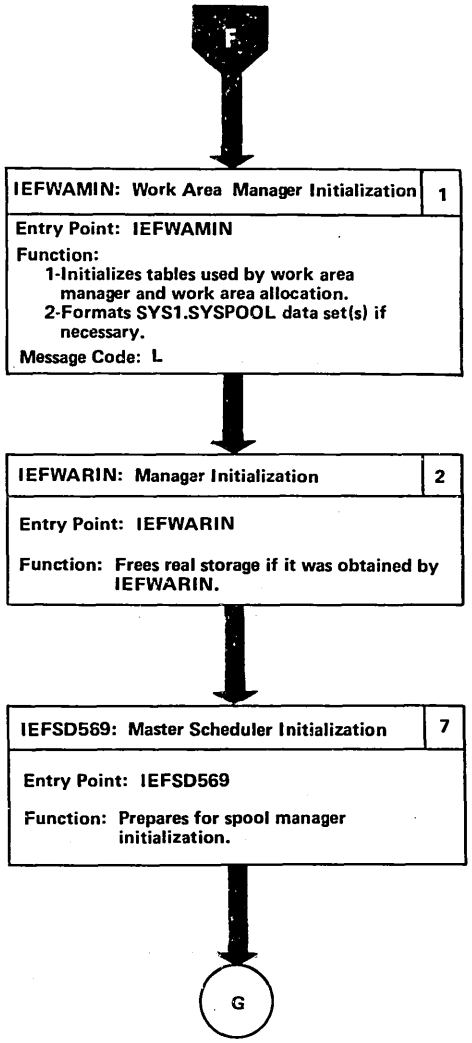
Figure 3-56. Master Scheduler Initialization (Part 3 of 11)

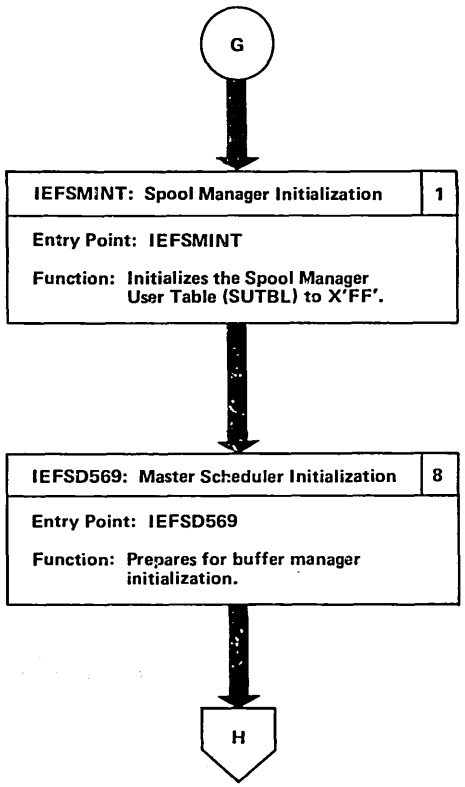




Messages Code	Number	Reason
I	IEE101A	Ready
	IEE114A	SET parameters
J	IEE545A	Unit changes
	IEE546A	Spool changes
	IEE547A	Delimiter error
	IEE548A	Parameter error
	IEE549A	Volume not mounted
	IEE550I	Spool table filled
	IEE552I	Changes canceled
	IEE553I	Spool changes completed
	IEE080I	Spool volume
	IEE551I	Deleted volumes
K	IFE343A	Volume not mounted
	IEE344A	Data set not found
	IEE313I	Invalid unit
	IEE139I	Initialization failed

Figure 3-56. Master Scheduler Initialization (Part 4 of 11)





Message Code	Number	Reason
L	IEF070I	Spool being formatted
	IEF055A	Volume xxx required
	IEF056A	Data set not found
	IEF057A	Buffer too large
	IEF058W	Data area too small
	IEF059A	Insufficient space or volume
	IEF060A	Request not valid
	IEF061A	Request not valid
	IEF062W	Permanent I/O error
	IEF063W	I/O error
	IEF064A	Volume requires SCCV
	IEF065A	SYS1.SYSJOBQE required on volume xxx
	IEF066A	Invalid buffer size
	IEF067A	Volume unformatted
	IEF068A	Formatting required
	IEF069W	Tables unable to be fixed

Figure 3-56. Master Scheduler Initialization (Part 5 of 11)

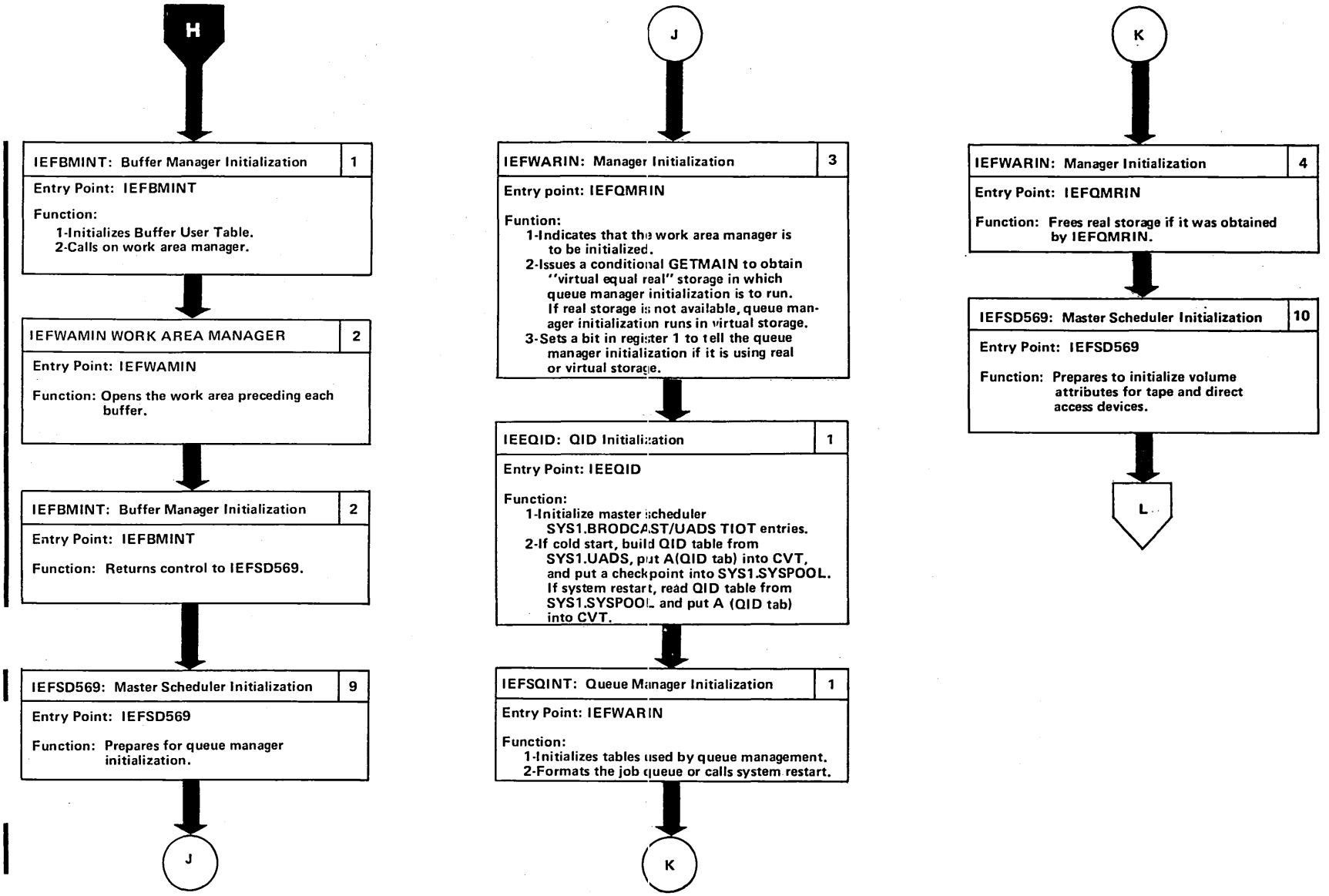
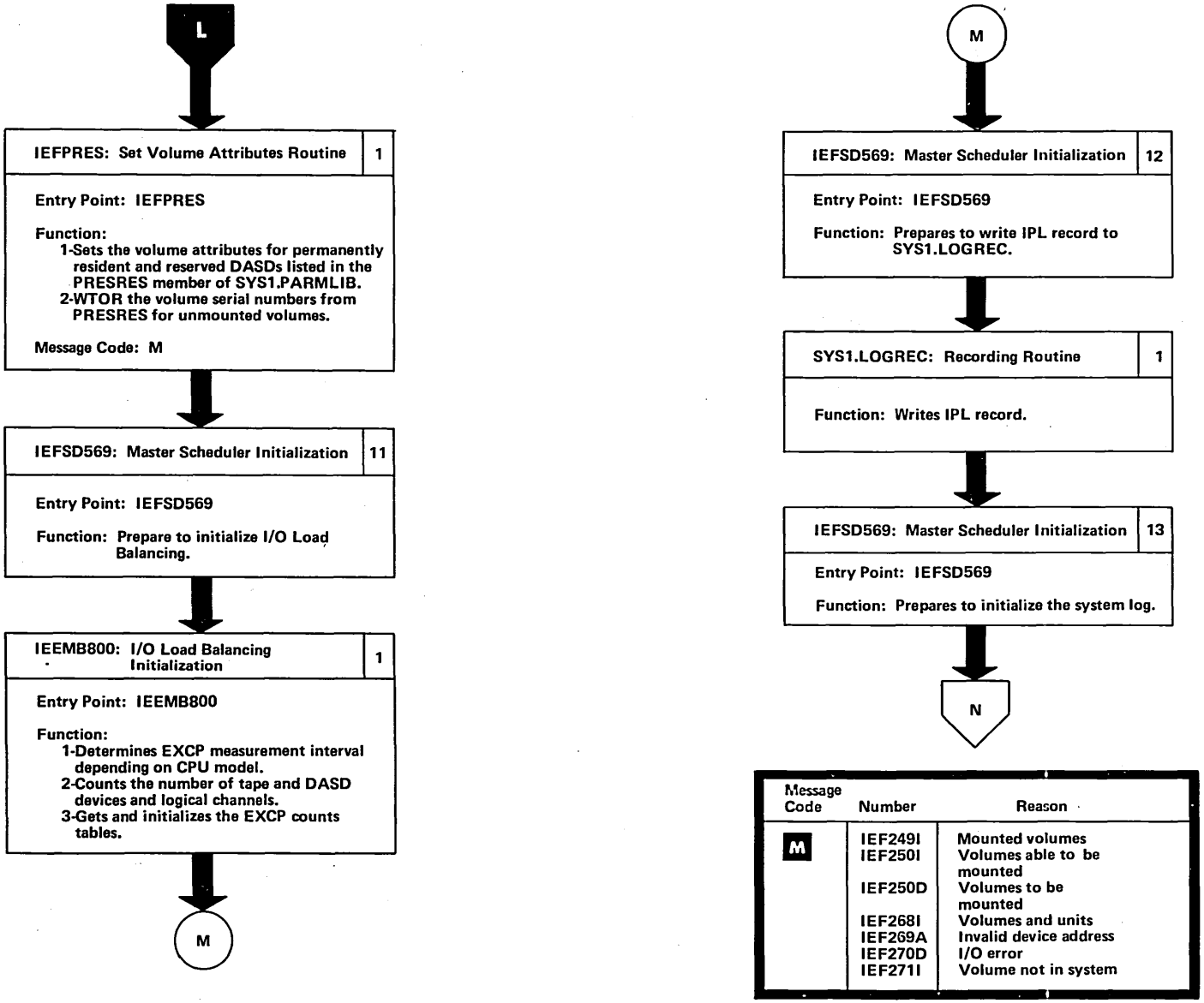
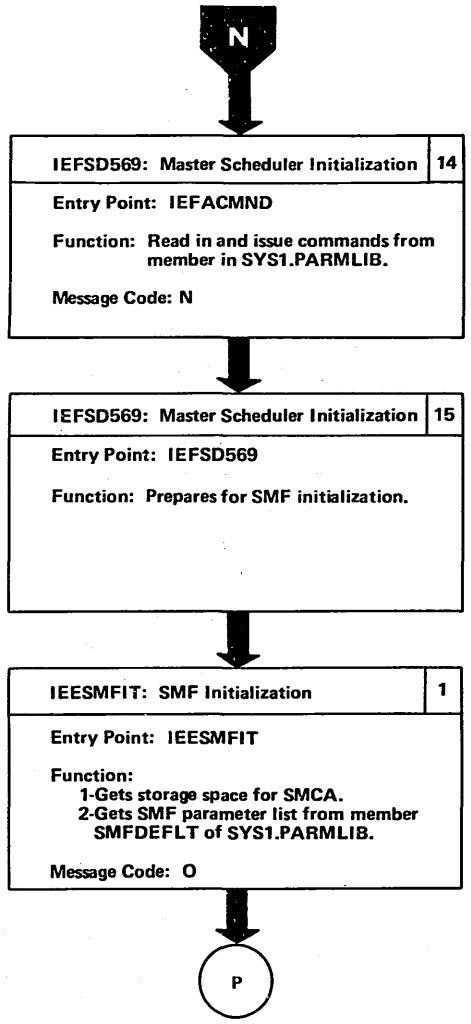


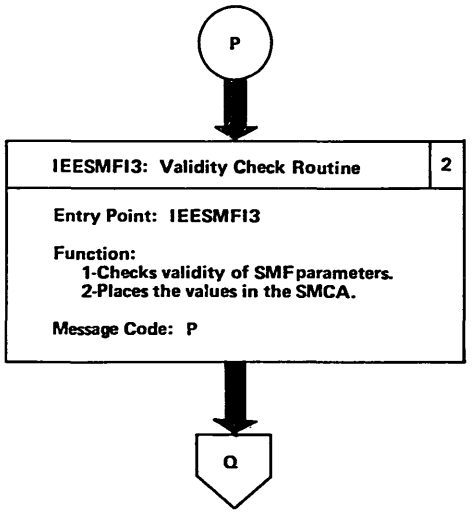
Figure 3-56. Master Scheduler Initialization (Part 6 of 11)



Message Code	Number	Reason
M	IEF249I	Mounted volumes
	IEF250I	Volumes able to be mounted
	IEF250D	Volumes to be mounted
	IEF268I	Volumes and units
	IEF269A	Invalid device address
	IEF270D	I/O error
	IEF271I	Volume not in system

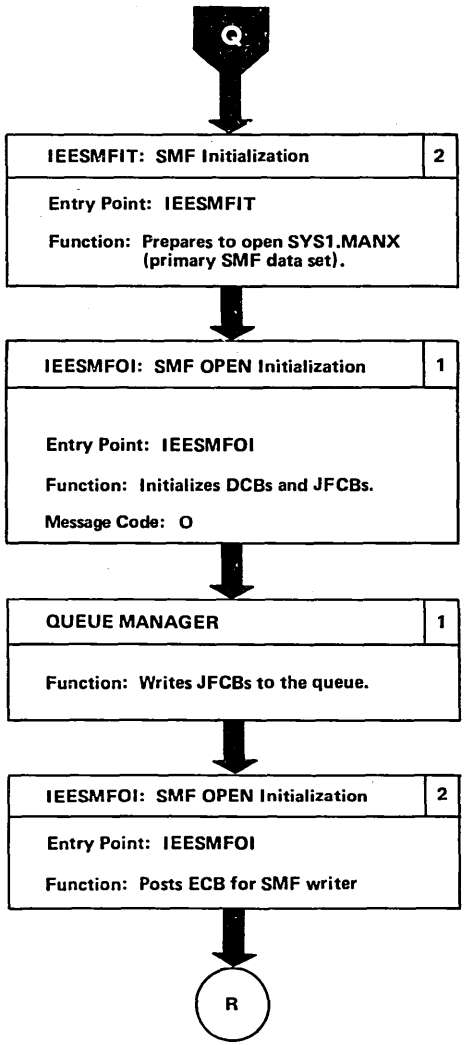
Figure 3-56. Master Scheduler Initialization (Part 7 of 11)

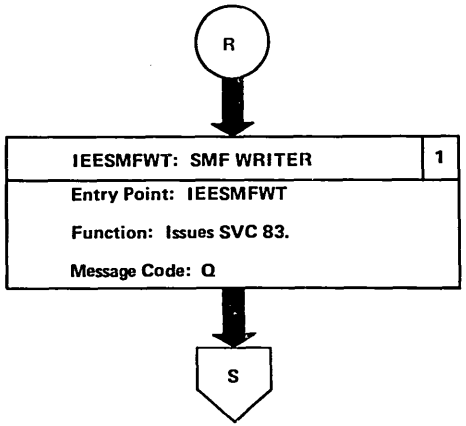




Messages Code	Number	Reason
N	IEE052I	Command Issued
O	IEE351I	No SMF recording
P	IEE352A	SMFDEFLT not found
	IEE353A	I/O error
	IEE354I	SMF parmeters
	IEE355I	Parameter errors
	IEE356A	SMF values
	IEE357A	Change SMF values

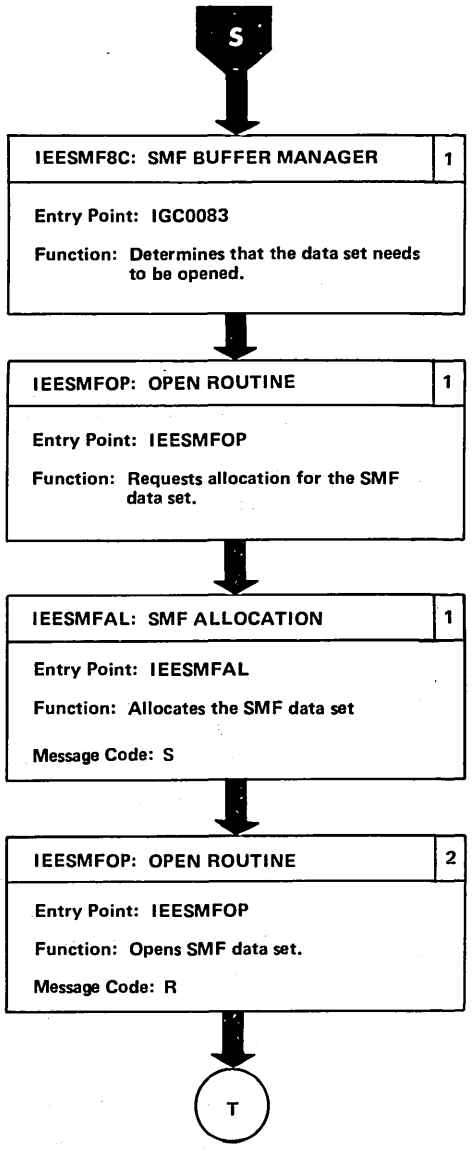
Figure 3-56. Master Scheduler Initialization (Part 8 of 11)

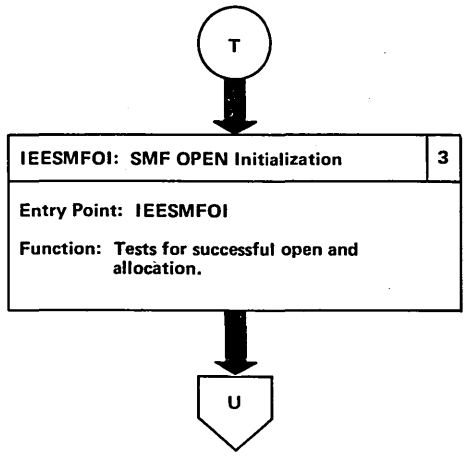




Message Code	Number	Reason
Q	IEE361I	SMF data lost

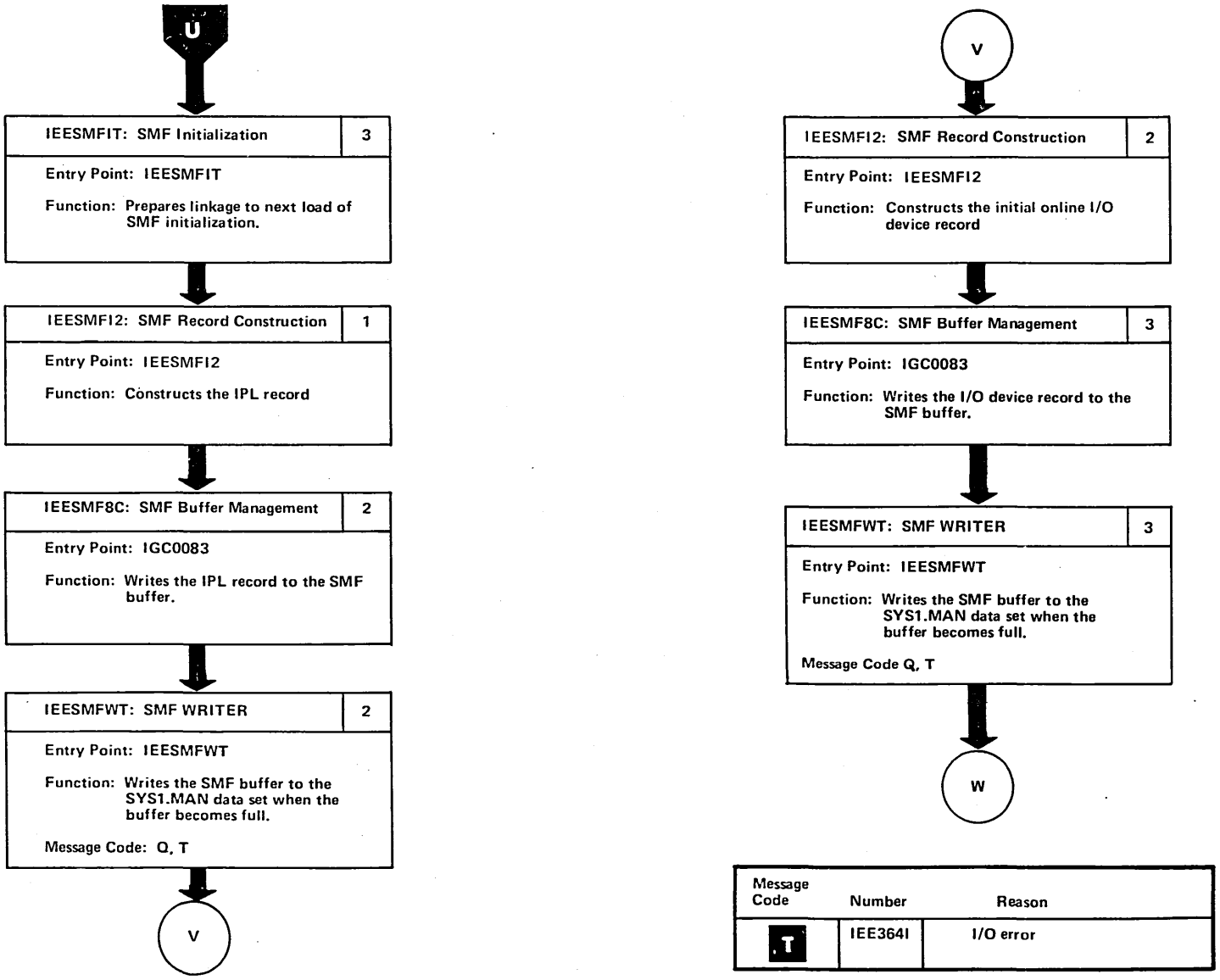
Figure 3-56. Master Scheduler Initialization (Part 9 of 11)





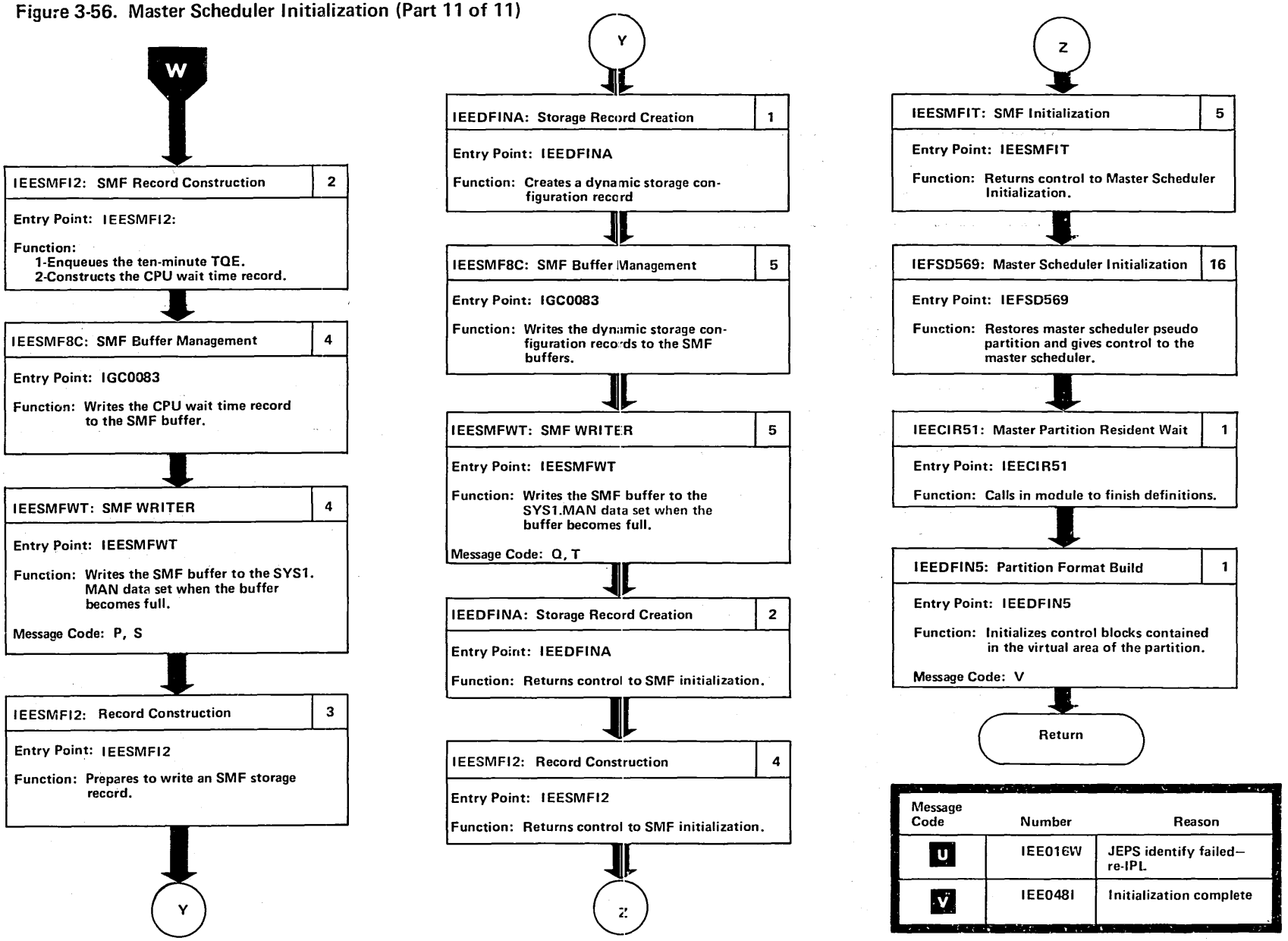
Messages Code	Number	Reason
R	IEE358I	SYS1.MANX not found
	IEE360I	SYS1.MANX unit and time
	IEE362A	Dump SYS1.MANX
S	IEE363I	Device status
	IEE350I	SYS1.MANX not defined

Figure 3-56. Master Scheduler Initialization (Part 10 of 11)



Message Code	Number	Reason
T	IEE364I	I/O error

Figure 3-56. Master Scheduler Initialization (Part 11 of 11)



Message Code	Number	Reason
U	IEE016W	JEPS identify failed—re-IPL
V	IEE048I	Initialization complete

Figure 3-57. Command Processing Interconnection Module Diagram (Part 1 of 6)

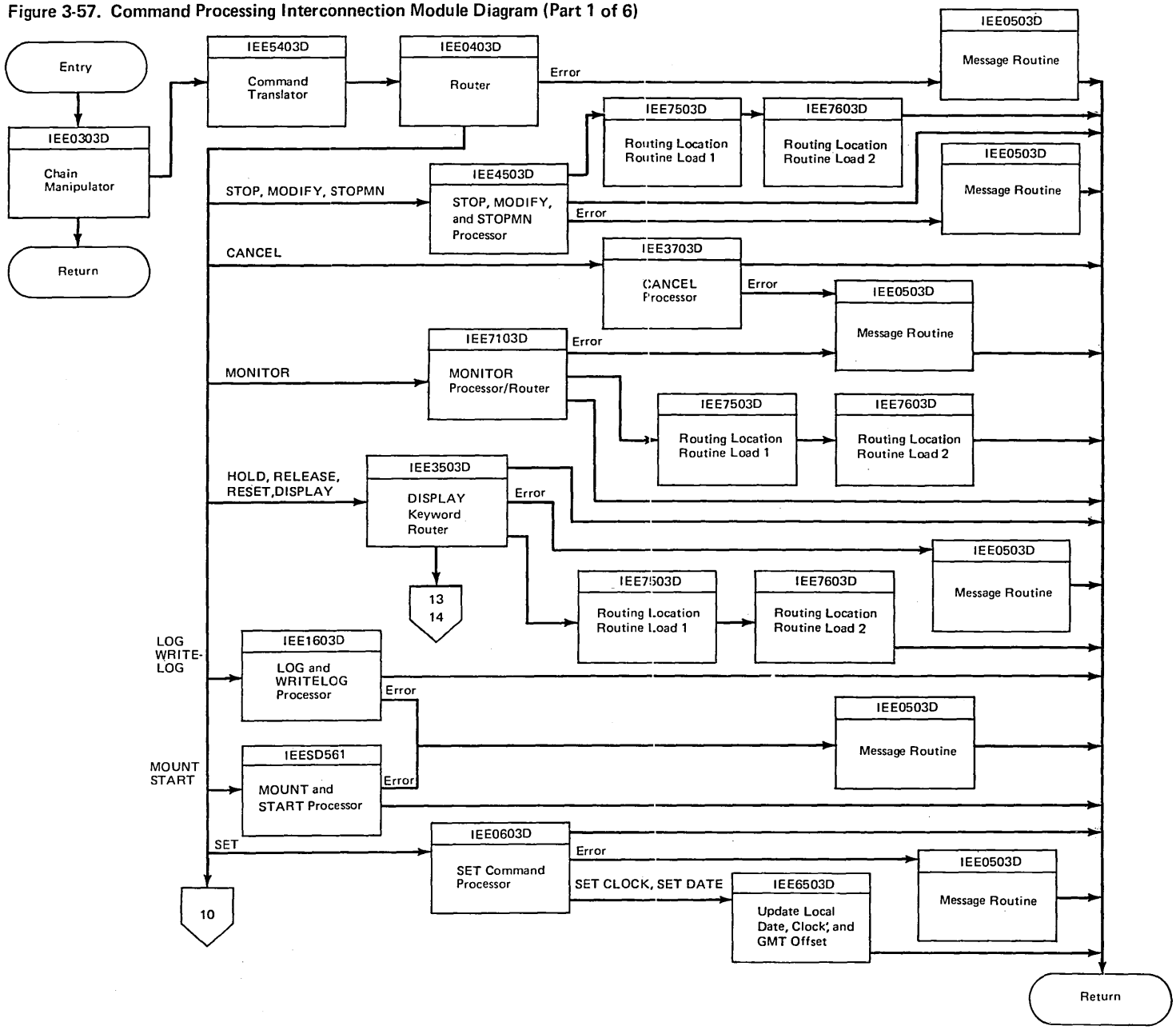


Figure 3-57. Command Processing Interconnection Module Diagram (Part 2 of 6)

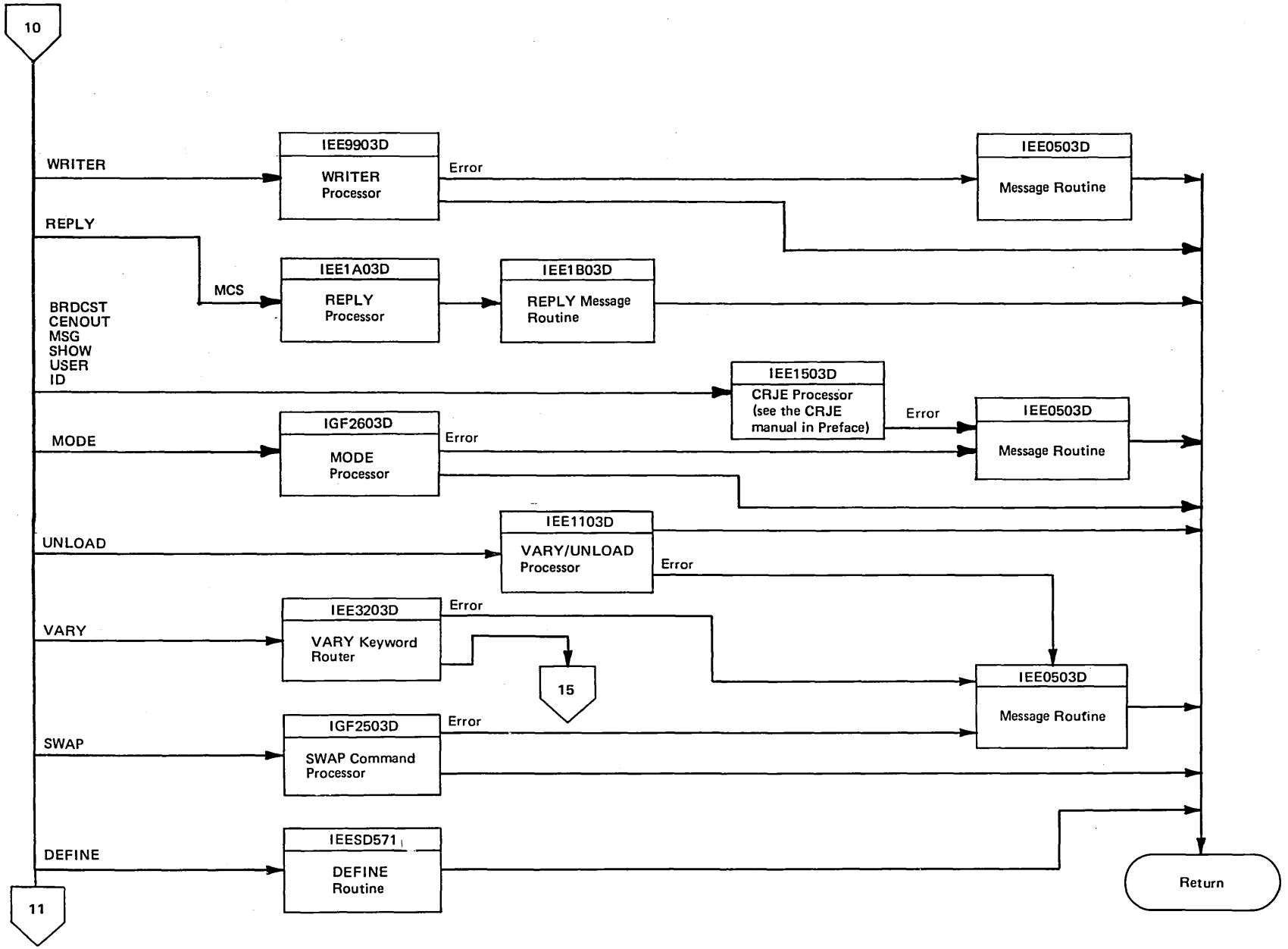


Figure 3-57. Command Processing Interconnection Module Diagram (Part 3 of 6)

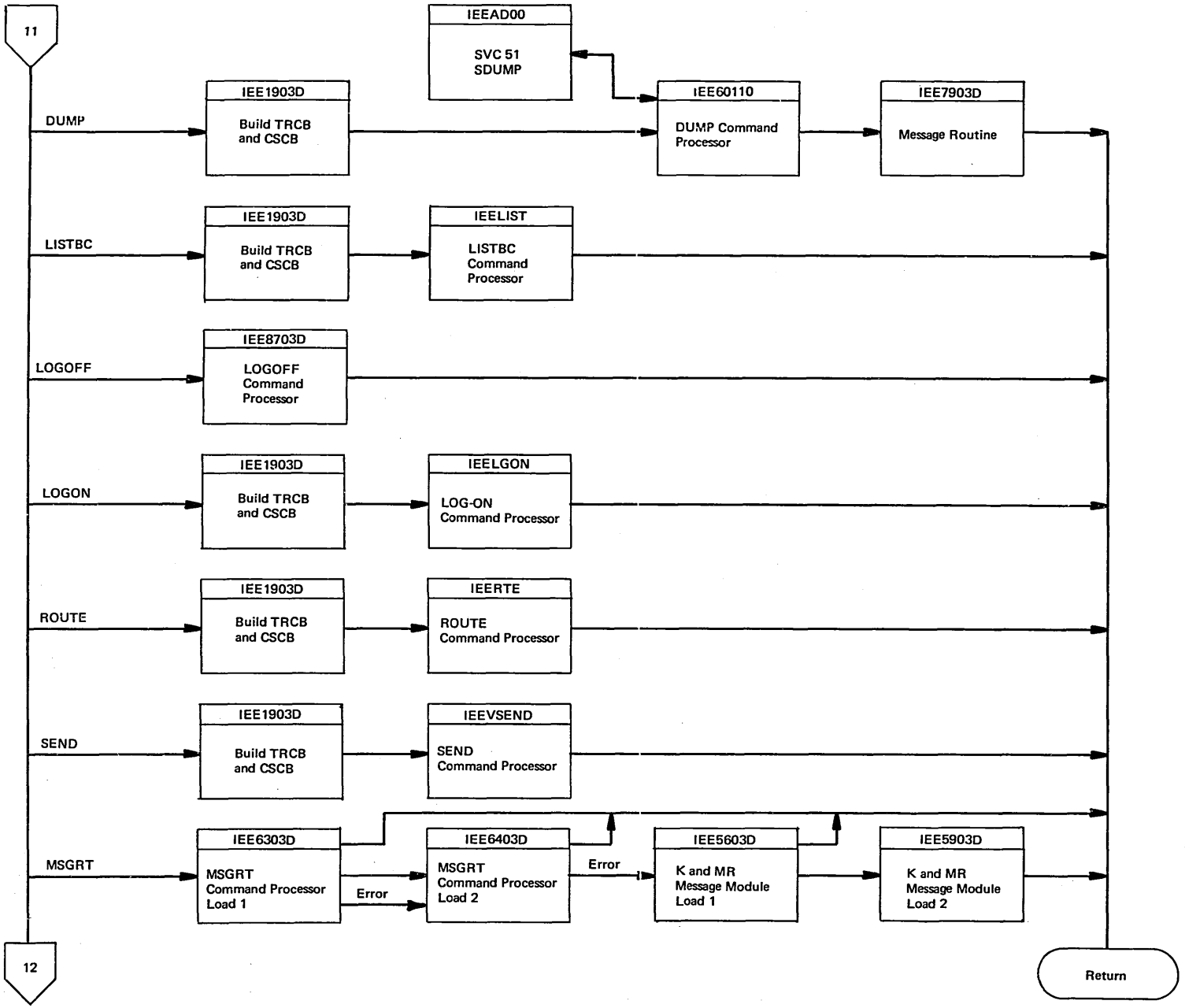


Figure 3-57. Command Processing Interconnection Module Diagram (Part 4 of 6)

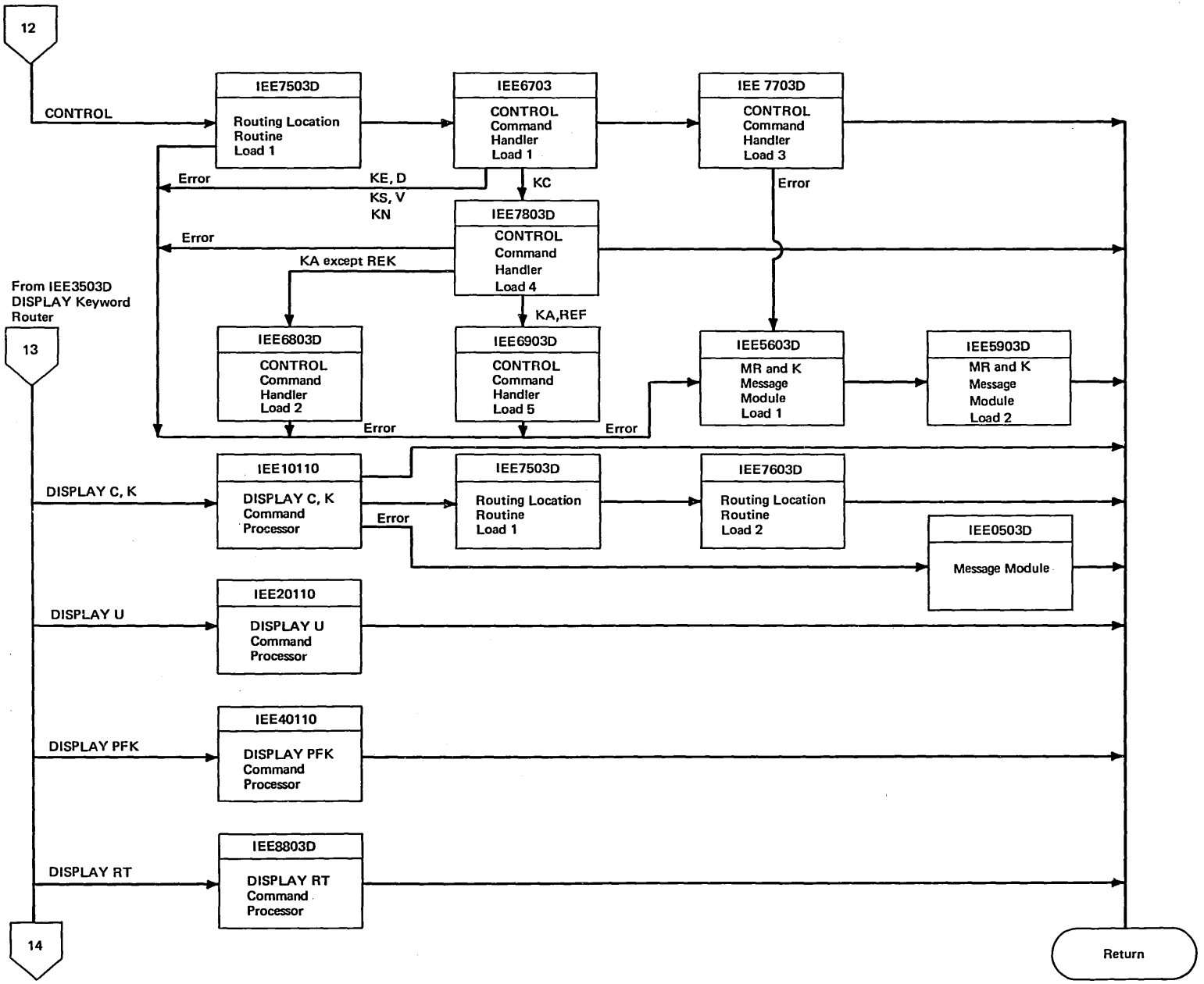


Figure 3-57. Command Processing Interconnection Module Diagram (Part 5 of 6)

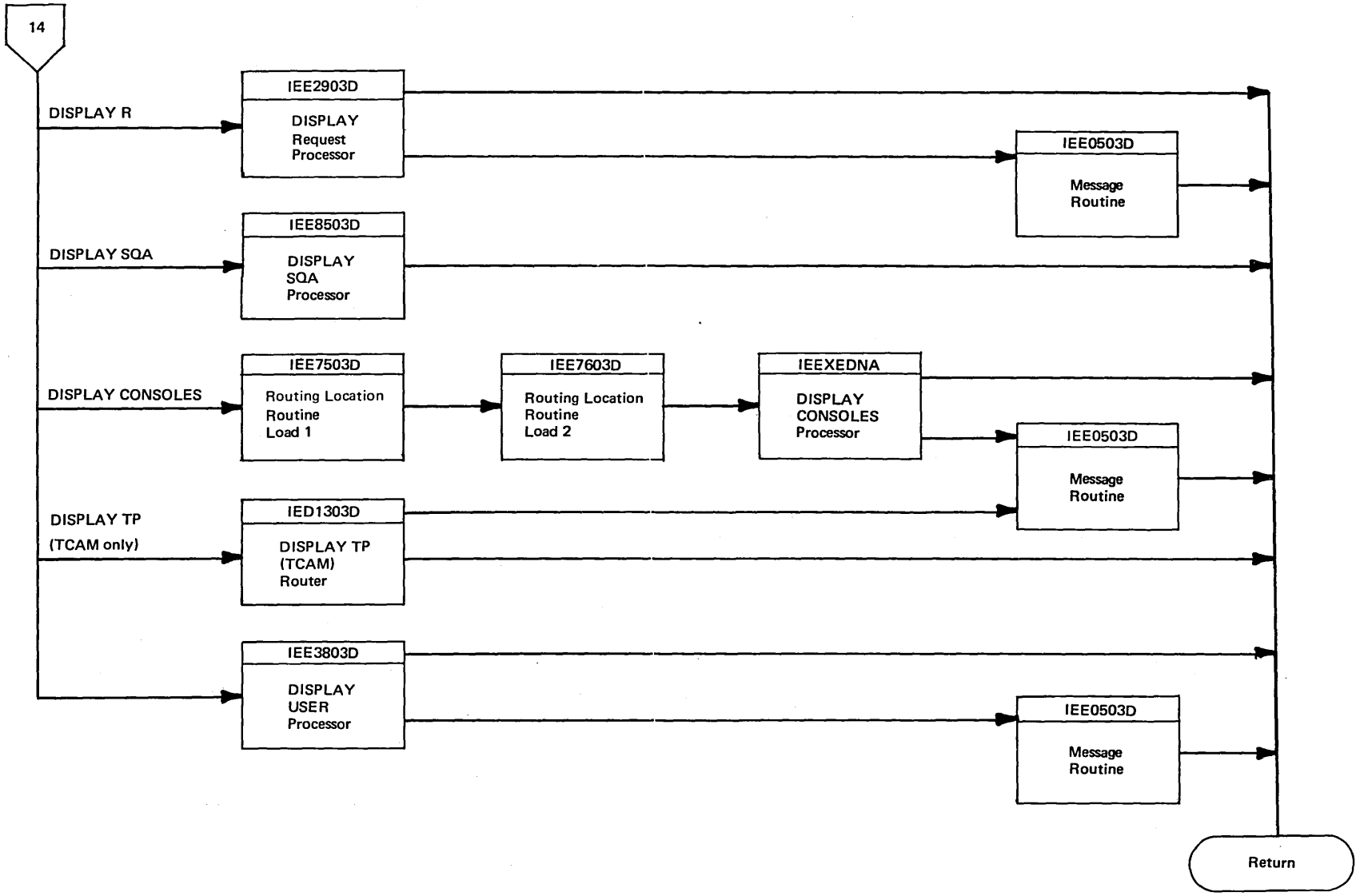


Figure 3-57. Command Processing Interconnection Module Diagram (Part 6 of 6)

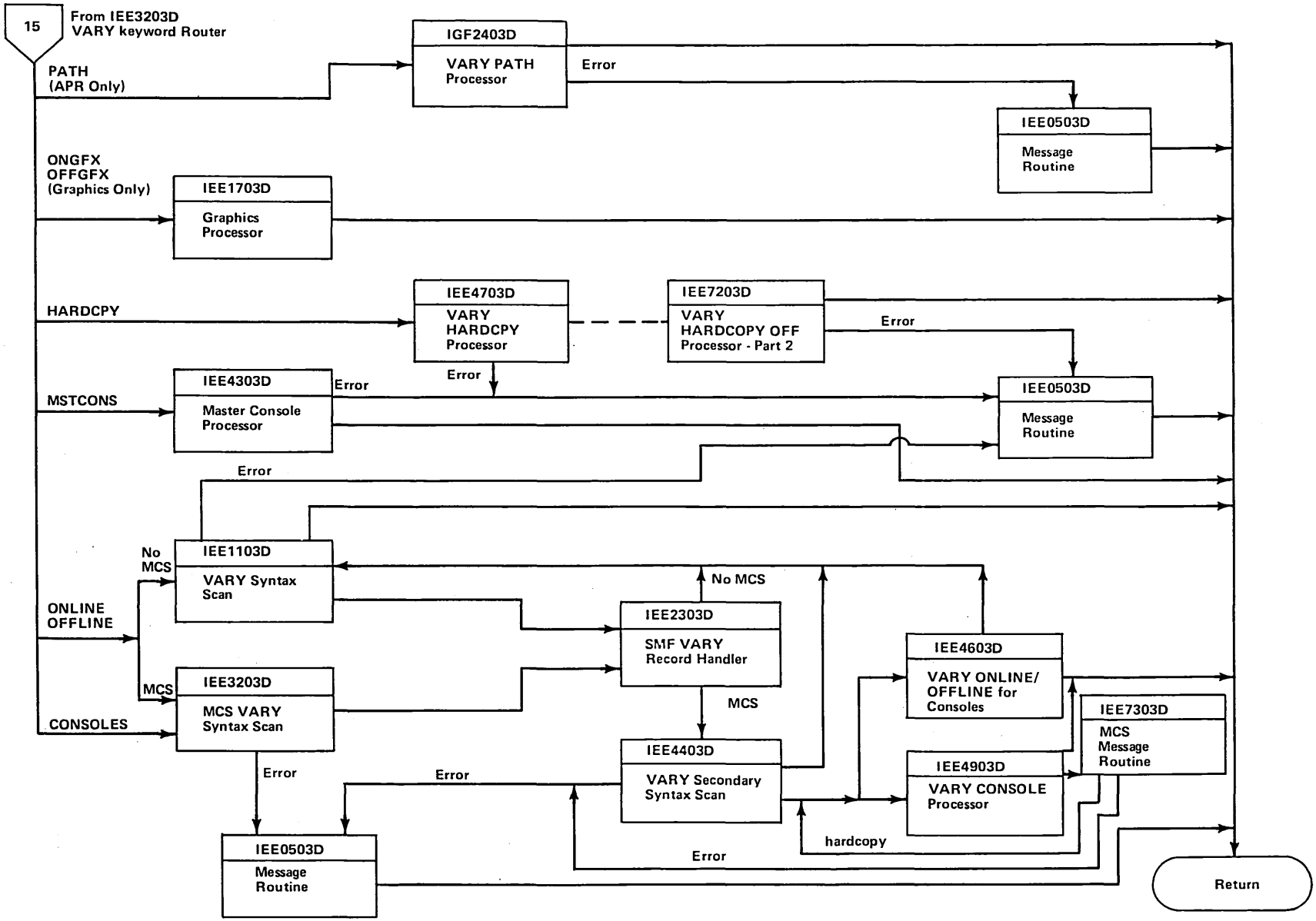
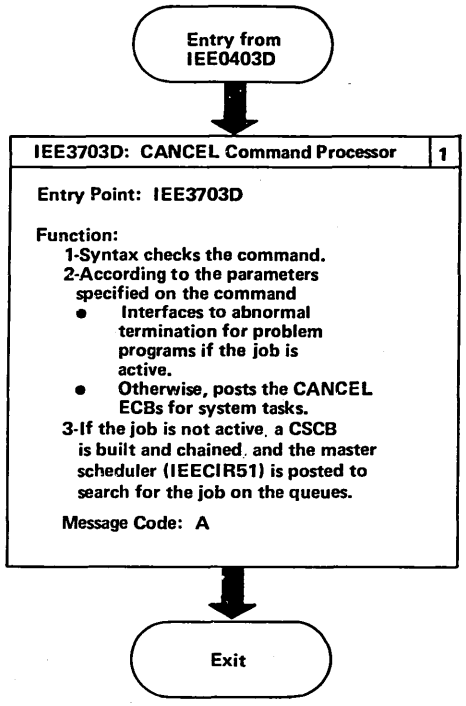
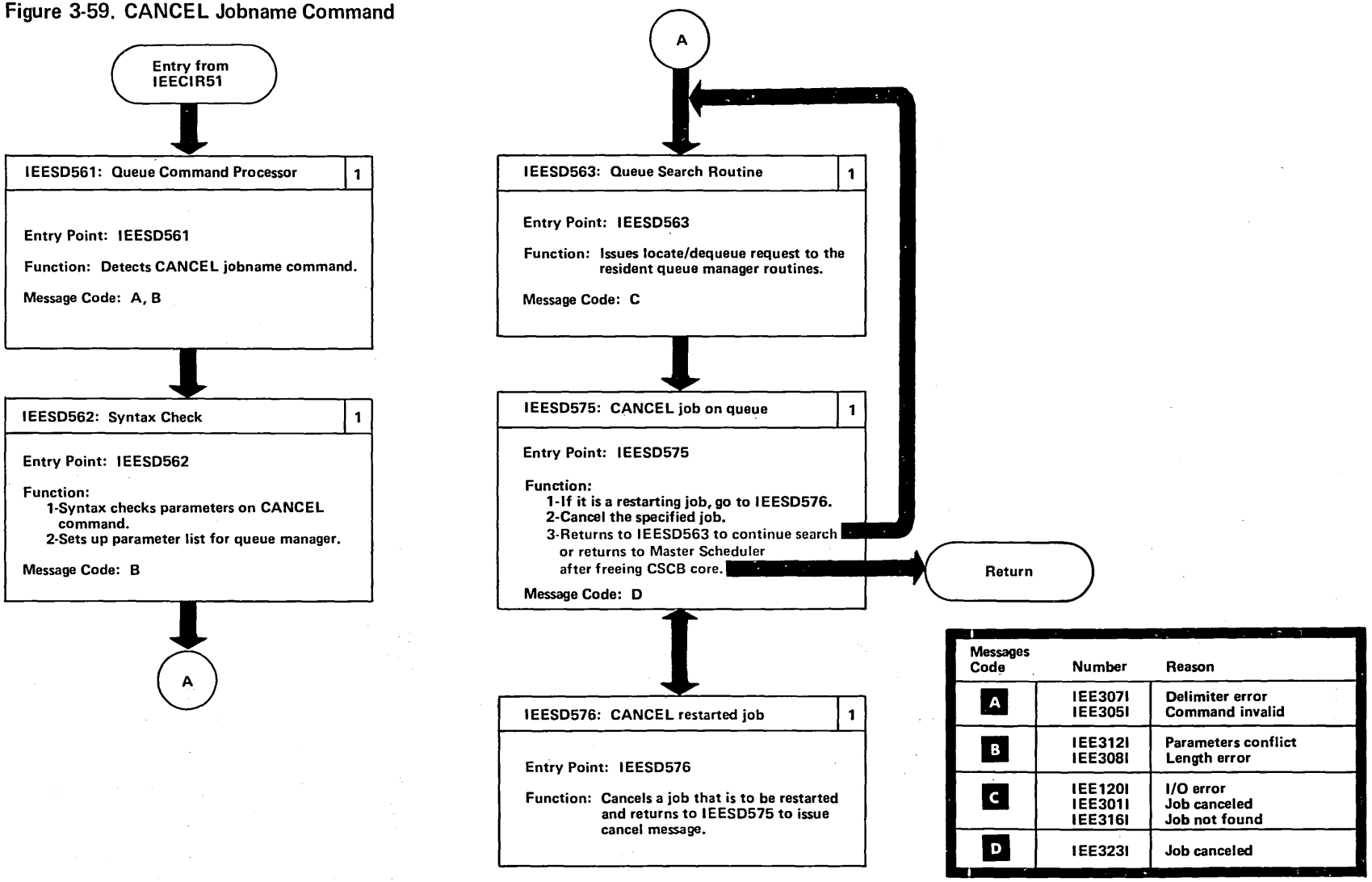


Figure 3-58. CANCEL Command



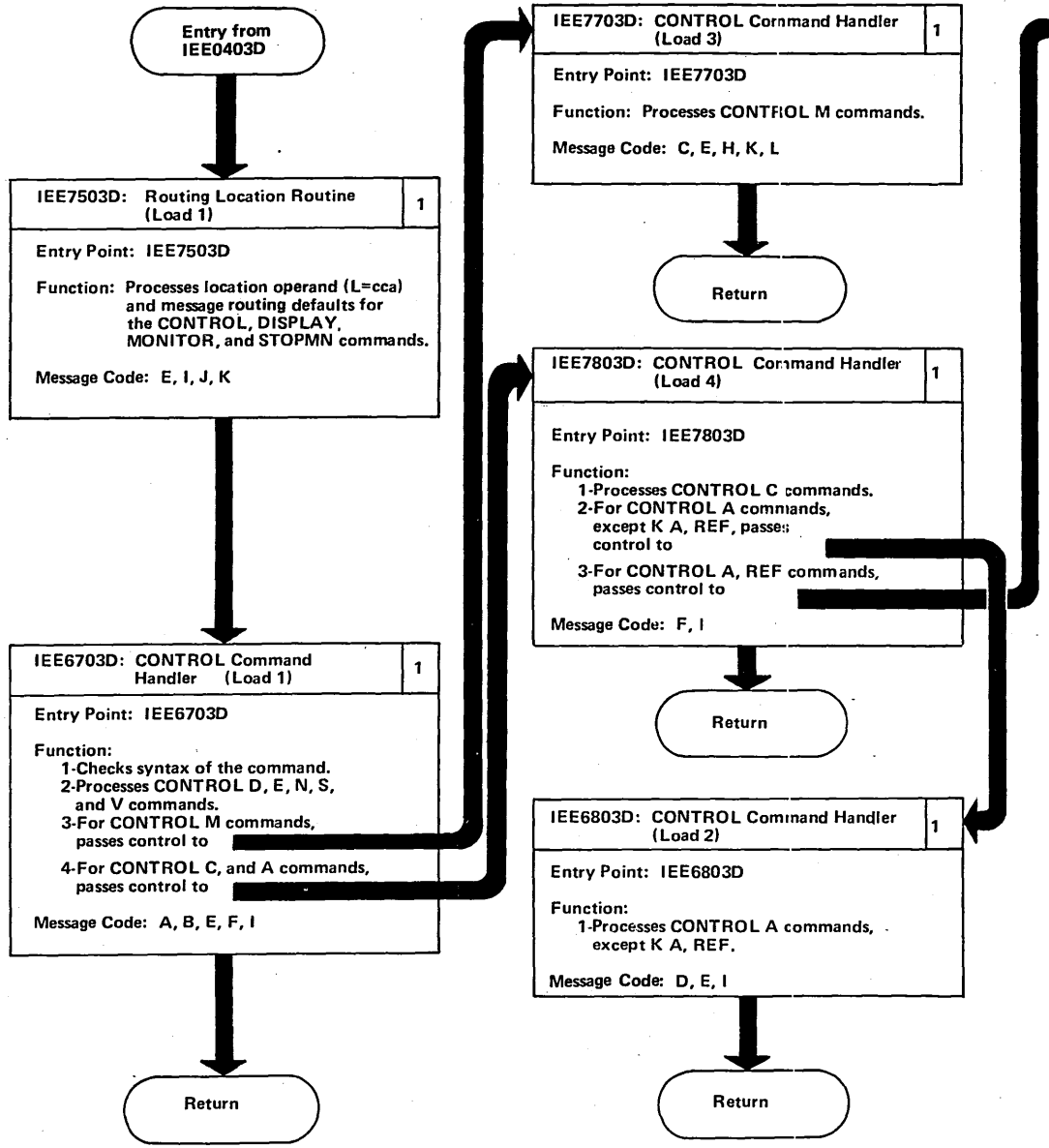
Message Code	Number	Reason
A	IEE342I	Rejected task busy
	IEE306I	Command invalid
	IEE301I	Command accepted
	IEE317I	Job selected
	IEE008I	Length error
	IEE311I	Parameter missing

Figure 3-59. CANCEL Jobname Command



Messages Code	Number	Reason
A	IEE307I IEE305I	Delimiter error Command invalid
B	IEE312I IEE308I	Parameters conflict Length error
C	IEE120I IEE301I IEE316I	I/O error Job canceled Job not found
D	IEE323I	Job canceled

Figure 3-60. CONTROL Command



Messages Code	Number	Reason
A	IEE151I	Delete request inconsistent - {use STOPMN to delete / no display on screen / no display in area}
B	IEE158I	K request inconsistent - {no display on screen / no display in area / last frame displayed / frame not complete / display is not dynamic}
C	IEE150I IEE156I	Change options if desired Invalid operand - xxxx
D	IEE924I IEE925I IEE928I IEE931I	Invalid area definition - display in or above area Operand for CRT only Display area IDs for console cc are xxx Insufficient storage for command
E	IEE926I	L=cca operand invalid - output only.
F	IEE927I	Request ambiguous - specify L operand.
G	IEE929I	K A, {NONE / length, . . .L=cc}
H	IEE931I	Insufficient storage for command
I	IEE156I	Invalid operand - xxxx
J	IEE305I	Command invalid
K	IEE345I	Authority invalid
L	IEE921I	Rejected {no timer / device not supported}

Figure 3-61. DEFINE Command (Part 1 of 2)

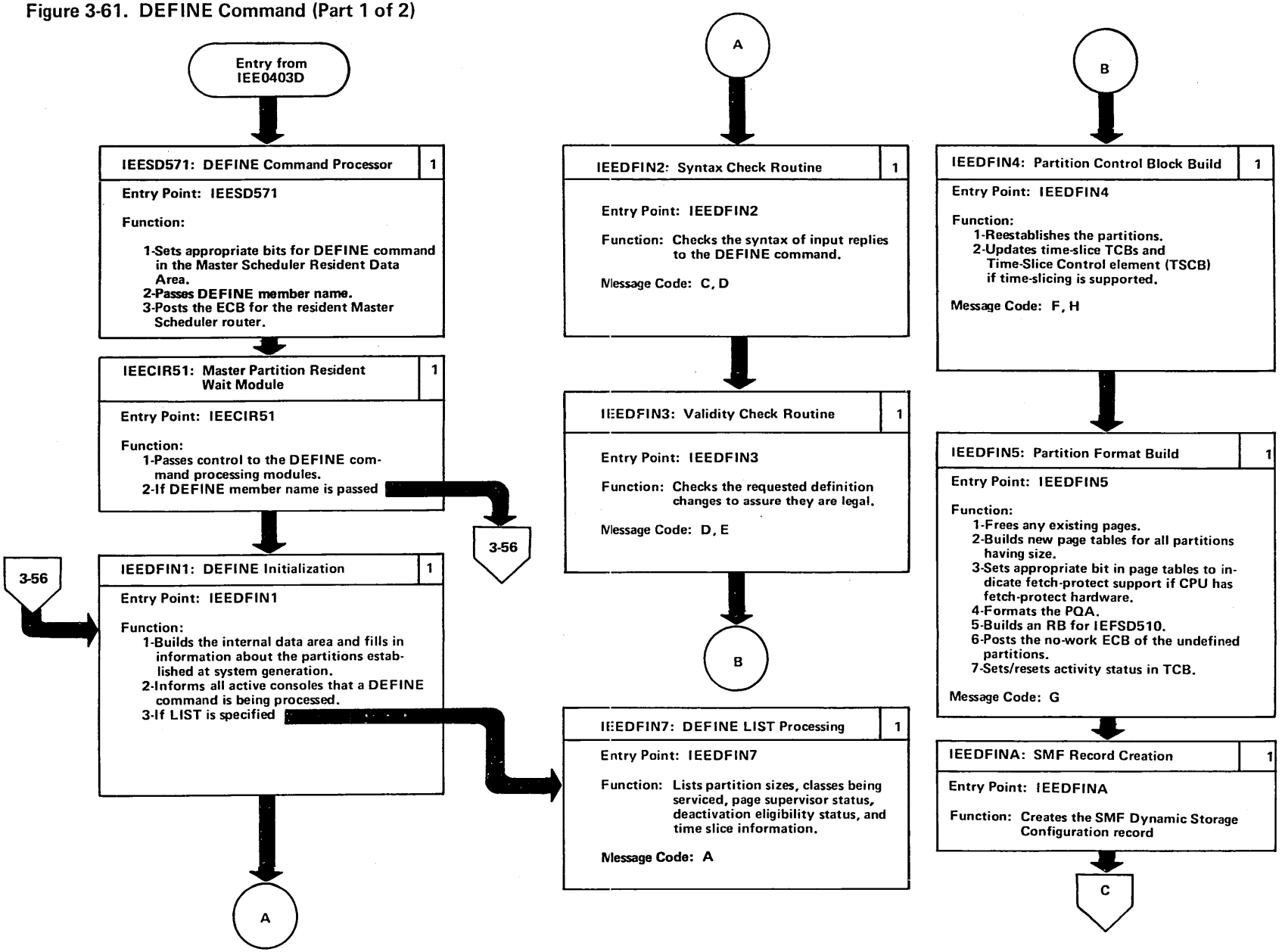
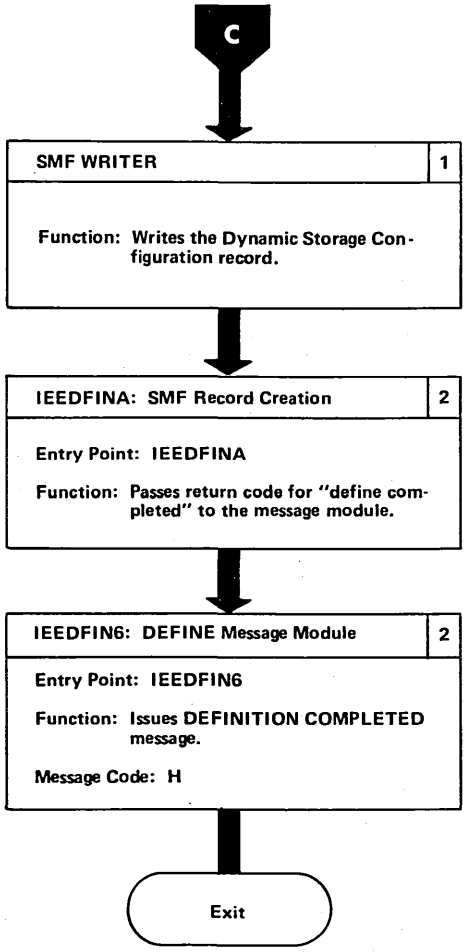


Figure 3-61. DEFINE Command (Part 2 of 2)



Message Code	Number	Reason
A	IEE866I	Command being processed
	IEE819I	Time slicing not supported
	IEE804I	Partition description
	IEE816I	Classes
	IEE817I	Time slicing data
	IEE543I	X Bytes of free space
	IEE544I	Total size too large
	IEE802A	Enter definition
	IEE803A	Continue definition
C	IEE815A	Delimiter error
	IEE823A	Time slicing or hierarchy not supported
	IEE803A	Continue definition
	IEE814I	DEFINE canceled
D	IEE808A	Undefinable partition
	IEE818A	TMSL specification error
	IEE807I	Parameter error
E	IEE809I	Total size too large
	IEE810I	Too many partitions
	IEE812I	Bytes added to partition
	IEE820I	Time slice default used
	IEE822A	No partition for excess bytes
	IEE806I	Total size too large
	IEE811I	Changed partition not adjacent
F	IEE542I	Changed partition stopped
G	IEE048I	Initiation complete
	IEE079I	Command failed
H	IEE805I	Definition completed

Figure 3-62. DISPLAY and DISPLAY ACTIVE Commands

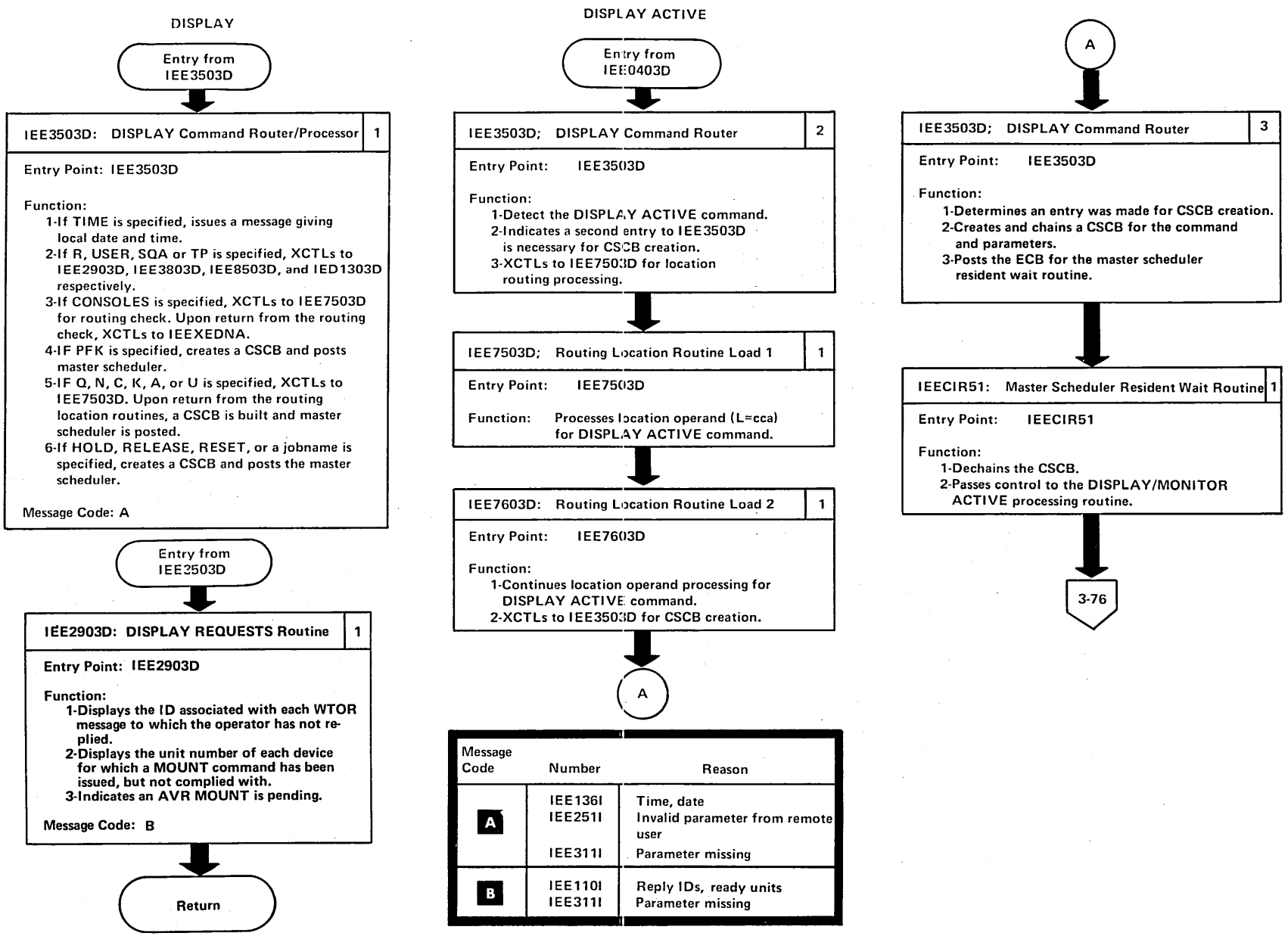


Figure 3-63. DISPLAY CONSOLES and DISPLAY RT Commands (Part 1 of 2)

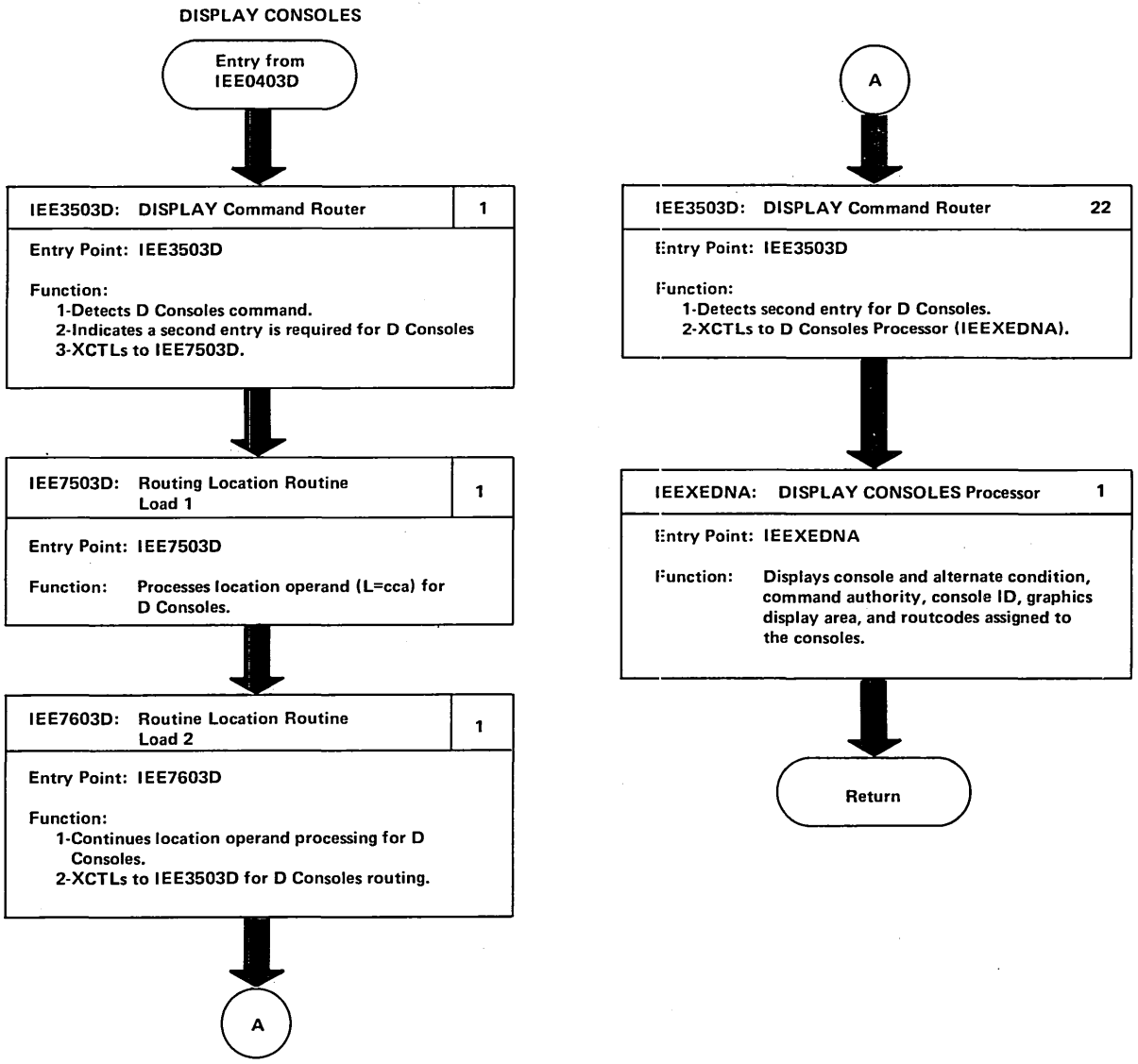
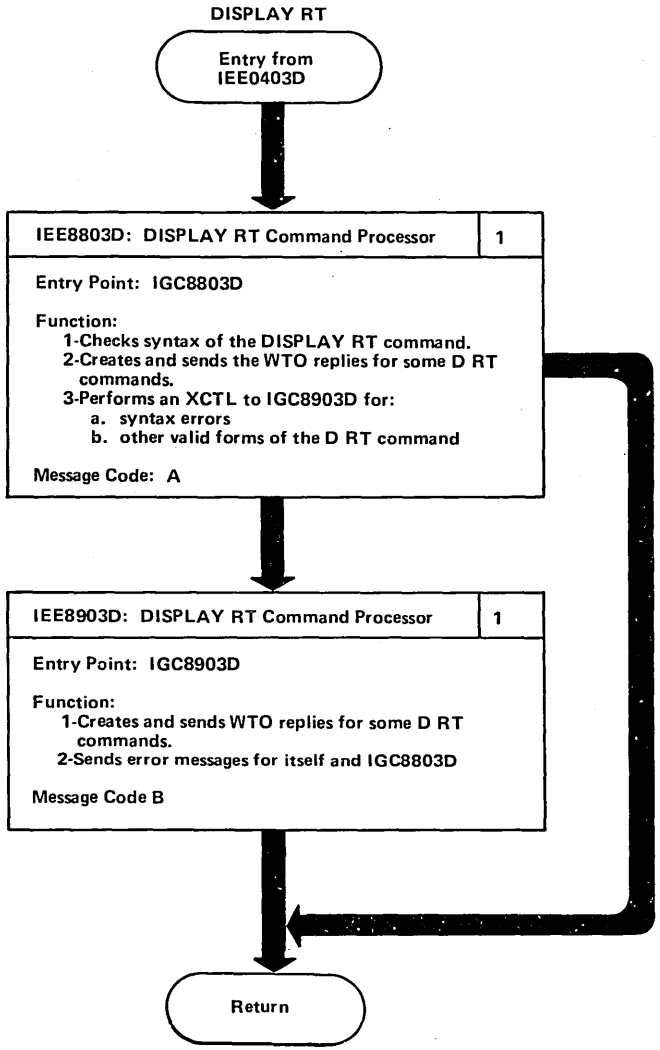
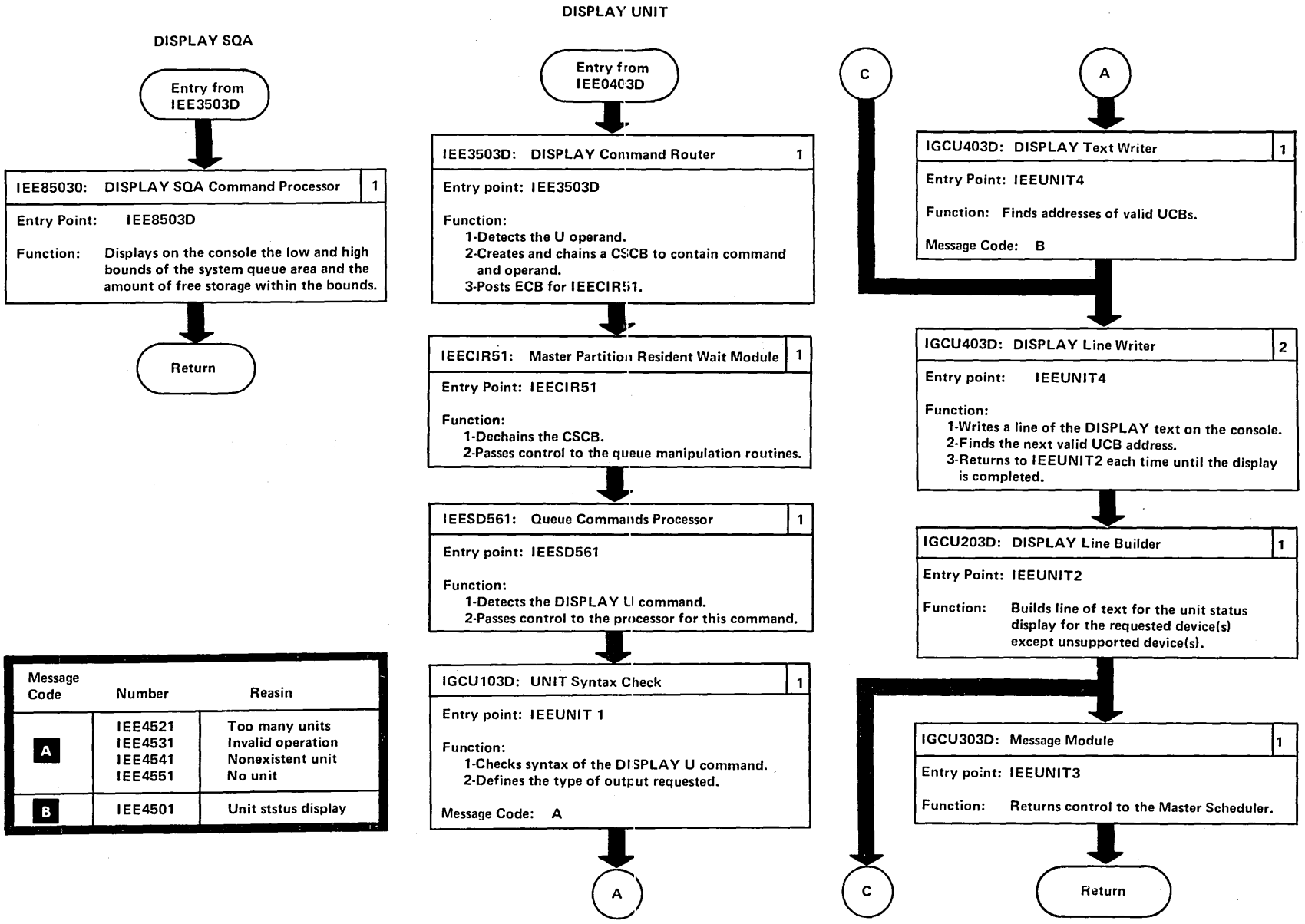


Figure 3-63. DISPLAY CONSOLES and DISPLAY RT Commands (Part 2 of 2)



Message Code	Number	Reason
A	IEE252I	Invalid termid
	IEE253I	Invalid device
	IEE374I	Invalid DISPLAY RT
	IEE444I	Active line
	IEE445I	Inactive line
	IEE446I	Lines available, enabled, and disabled
	IEE447I	Terminal logged on
	IEE448I	Terminal not logged on
B	IEE375I	Device (not) started

Figure 3-64. Display SQA and DISPLAY Unit Commands



Message Code	Number	Reasin
A	IEE4521	Too many units
	IEE4531	Invalid operation
	IEE4541	Nonexistent unit
	IEE4551	No unit
B	IEE4501	Unit ststus display

Figure 3-65. DISPLAY Command with O, N, or Jobname

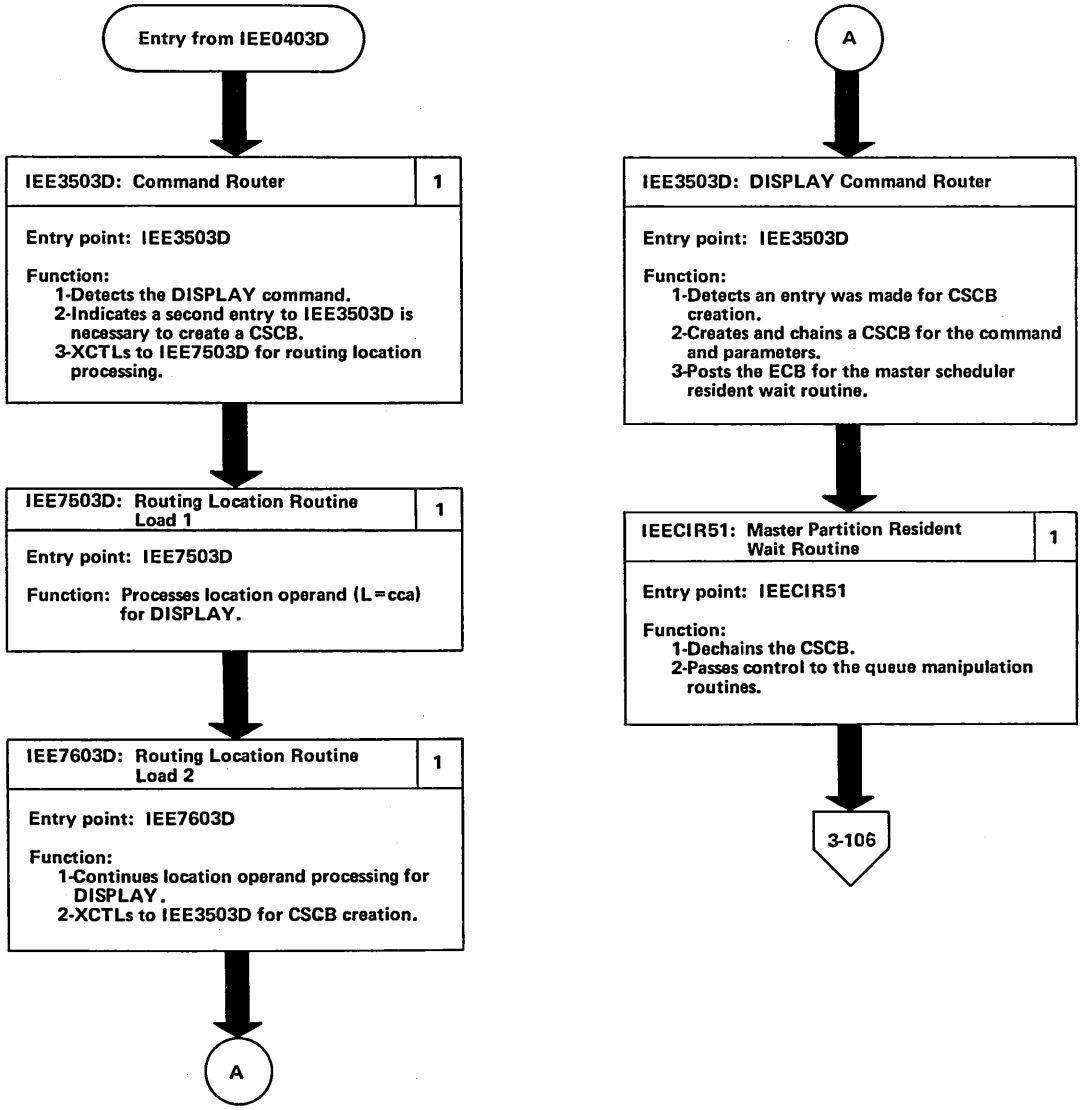
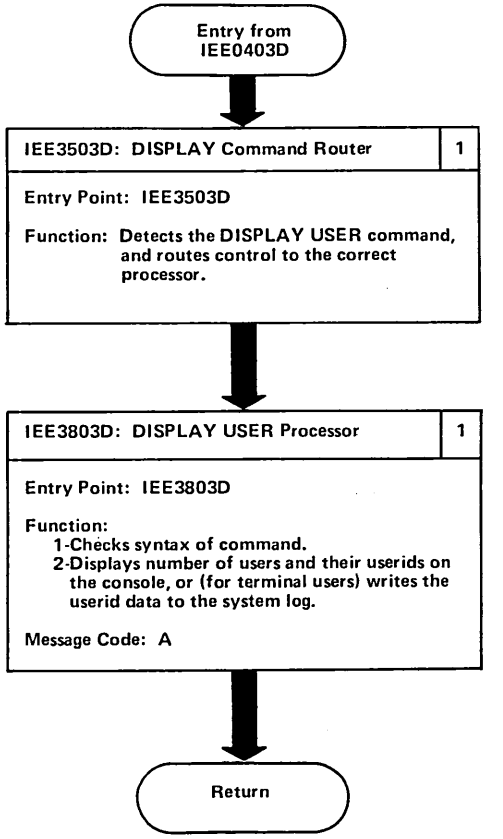
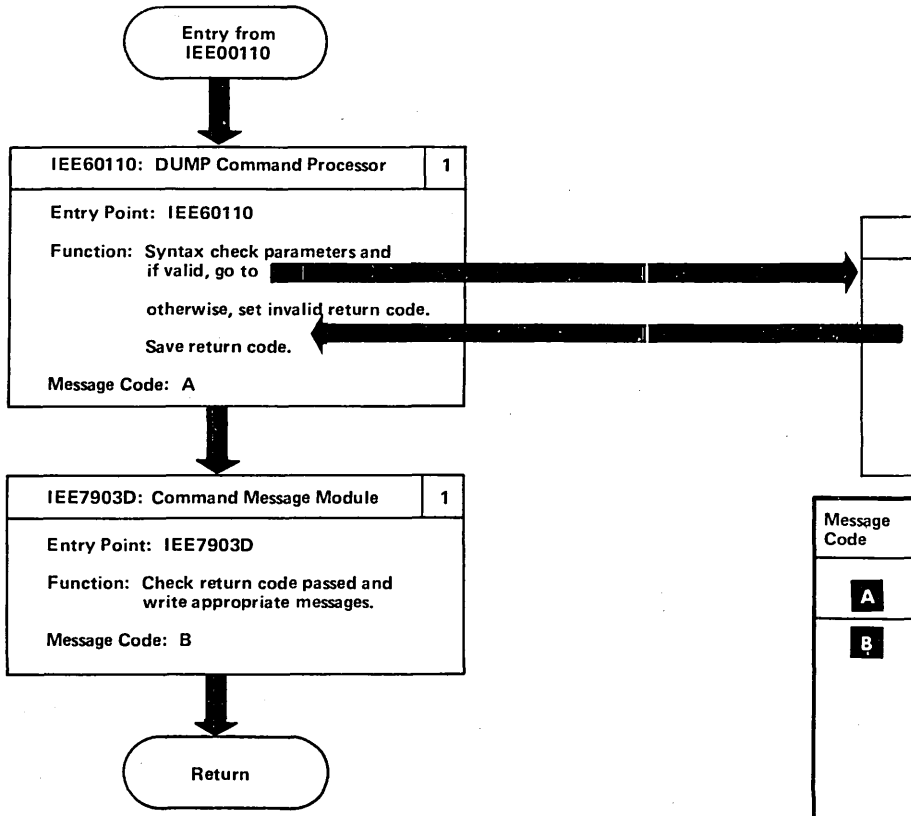


Figure 3-66. DISPLAY Command for RTAM User



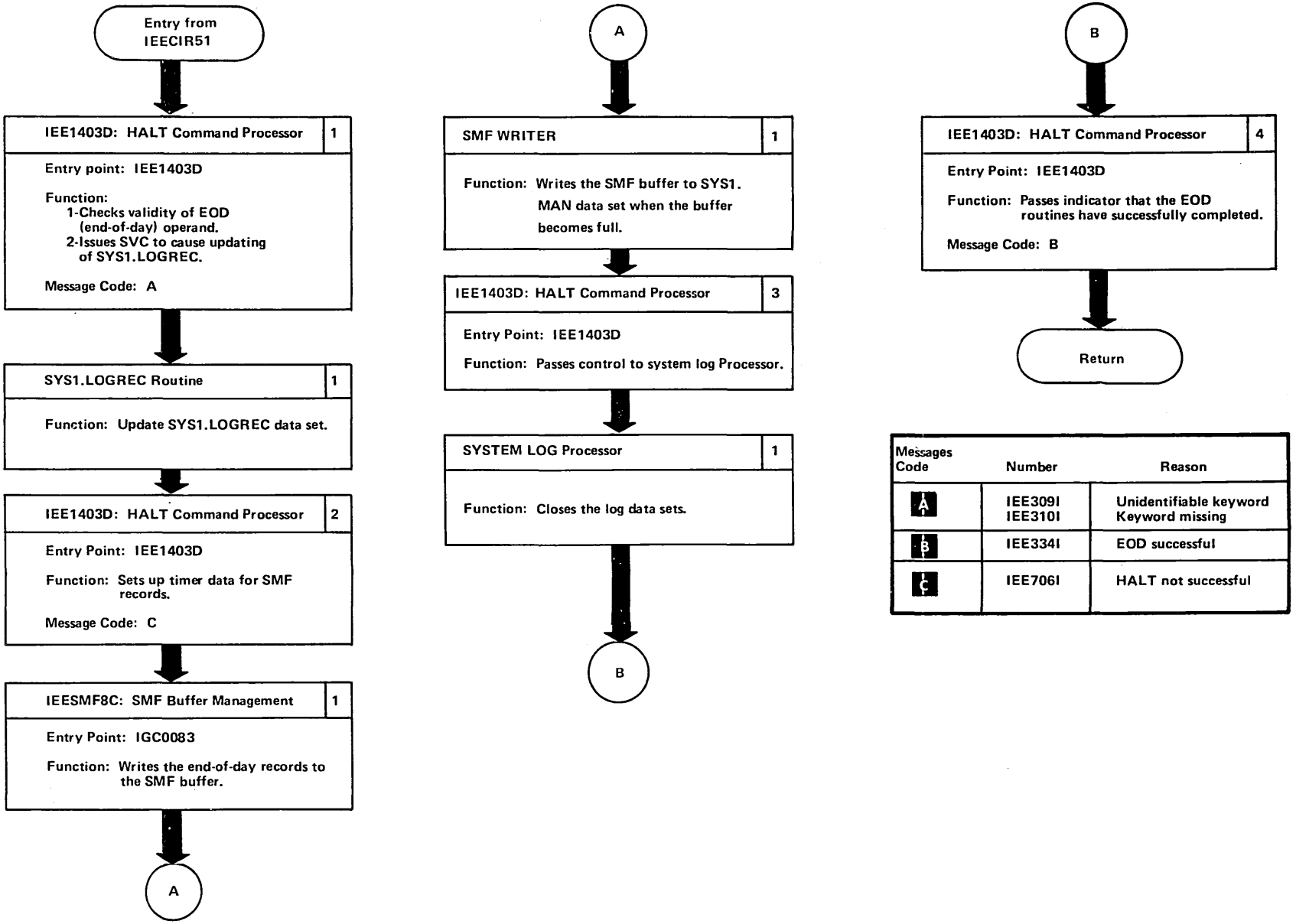
Message Code	Number	Reason
A	IEE308I	Term length error
	IEE309I	Unidentifiable keyword
	IEE347I	Invalid userid
	IEE402I	Userid not found

Figure 3-67. DUMP Command



Message Code	Number	Reason
A	IEE094D	Specify operands for DUMP command.
B	IEE090I	cm-x I/O error dump terminated.
	IEE091I	cm-x SYS1.DUMP data set full or in use.
	IEE092I	Core dump successfully completed.
	IEE093I	Op Operand is invalid reply for dump command.
	IEE095I	cm-x-retcde Invalid return code from SVC dump.
	IEE096I	cm-x SYS1.DUMP device not specified or supported.

Figure 3-68. HALT Command with EOD



Messages Code	Number	Reason
A	IEE309I IEE310I	Unidentifiable keyword Keyword missing
B	IEE334I	EOD successful
C	IEE706I	HALT not successful

Figure 3-69. HOLD Command

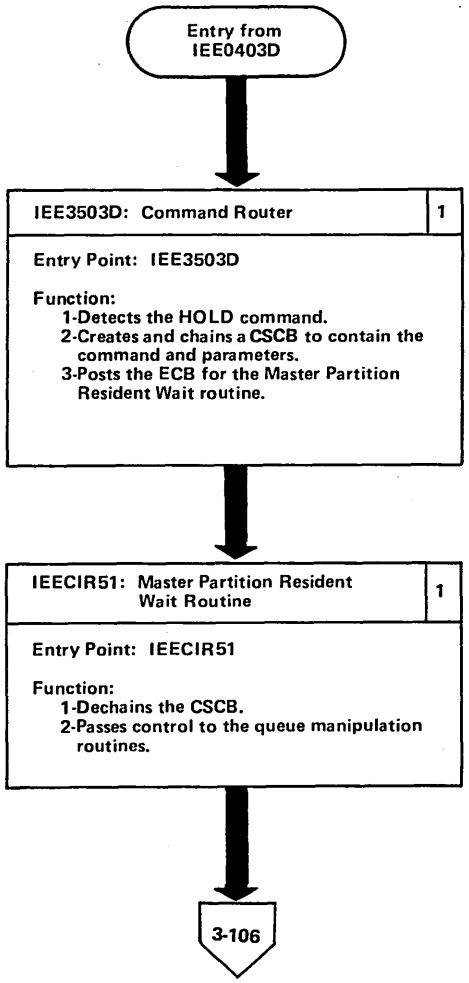
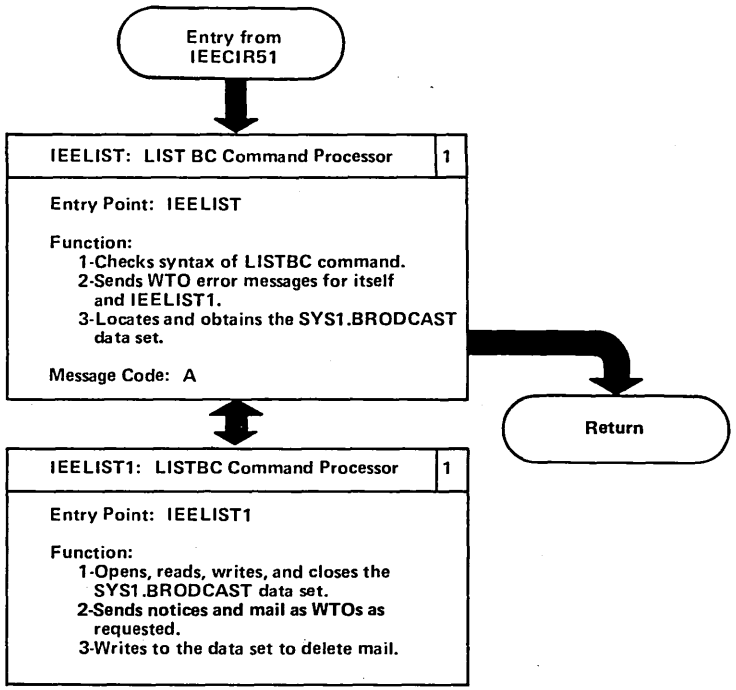
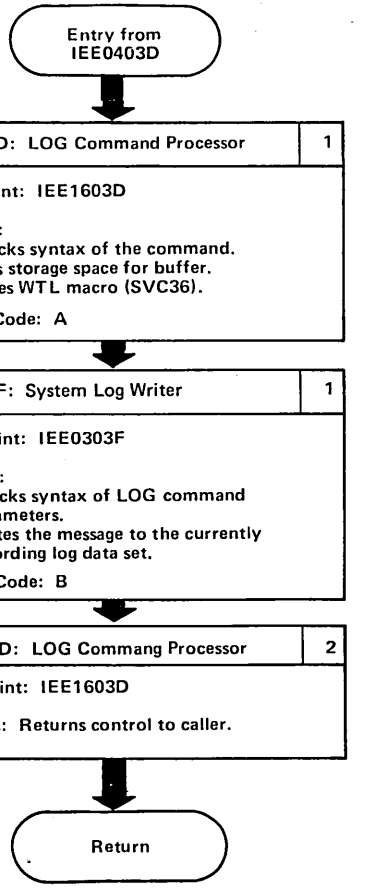


Figure 3-70. LISTBC Command



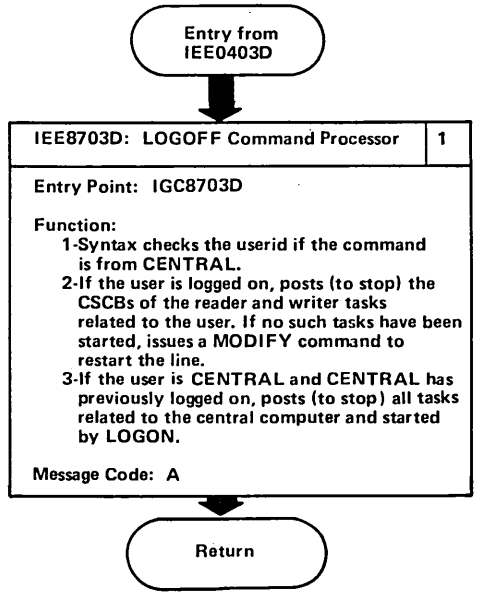
Message Code	Number	Reason
A	IEE393I	Unable to locate SYS1.BRODCAST
	IEE394I	I/O error
	IEE402I	Userid not found
	IEE439I	Unable to open SYS1.BRODCAST
	IEE440I	No mail in SYS1.BRODCAST
	IEE441I	No notices in SYS1.BRODCAST
	IEE442I	No mail for userid in SYS1.BRODCAST

Figure 3-71. LOG Command



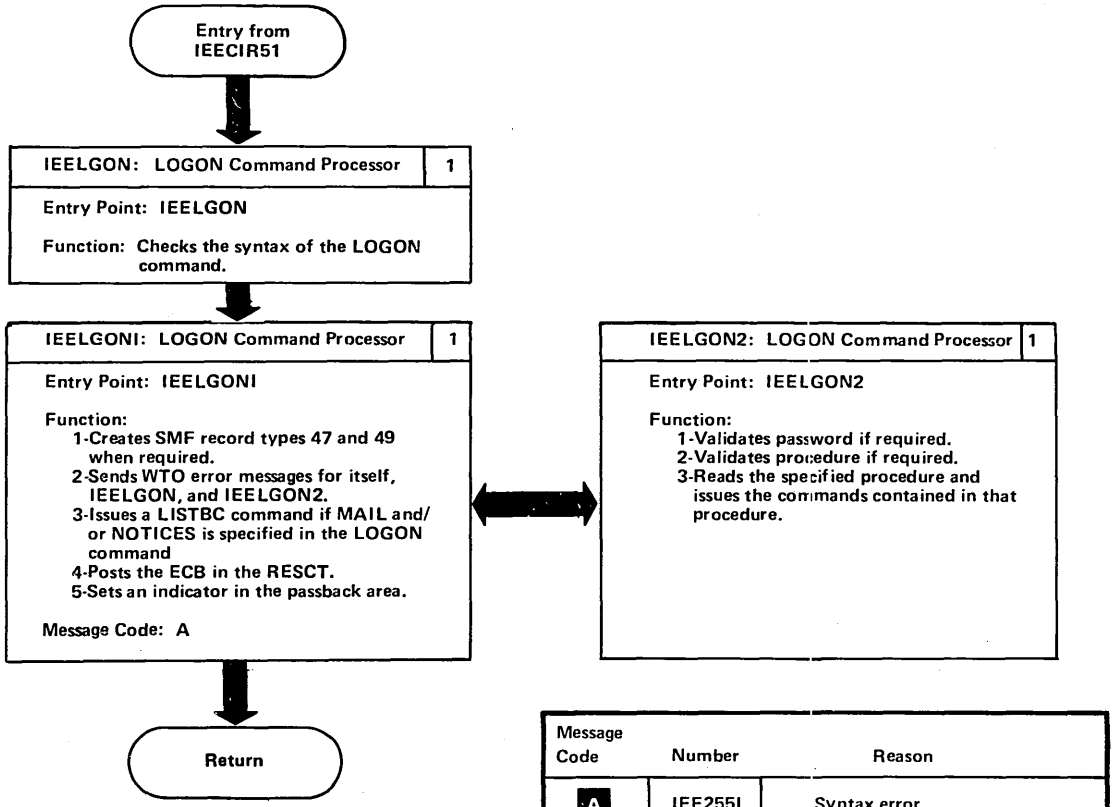
Messages Code	Number	Reason
A	IEE007I	Delimiter error
	IEE011I	Parameter missing
	IEE019I	Quote(s) missing
	IEE026I	Log not supported
	IEE033I	Command issued
	IEE319I	Text length error
B	IEE147I	LOG and WTL text
	IEE014I	I/O error on SYS1.SYSPOOL

Figure 3-72. LOGOFF Command



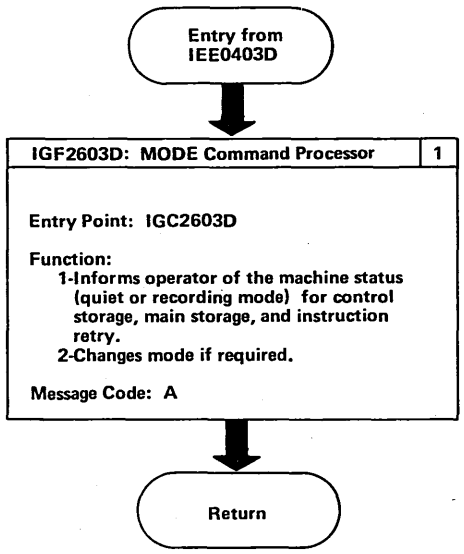
Message Code	Number	Reason
A	IEE324I	Userid not logged on
	IEE328I	Command aborted
	IEE336I	Userid logging off from RES
	IEE402I	Userid not found
	IEE437I	LOGOFF userid missing
	IEE438I	LOGOFF failed

Figure 3-73. LOGON Command



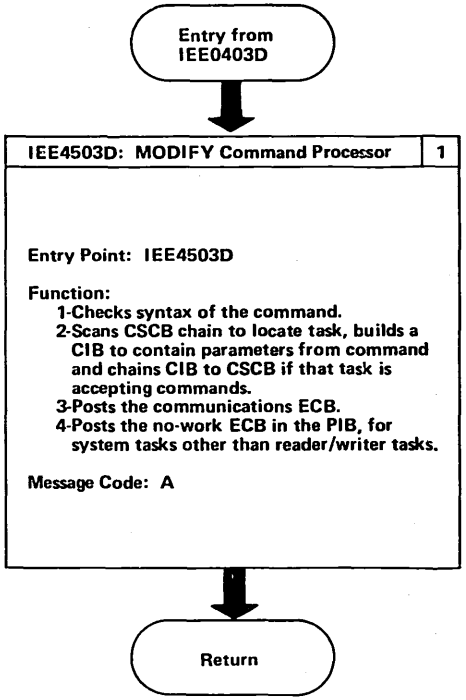
Message Code	Number	Reason
A	IEE255I	Syntax error.
	IEE337I	User id logged on to RES.
	IEE347I	Userid invalid.
	IEE372I	Unable to locate SYS1.UADS.
	IEE373I	Unable to open SYS1.UADS.
	IEE417I	Number of inactive lines.
	IEE420I	Procedure name invalid.
	IEE428I	LOGON failed.
	IEE429I	Password invalid or missing.
	IEE430I	I/O error reading SYS1. UADS.
	IEE431I	I/O error reading SYS1.PROCLIB.
	IEE432I	Uerid membername not found.
	IEE433I	Unable to open SYS1.PROCLIB.
	IEE434I	Unable to locate SYS1.PROCLIB.
	IEE435I	Command in LOGON invalid.
	IEE436I	No LOGON procedure available.
	IEE443I	Mail for userid in SYS1.BRODCAST

Figure 3-74. MODE Command



Message Code	Number	Reason
A	IEE008I	Length error
	IGF953I	Mode status
	IGF955I	Mode switch done

Figure 3-75. MODIFY Command



Message Code	Number	Reason
A	IEE341I	Task not active
	IEE342I	Rejected task busy
	IEE308I	Length error
	IEE311I	Parameter missing
	IEE305I	Invalid command

Figure 3-76. MONITOR and MONITOR/DISPLAY ACTIVE Commands

MONITOR

Entry from IEE0403D

IEE7103D: MONITOR Command Processor | 1

Entry Point: IEE7103D

Function: For DSNAME, STATUS, JOBNAMES, and SPACE, sets bits in master scheduler data area and UCME of issuing console. For ACTIVE: XCTLs to IEE7503D

Message Code: A

Return

MONITOR/DISPLAY ACTIVE

Entry from IEECIR51

IEESD566: MONITOR/DISPLAY ACTIVE Command Processor | 1

Entry Point: IEESD566

Function: Displays on the operator console (DISPLAY ACTIVE) or the specified graphics console (MONITOR ACTIVE) a list of all active tasks and jobs, the number of subtasks associated with each task, the status of each job (active/inactive) and the deactivation characteristics of each job (eligible or ineligible).

Message Code: B

Return

Entry from IEE3503D or IEE7103D

IEE7503D: Routing Location Routine (Load 1) | 1

Entry point: IEE7503D

Function: Processes location operand (L=cca) and message routing defaults for CONTROL, DISPLAY, MONITOR, and STOPMN commands.

Message Code: C, D, F

IEE7603D: Routing Location Routine (Load 2) | 1

Entry point: IEE7603D

Function: Continues processing of the location operand (L=cca) for the DISPLAY, MONITOR, and STOPMN commands.

Message Code: C, E

Return

Message Code	Number	Reason
A	IEE305I IEE311I IEE402I IEE251I	Command syntax invalid Parameter missing Userid not found Parameter invalid from remote user
B	IEE328I IEE821I	Insufficient SQA space for tables JOBNAME, STEPNAME, number of ATTACHs, page supervisor status
C	IEE156I	Invalid operand-xxxx.
D	IEE305I IEE345I	Invalid command. Authority invalid.
E	IEE921I IEE926I	Rejected – Display already exists Display area busy No monitor to stop Needs display area, L=cca operand invalid– Dynamic display in area
F	IEE926I	L=cca operand invalid– Not an area Not a console Full capability.

Figure 3-77. MOUNT Command

Entry from IEE0403D

IEE1903D MOUNT Command Processor | 1

Entry Point: IEE1903D

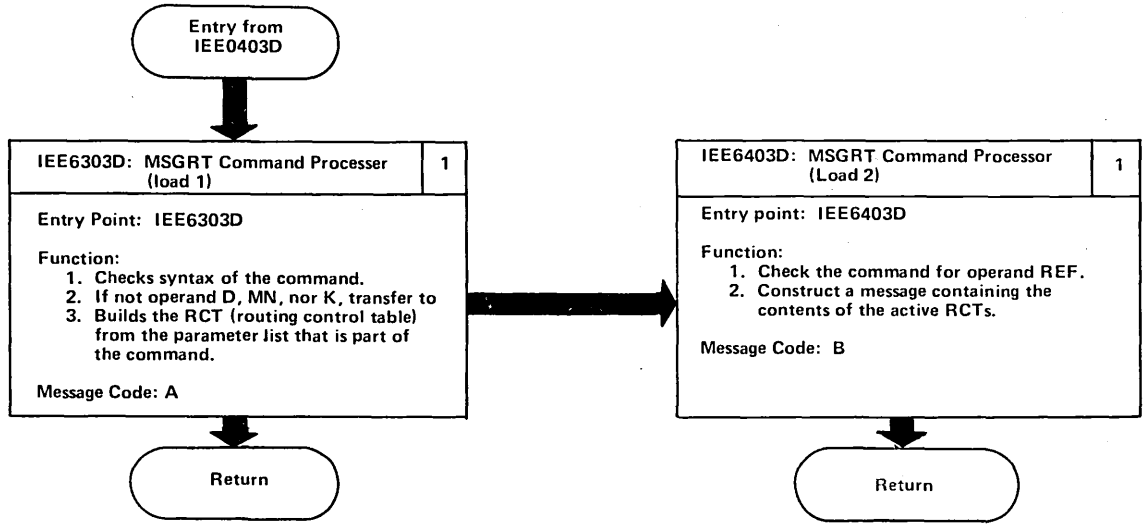
Function:
1-Chains CSCB and TRCB.
2-Posts the no-work ECB of the first available partition.

Message Code: A

3-12

Message Code	Number	Reason
A	IEE308I IEE311I IEE378I IEE379I	Length error Parameter missing Partition does not exist Partition not available

Figure 3-78. MSGRT Command



Message Code	Number	Reason
A	IEE137I IEE156I IEE345I IEE925I IEE926I IEE931I	Valid only in MCS. Invalid operand. Authority invalid. Operand for non-CRT only. L=cca operand invalid. {Not an area. {Not a console. Insufficient storage for command.
B	IEE156I IEE930I	Invalid operand. 'text' (Display of the contents of the RCT in a format resembling the operands of the MSGRT command.)

Figure 3-79. RELEASE Command

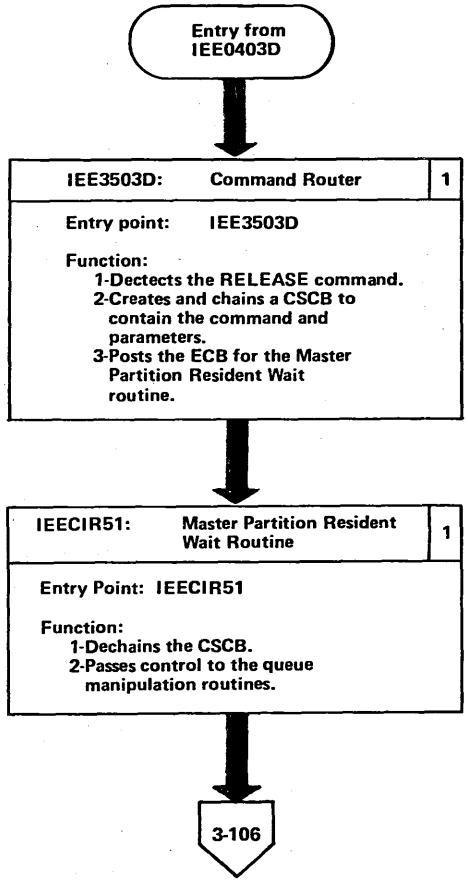
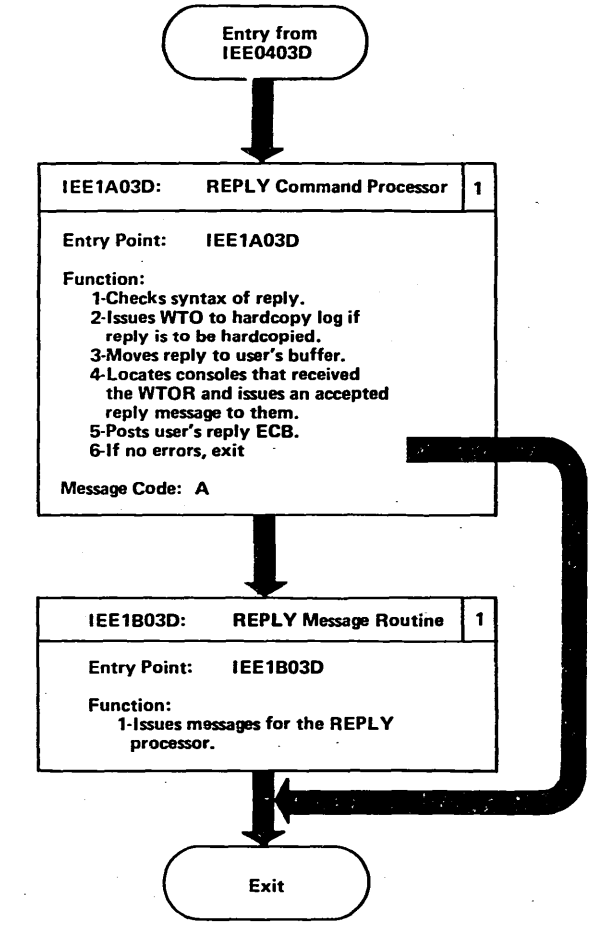


Figure 3-80. REPLY Command



Message Code	Number	Reason
A	IEE600I	Reply message text
	IEE701I	Reply unnecessary
	IEE702I	Syntax error in reply ID
	IEE703I	Unauthorized source
	IEE704I	Invalid ID

Figure 3-81. RESET Command

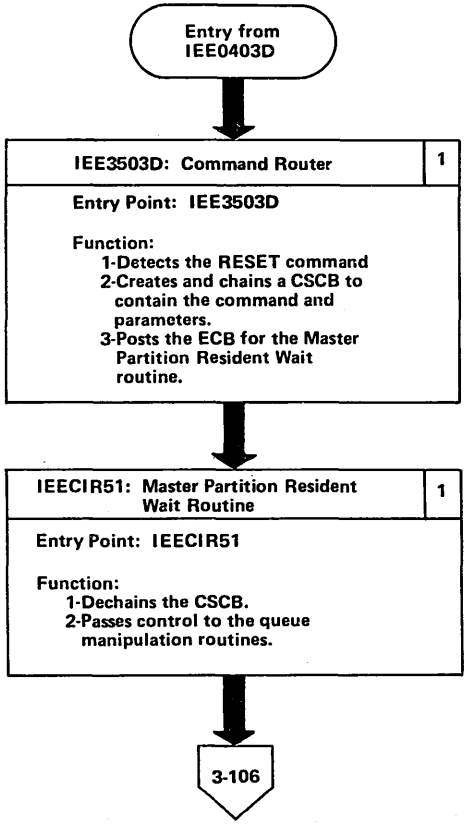
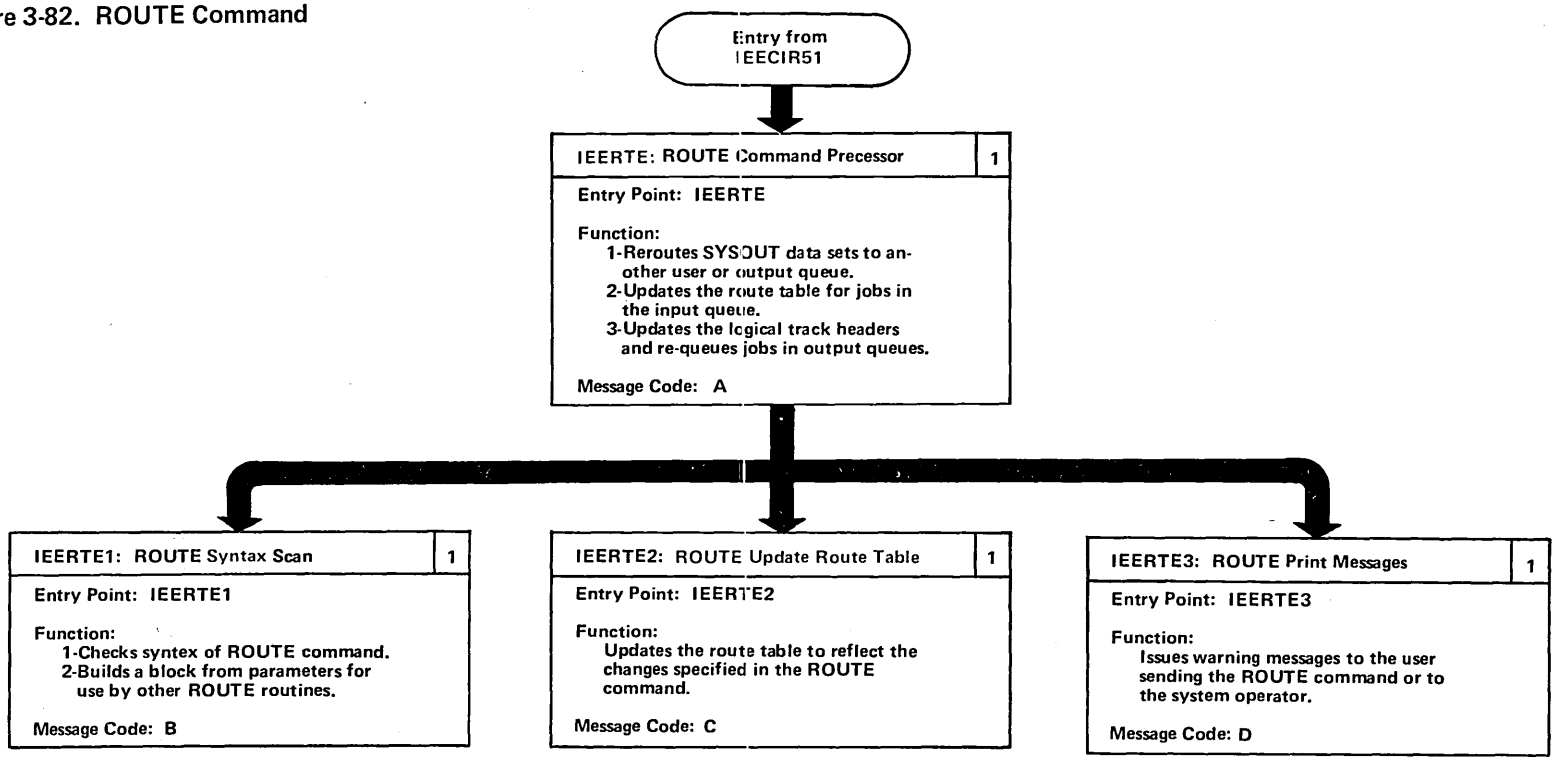
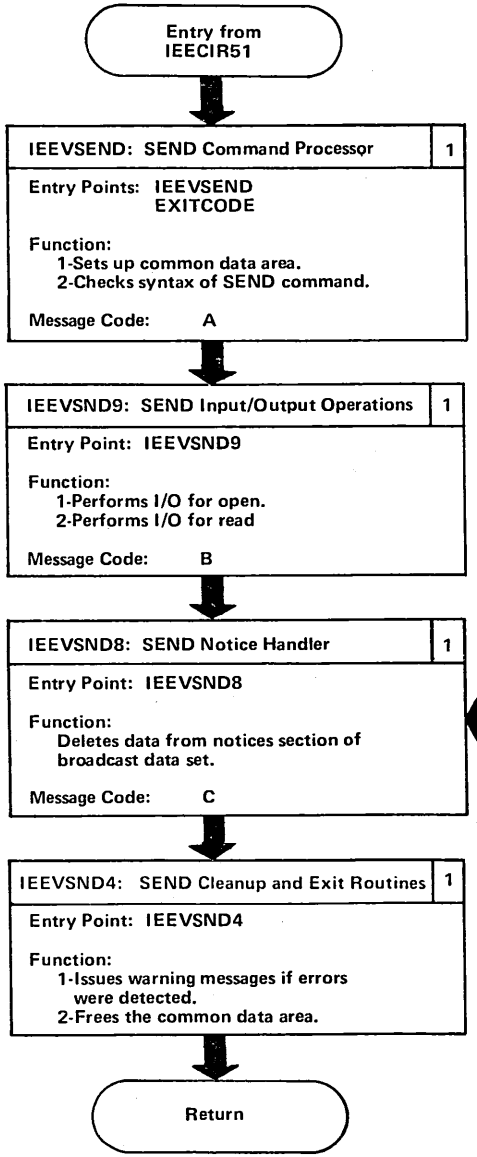


Figure 3-82. ROUTE Command



Message Code	Number	Reason
A	IEE254I	I/O error.
	IEE392I	No data sets to route.
	IEE398I	Data sets successfully routed.
	IEE474I	Route rejected.
B	IEE251I	Remote parameter invalid.
	IEE255I	Syntax error.
	IEE310I	Keyword missing.
	IEE347I	Userid invalid.
	IEE396I	Output class invalid.
	IEE471I	Cannot route to this user.
C	IEE254I	I/O error.
	IEE399I	Route table error.
	IEE414I	Output already selected by DSO.
D	IEE416I	Number of active RES terminals.

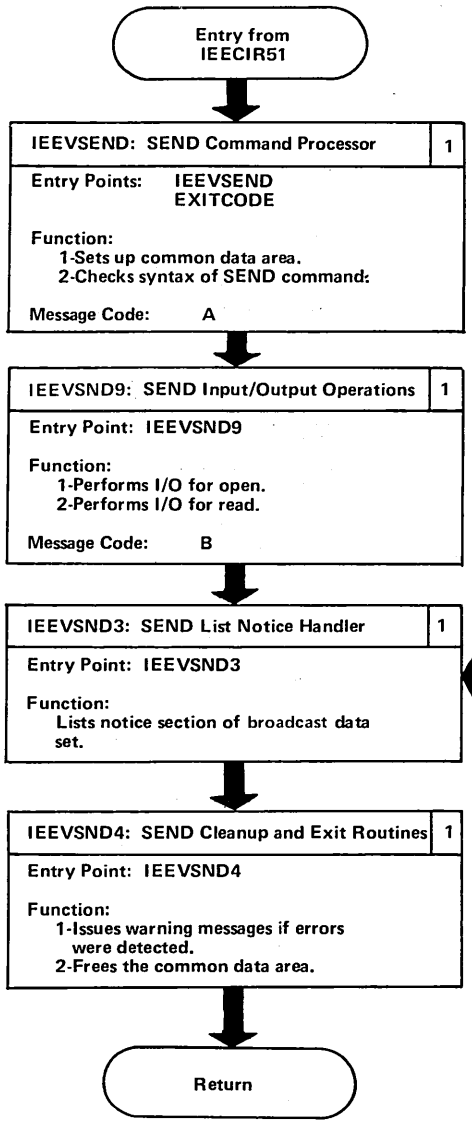
Figure 3-83. SEND Delete Command



Note: IEEVSND6, which contains message text only, is always a part of IEEVSND4.

Message Code	Number	Reason
A	IEE473I	Insufficient storage.
B	IEE464I IEE473I	SYS1.BRODCAST unusable. Insufficient storage.
C	IEE466I IEE472I	No space in SYS1.BRODCAST. Userid invalid.
D	IEE449I IEE465I IEE466I IEE468I	Message deleted from NOTICES. No broadcast messages. No space in SYS1.BRODCAST. Message entered in NOTICES.

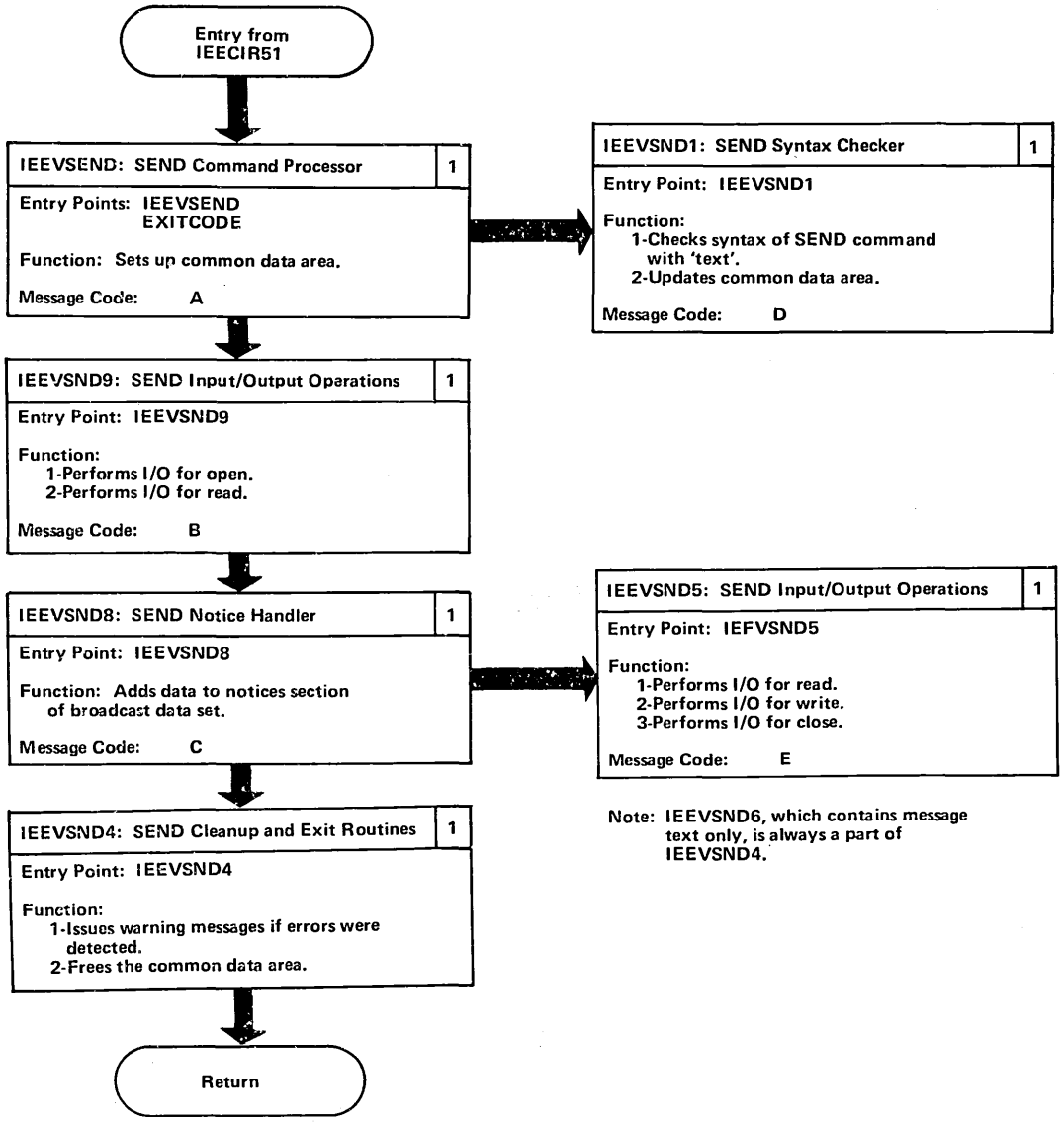
Figure 3-84. SEND List Command



Note: IEEVSND6, which contains message text only, is always a part of IEEVSND4.

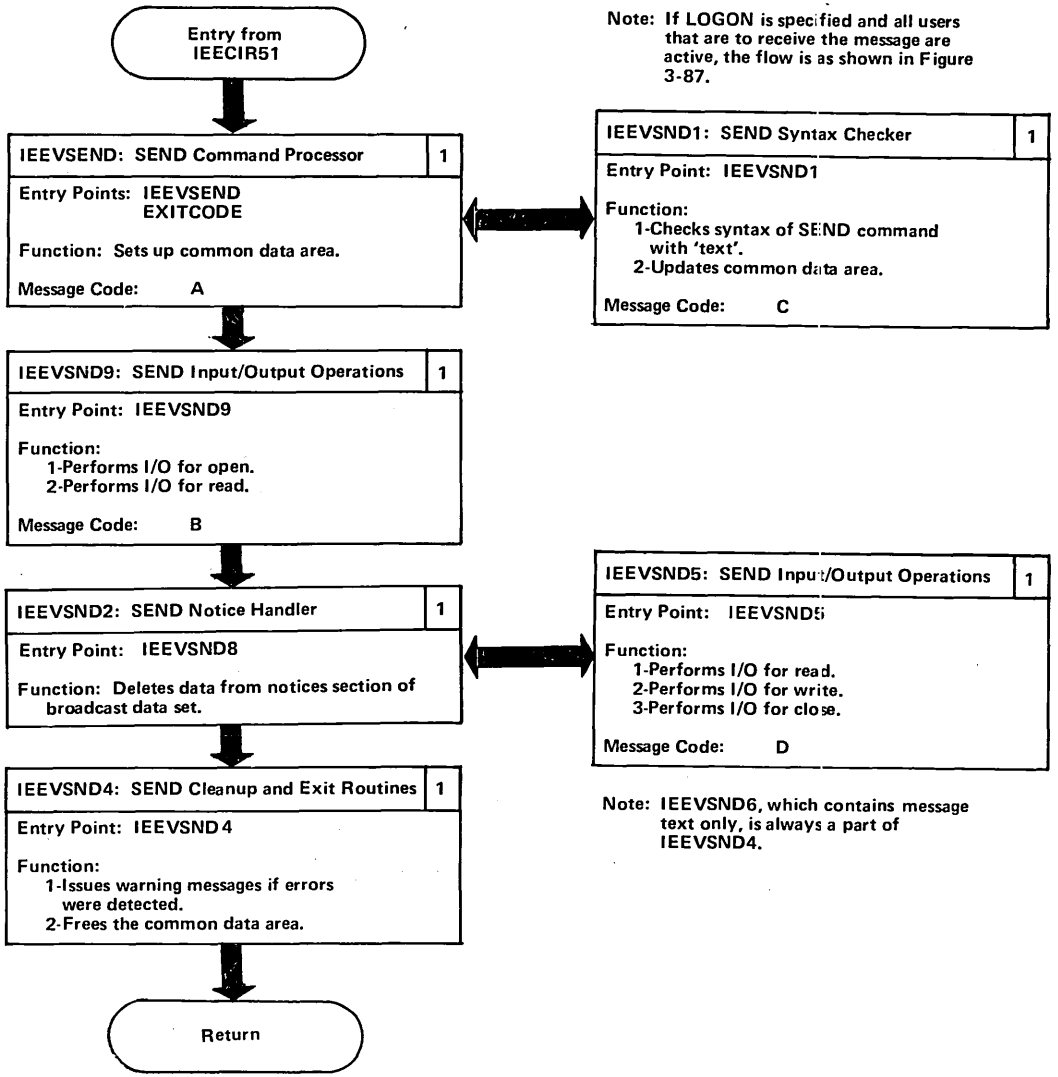
Message Code	Number	Reason
A	IEE473I	Insufficient storage.
B	IEE4641 IEE473I	SYS1.BRODCAST unusable. Insufficient storage.
C	IEE449I IEE465I IEE466I IEE468I	Message deleted from NOTICES. No broadcast messages. No space in SYS1.BRODCAST Message entered in NOTICES.

Figure 3-85. SEND 'text' ALL Command with SAVE or LOGON Operands



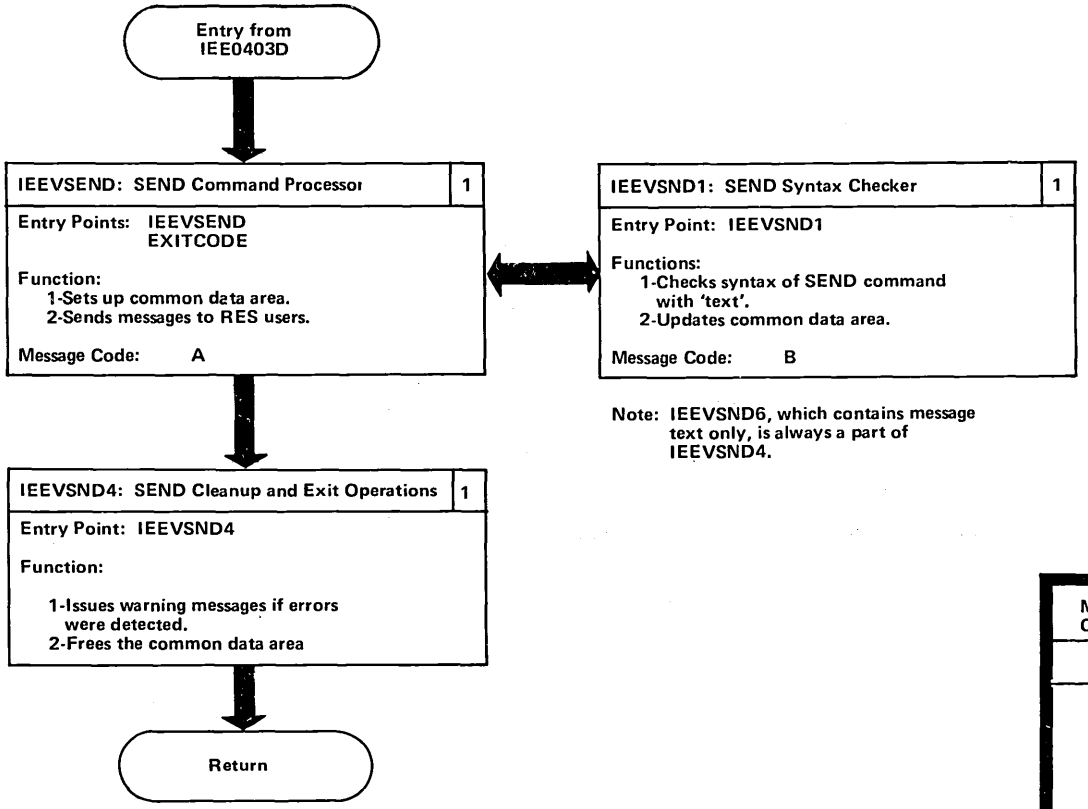
Message Code	Number	Reason
A	IEE473I	Insufficient storage.
B	IEE464I	SYS1.BRODCAST unusable.
	IEE473I	Insufficient storage.
C	IEE466I	No space in SYS1.BRODCAST.
	IEE472I	Userid invalid.
D	IEE312I	Parameters conflict.
	IEE411I	Message sent to central operator only.
	IEE412I	'USER=' parameter missing.
	IEE413I	Excessive number of userids.
	IEE467I	Message truncated.
	IEE472I	Userid invalid.
	IEE473I	Insufficient storage.
E	IEE449I	Message deleted from NOTICES.
	IEE465I	No broadcast messages.
	IEE466I	No space in SYS1.BRODCAST.
	IEE468I	Message entered in NOTICES.

Figure 3-86. SEND 'text' USER Command with SAVE or LOGON Operands



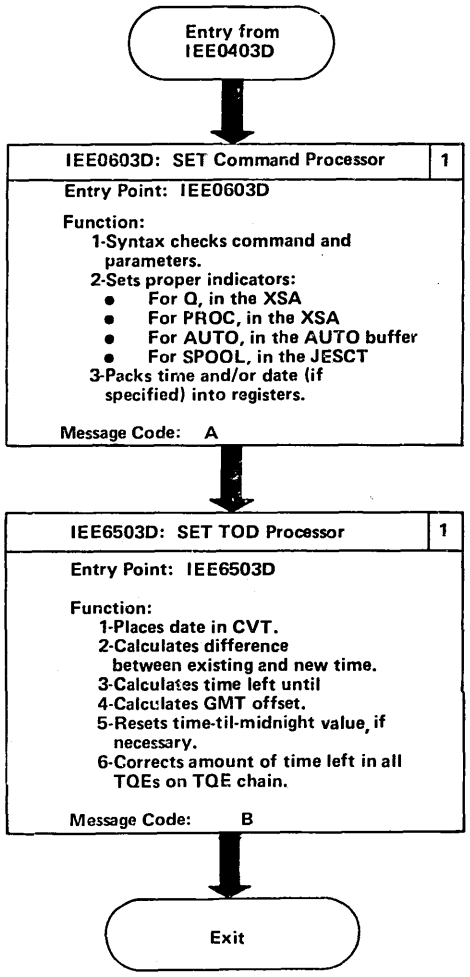
Message Code	Number	Reason
A	IEE473I	Insufficient storage.
B	IEE464I	SYS1.BRODCAST unusable.
	IEE473I	Insufficient storage.
C	IEE312I	Parameters conflict.
	IEE411I	Message sent to central operator only.
	IEE412I	'USER=' parameter missing.
	IEE413I	Excessive number of userids.
	IEE467I	Message truncated.
	IEE472I	Userid invalid.
	IEE473I	Insufficent storage.
D	IEE449I	Message deleted from messages.
	IEE465I	No broadcast messages.
	IEE466I	No space in SYS1.BRODCAST.
	IEE468I	Message entered in NOTICES.

Figure 3-87. SEND'text' ALL and SEND 'text' USER Commands with or without NOW Operand



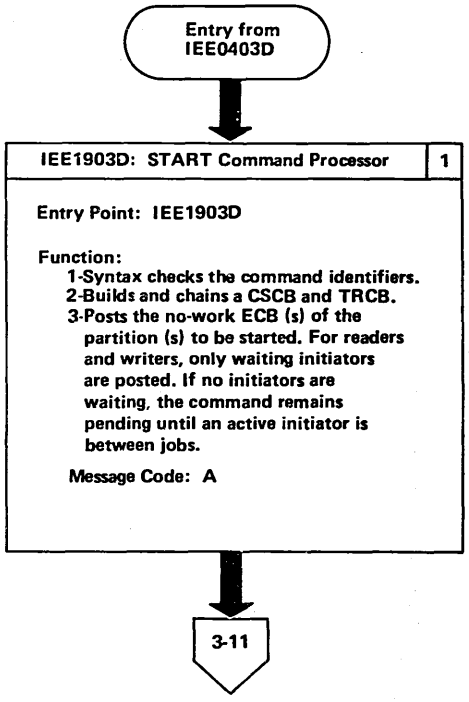
Message Code	Number	Reason
A	IEE473I	Insufficient storage.
B	IEE312I	Parameters conflict.
	IEE411I	Message sent to central operator only.
	IEE412I	'USER=' parameter missing.
	IEE413I	Excessive number of userids.
	IEE467I	Message truncated.
	IEE472I	Userid invalid.
	IEE473I	Insufficient storage.

Figure 3-88. SET Command



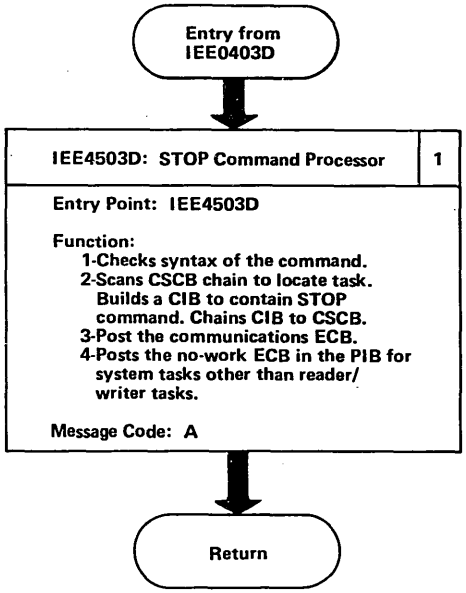
Messages Code	Number	Reason
A	IEE305I	Command invalid
	IEE006I	Invalid numerics
	IEE007I	Delimiter error
	IEE008I	Length error
	IEE009I	Keyword misspelled
	IEE010I	Keyword missing
	IEE011I	Parameter missing
	IEE012I	Parameters conflict
	IEE013I	Invalid unit
B	IEE118I	SET parameters accepted

Figure 3-89. START Command



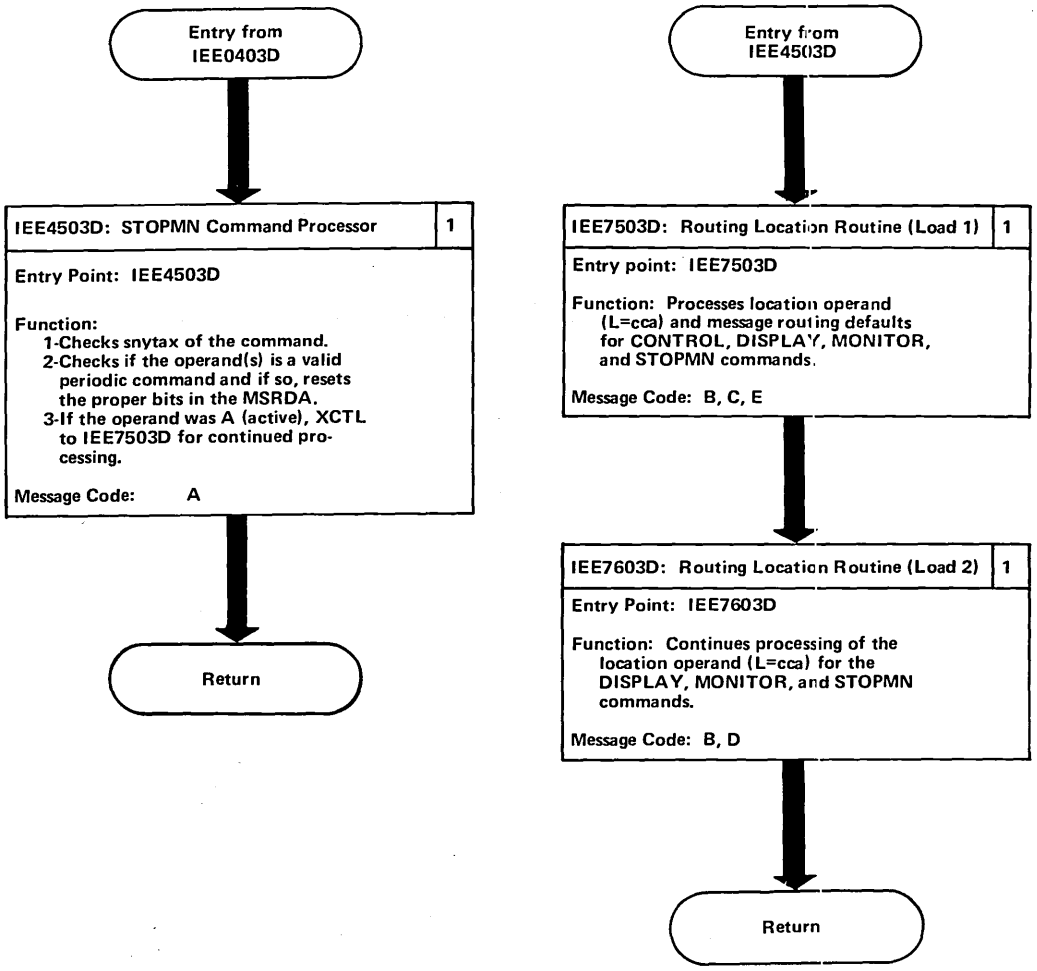
Message Code	Number	Reason
A	IEE308I	Length error
	IEE311I	Parameter missing
	IEE378I	Partition does not exist
	IEE379I	Partition not available
	IEE813I	START FAILED

Figure 3-90. STOP Command



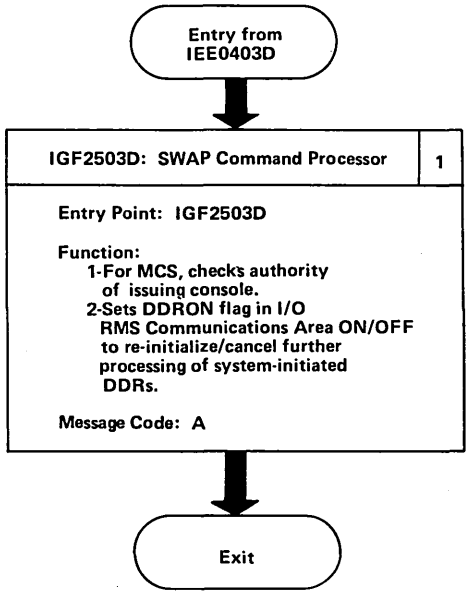
Message Code	Number	Reason
A	IEE341I	Task not active
	IEE342I	Rejected task busy
	IEE308I	Length error
	IEE311I	Parameter missing
	IEE305I	Invalid command

Figure 3-91. STOPMN Command



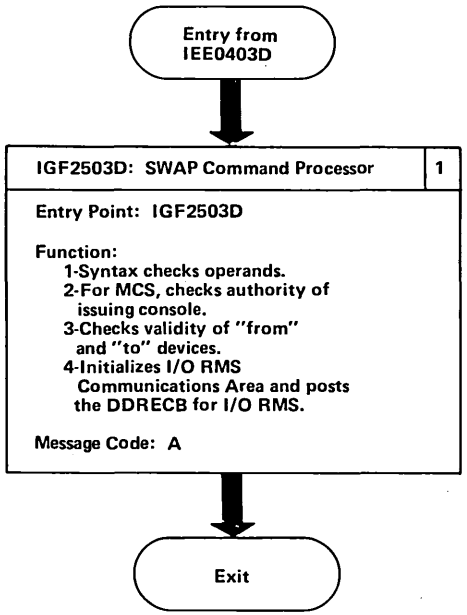
Message Code	Number	Reason
A	IEE305I IEE308I IEE311I	Invalid command. Length error. Parameter missing
B	IEE156I	Invalid operand - xxxx.
C	IEE305I IEE345I	Invalid command. Authority invalid.
D	IEE921I IEE926I	Rejected - {Display already exists Display area busy No monitor to stop Needs display area.} L=cca operand invalid - Dynamic display in area
E	IEE926I	L=cca operand invalid - {Not an area Not a console Full capability.}

Figure 3-92. SWAP (ON/OFF) Command



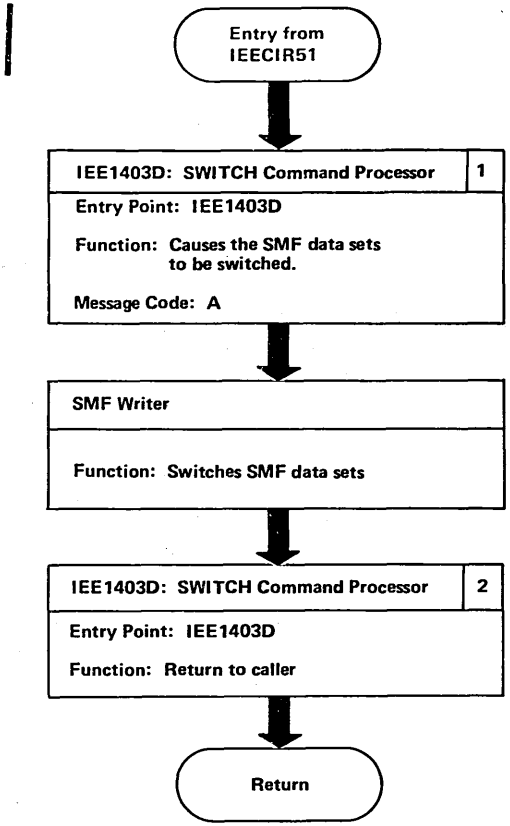
Message Code	Number	Reason
A	IEE307I	Delimiter error
	IEE311I	Parameter missing
	IEE313I	Invalid unit
	IEE026I	SYSRES not supported
	IEE345I	Invalid authority
	IEE380I	Invalid device type
	IEE381I	Unallocated device
	IEE382I	Device currently active

Figure 3-93. SWAP Devices Command



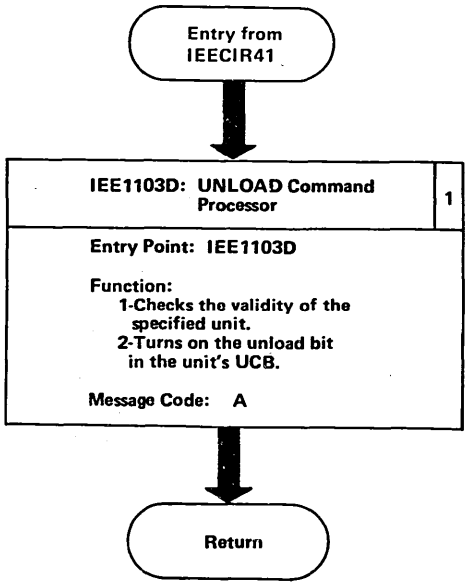
Message Code	Number	Reason
A	IEE307I	Delimiter error
	IEE311I	Parameter missing
	IEE313I	Invalid unit
	IEE026I	SYSRES not supported
	IEE345I	Invalid authority
	IEE380I	Invalid device type
	IEE381I	Unallocated device
	IEE382I	Device currently active

Figure 3-94. SWITCH Command



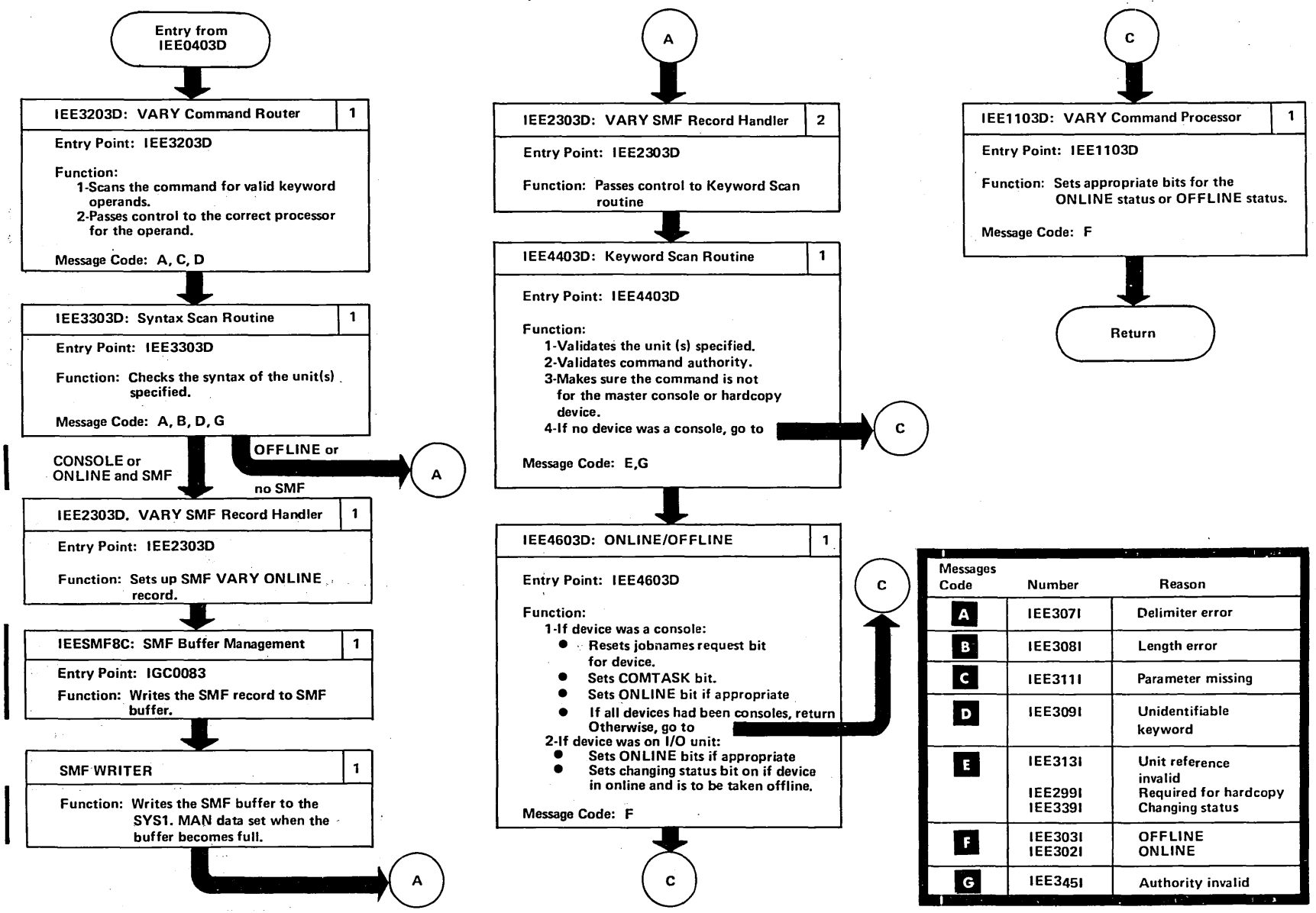
Message Code	Number	Reason
A	IEE706I	Switch not successful

Figure 3-95. UNLOAD Command



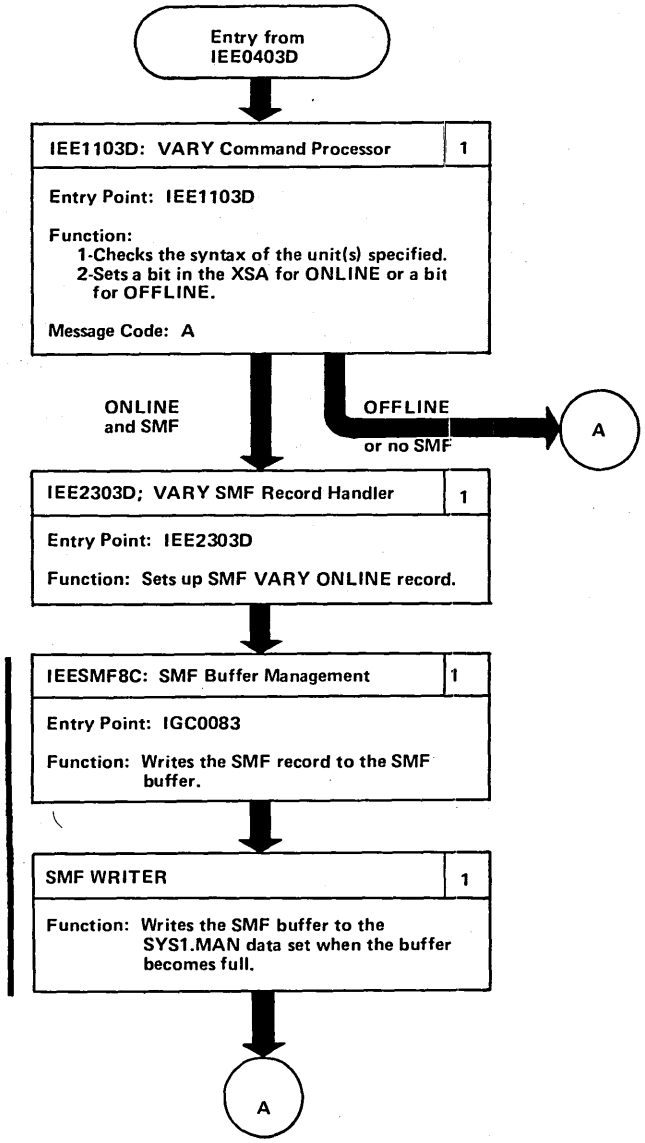
Message Code	Number	Reason
A	IEE307I IEE308I IEE311I	Delimiter error Length error Parameter missing

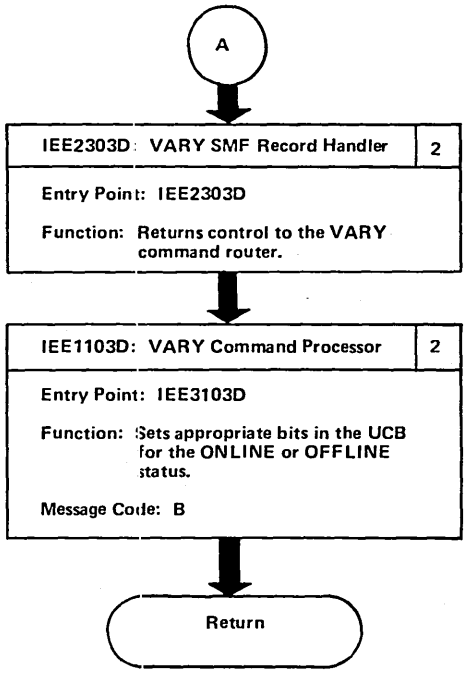
Figure 3-96. VARY an I/O Device ONLINE/OFFLINE, MCS



Messages Code	Number	Reason
A	IEE307I	Delimiter error
B	IEE308I	Length error
C	IEE311I	Parameter missing
D	IEE309I	Unidentifiable keyword
E	IEE313I	Unit reference invalid
	IEE299I	Required for hardcopy
	IEE339I	Changing status
F	IEE303I	OFFLINE
	IEE302I	ONLINE
G	IEE345I	Authority invalid

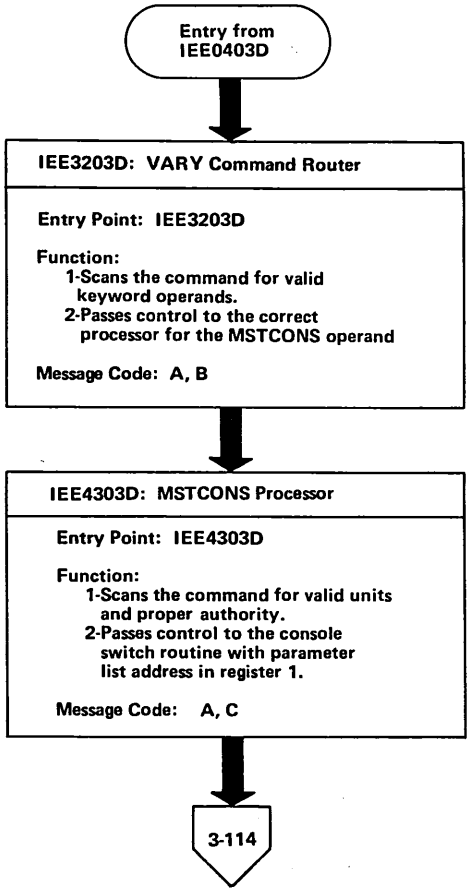
Figure 3-97. VARY an I/O Device ONLINE/OFFLINE, non-MCS





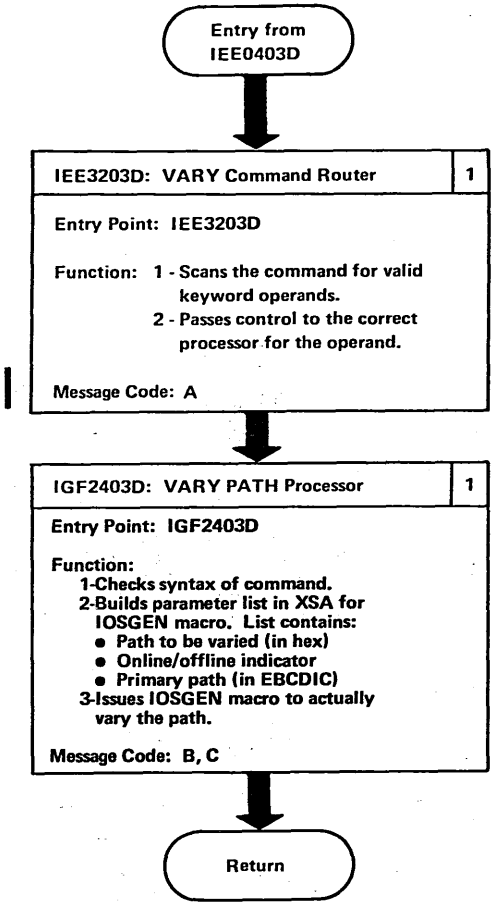
Messages Code	Number	Reason
A	IEE307I IEE311I IEE309I	Delimiter error Parameter missing Unidentifiable keyword
B	IEE302I IEE303I	ONLINE OFFLINE

Figure 3-98. VARY MSTCONS Command



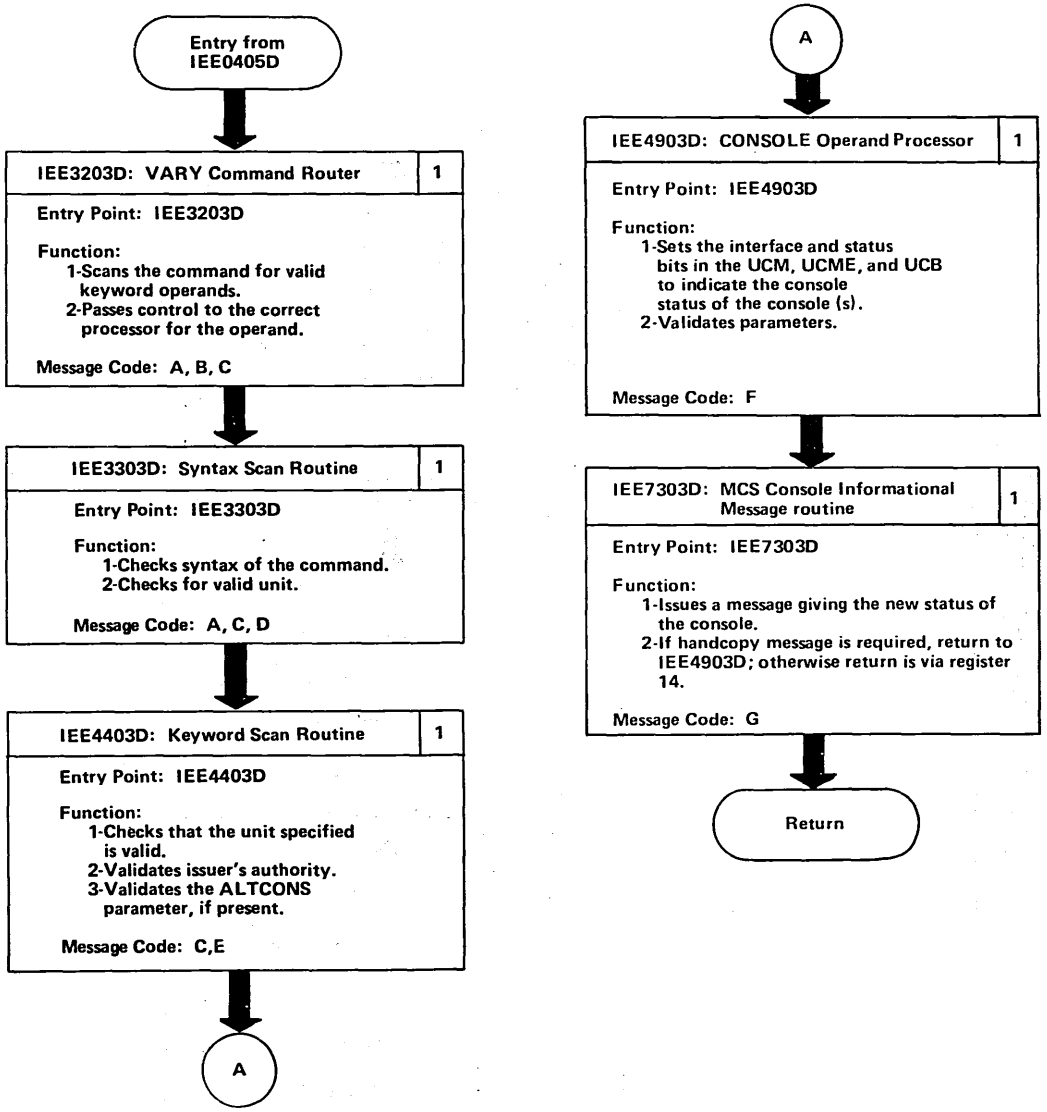
Messages Code	Number	Reason
A	IEE307I	Delimiter error
B	IEE309I	Unidentifiable keyword
	IEE311I	Parameter missing
C	IEE339I	Device changing status
	IEE313I	Unit reference invalid
	IEE345I	Authority invalid

Figure 3-99. VARY PATH Command



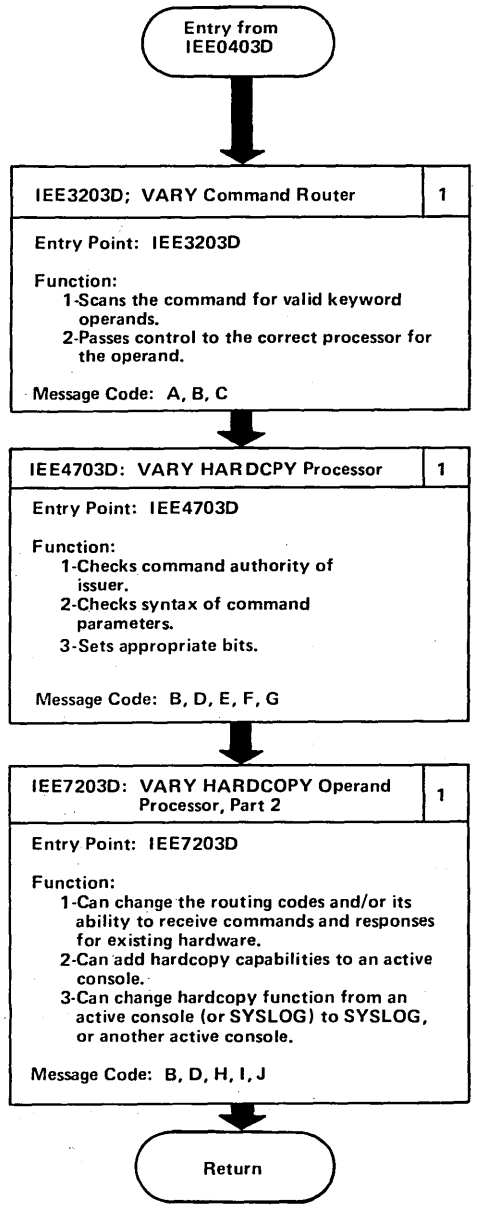
Message Code	Message Number	Reason
A	IEE311I	Parameter missing
B	IEE309I	Unidentifiable keyword
	IEE307I	Invalid delimiter
C	IEE302I	Path now online
	IEE303I	Path now offline
	IEE313I	Invalid unit reference
	IEE345I	Invalid authority
	IEE376I	Last path to device
	IEE377I	TP device - not varied
	IEE378I	Invalid path
	IEE379I	Device unavailable

Figure 3-100. VARY CONSOLE Command



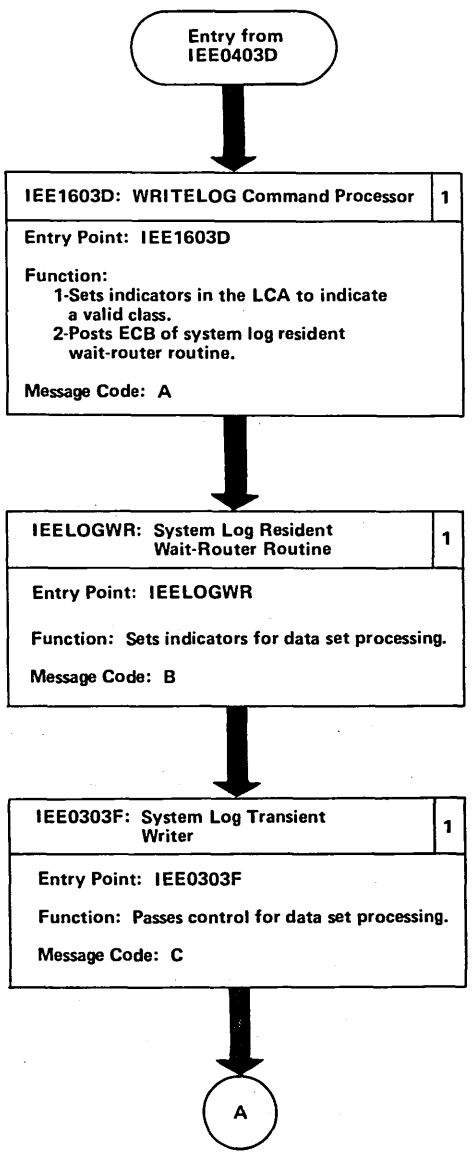
Message Code	Number	Reason
A	IEE307I	Delimiter error
B	IEE311I	Parameter missing
C	IEE309I	Unidentifiable keyword
D	IEE308I IEE345I	Length error Authority invalid
E	IEE306I IEE313I IEE339I IEE300I	Invalid numerics Invalid unit Changing status Invalid alternate console
F	IEE349I	Console description
G	IEE931I	Insufficient storage

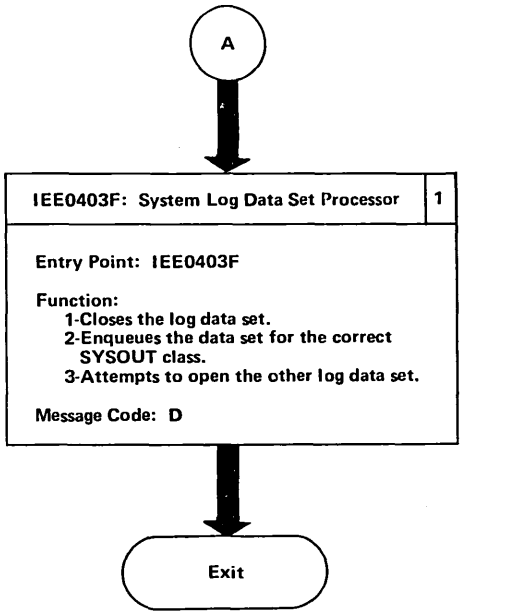
Figure 3-101. VARY HARDCOPY Command



Message Code	Number	Reason
A	IEE307I	Delimiter error
B	IEE309I	Unidentifiable keyword
C	IEE311I	Parameter missing
D	IEE345I	Authority invalid
E	IEE313I	Invalid unit
F	IEE306I	Invalid numerics
G	IEE229I IEE338I	Required for hardcopy Inactive as hardcopy
H	IEE026I	Log not supported
I	IEE339I IEE349I	Device changing status Hardcopy console
J	IEE931I	Insufficient storage

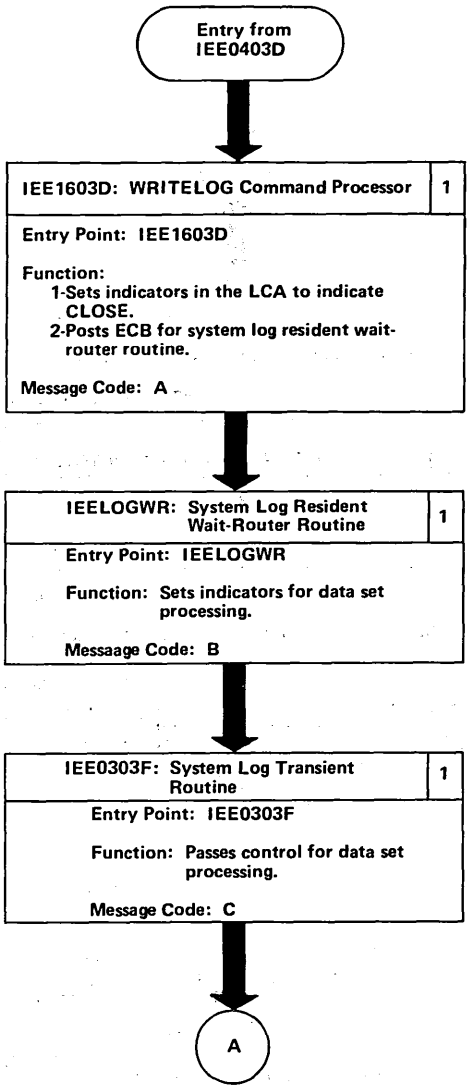
Figure 3-102. WRITELOG Command with Class

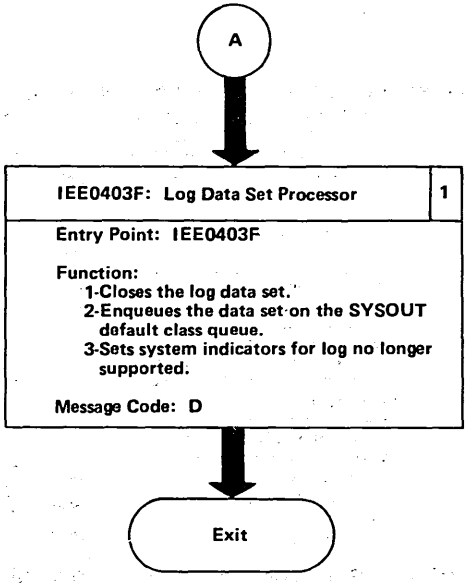




Messages Code	Number	Reason
A	IEE007I	Delimiter error
	IEE011I	Parameter missing
	IEE019I	Quote(s) missing
	IEE023I	Classname error
	IEE026I	Log not supported
	IEE032I	WRITELOG pending
	IEE033I	Command issued
	IEE034I	Execution deferred
	IEE319I	Text length error
B	IEE044I	Abend completion code
	IEE037I	Log not supported
C	IEE014I	I/O error on SYS1.SYSPOOL
D	IEE043I	Sysout class
	IEE045I	Log inactive
	IEE009I	Log recording
	IEE010I	Log closed
	IEE018I	Data set empty

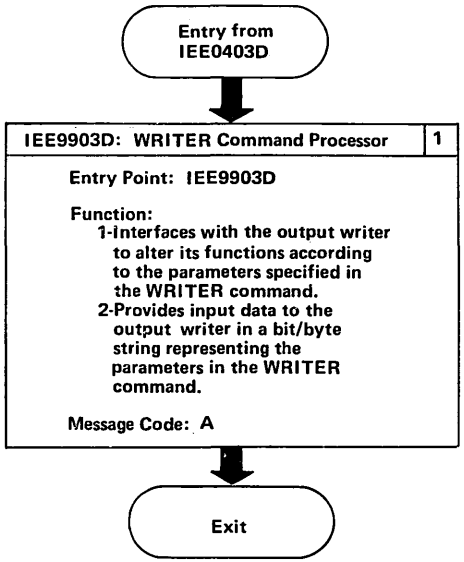
Figure 3-103. WRITELOG Command with CLOSE





Messages Code	Number	Reason
A	IEE007I	Delimiter error
	IEE011I	Parameter missing
	IEE019I	Quote(s) missing
	IEE026I	Log not supported
	IEE032I	WRITELOG pending
	IEE033I	Command issued
	IEE034I	Execution deferred
B	IEE044I	Abend completion code
	IEE037I	Log not supported
C	IEE014I	I/O error on SYS1.SYSPOOL
D	IEE043I	SYSOUT class
	IEE045I	Log inactive
	IEE010I	Log closed

Figure 3-104. WRITER Command



Message Code	Number	Reason
A	IEE298I	Invalid character
	IEE305I	Command invalid
	IEE306I	Command invalid
	IEE307I	Delimiter error
	IEE308I	Length error
	IEE309I	Unidentifiable keyword
	IEE310I	Keyword missing
	IEE312I	Parameters conflict
	IEE341I	Task not active
	IEE342I	Rejected task busy

Figure 3-105. Message Assembly Module

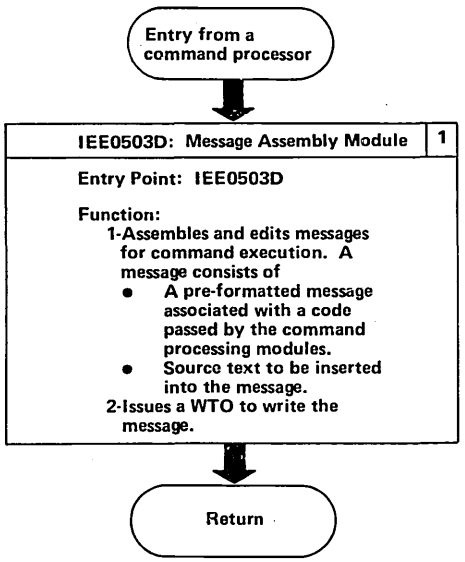


Figure 3-106. Queue Alter Routines

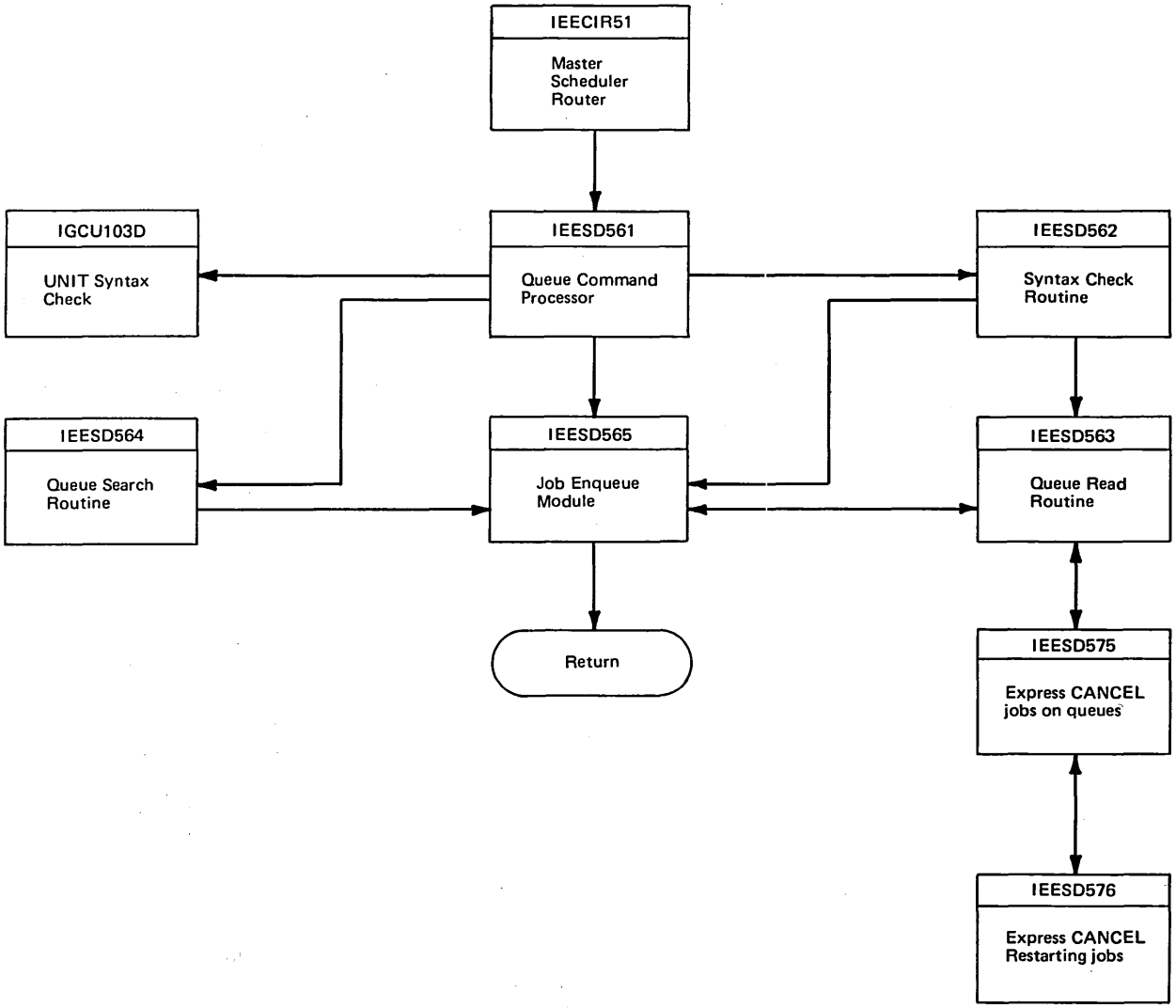


Figure 3-107. System Log Interconnection Module Diagram

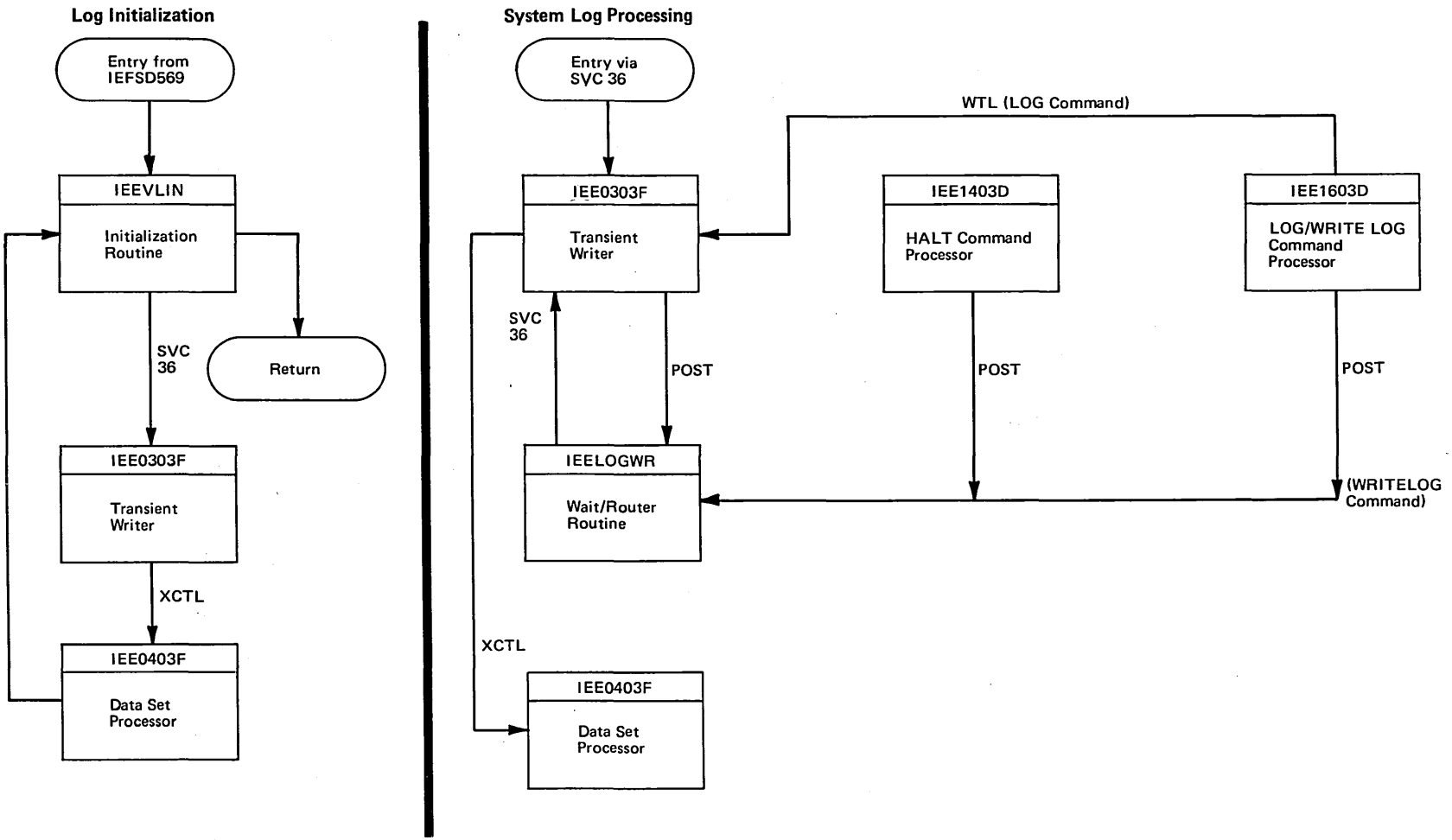


Figure 3-108. Writing to the system Log Using the WTL Macro

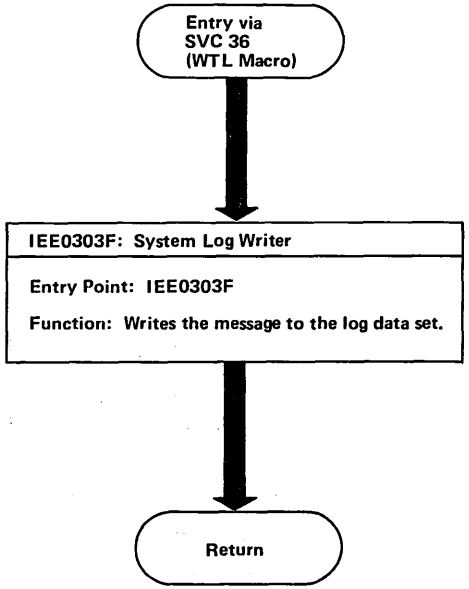
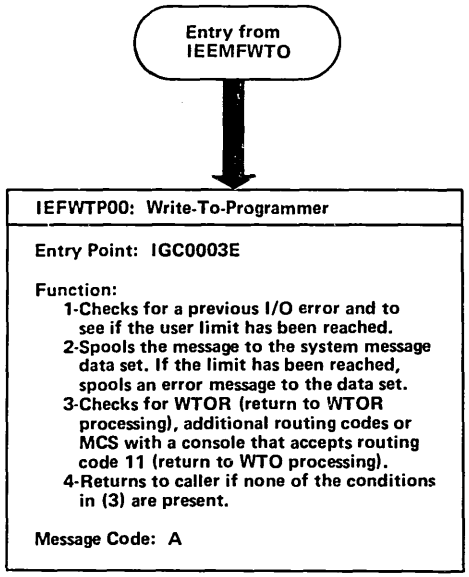
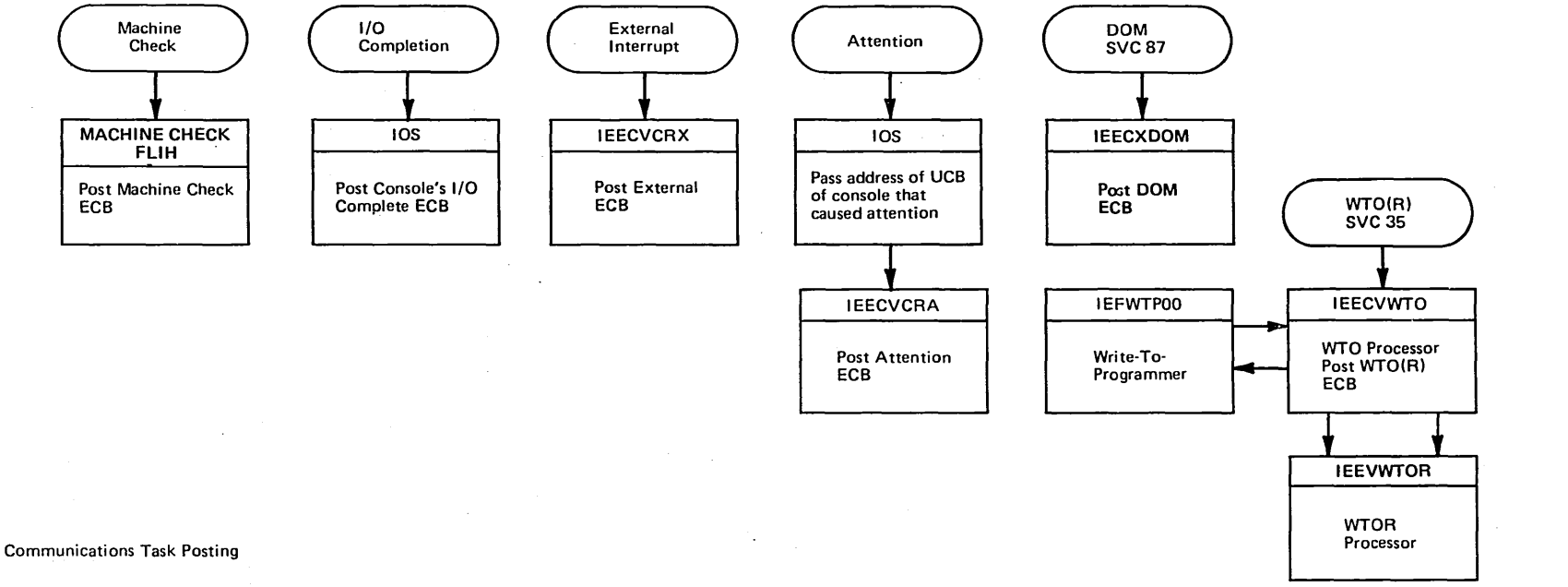


Figure 3-109. Write-to-Programmer

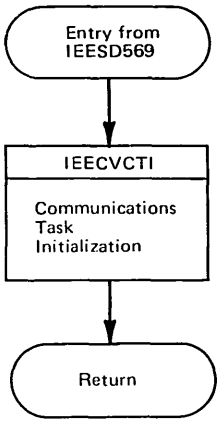


Message Code	Number	Reason
A	IEF094I IEF095I	User limit reached SYS1.SYSPOOL

Figure 3-110. COMTASK Initialization and Posting



Communications Task Posting



Communications Task Initialization

Figure 3-111. COMTASK Interconnection Module Diagram

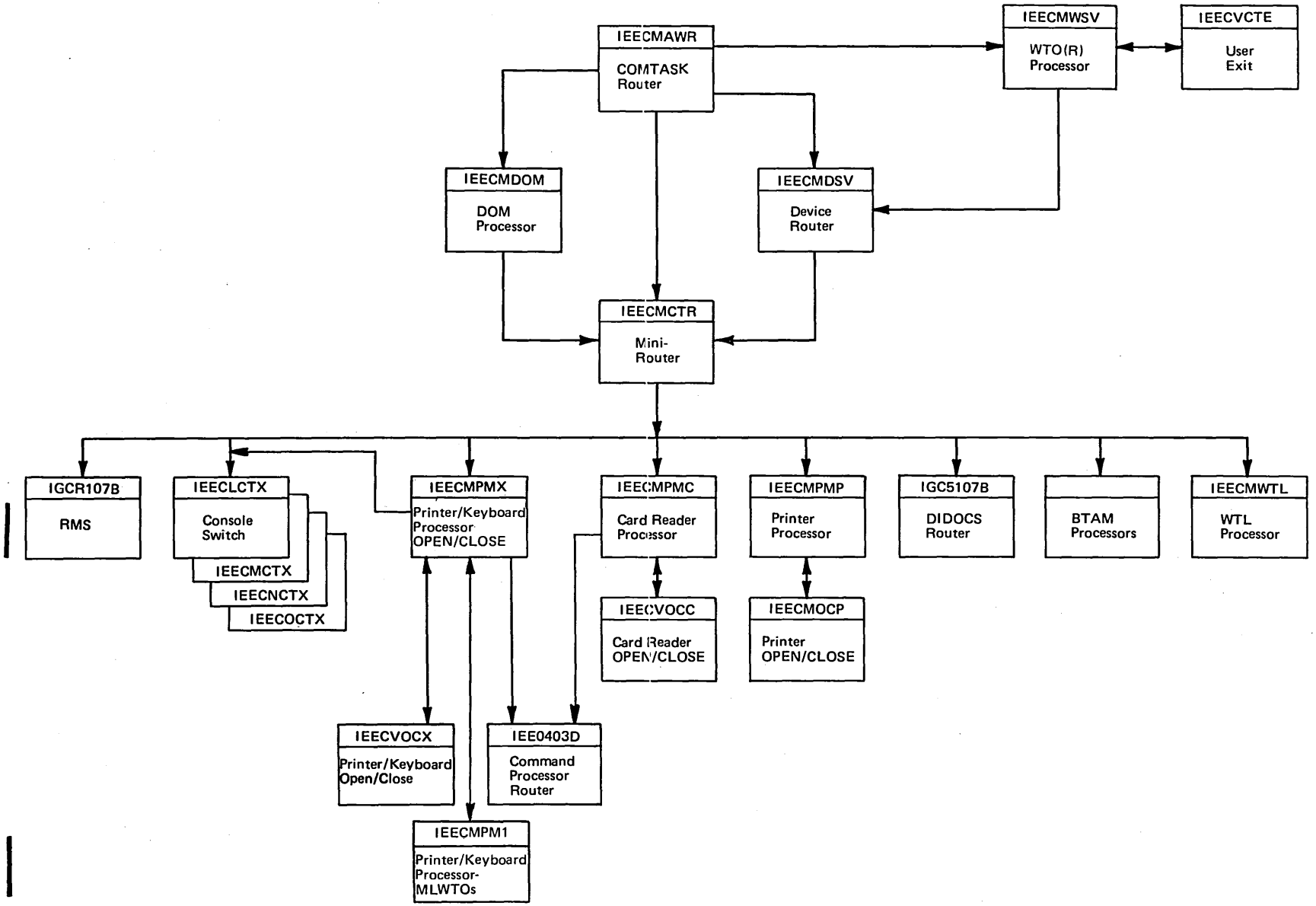


Figure 3-112. COMTASK Initialization (Part 1 of 2)

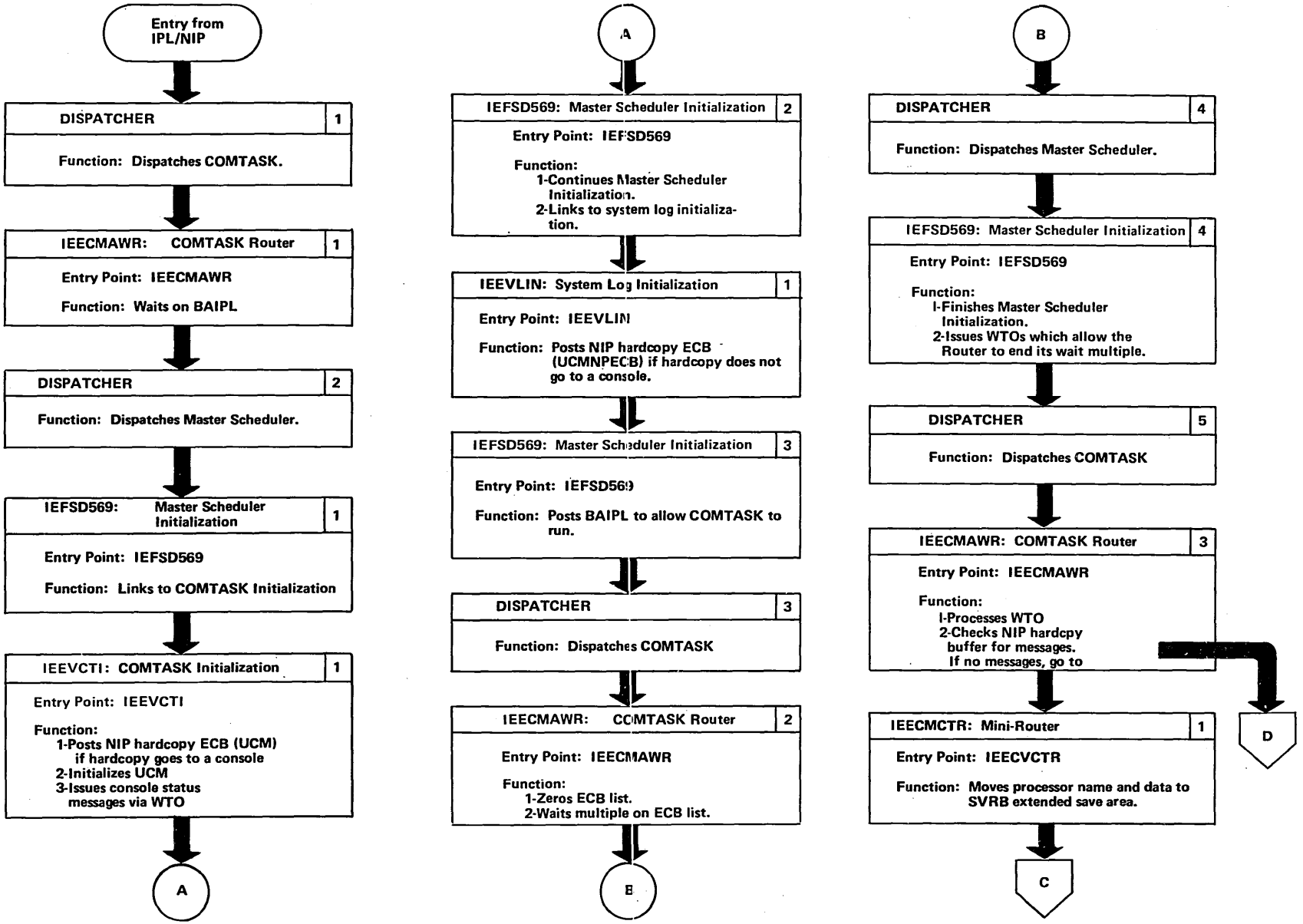


Figure 3-112. Comtask Initialization (Part 2 of 2)

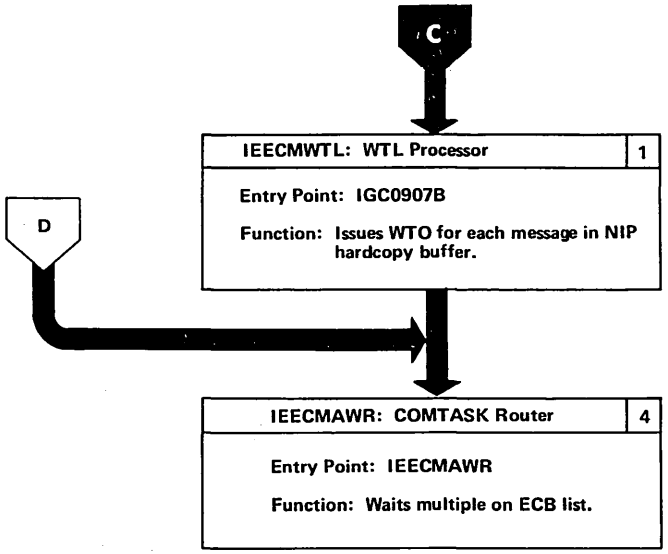
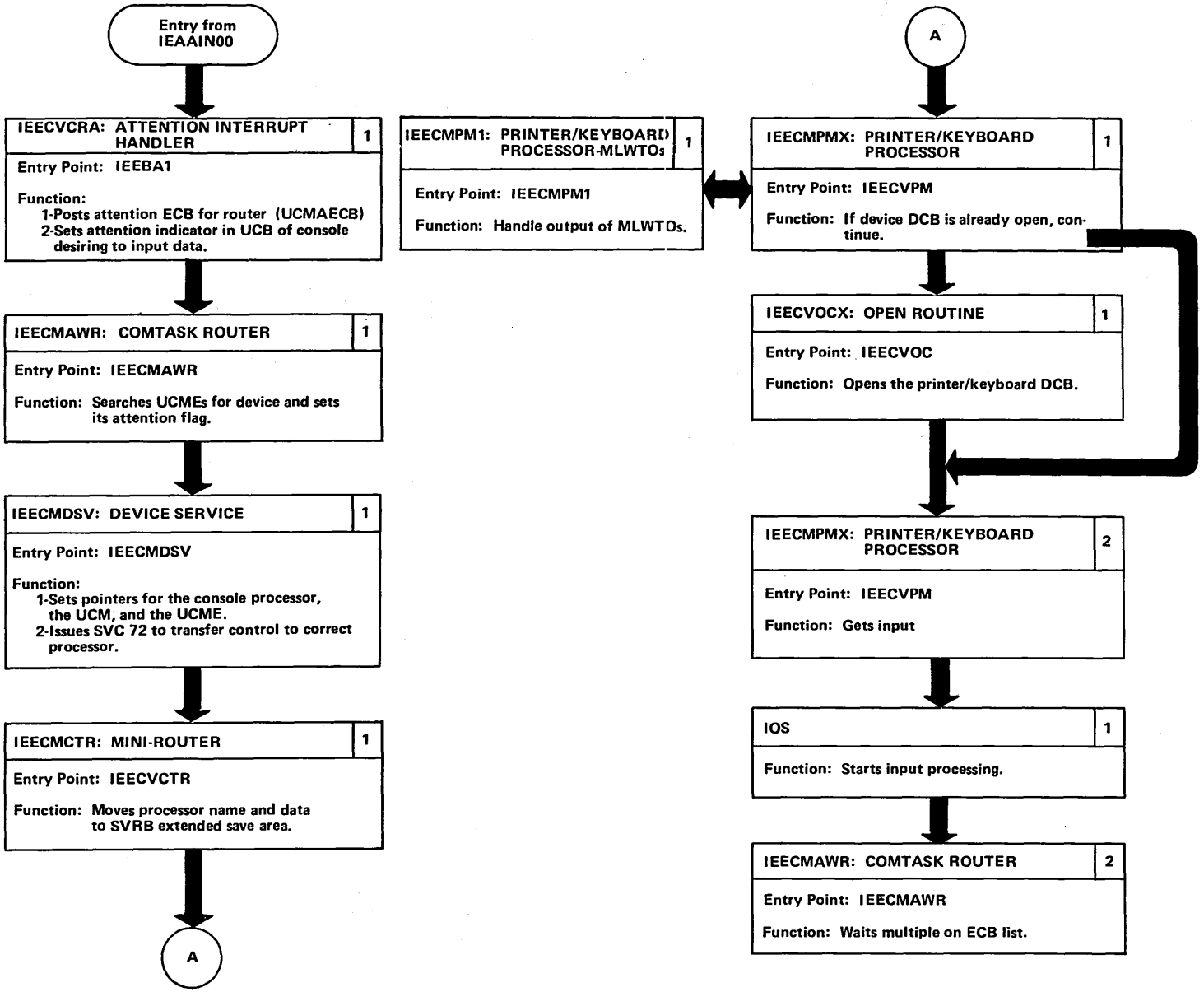


Figure 3-113. Printer/Keyboard Attention Processing



Printer 3-114. External Interrupt Processing

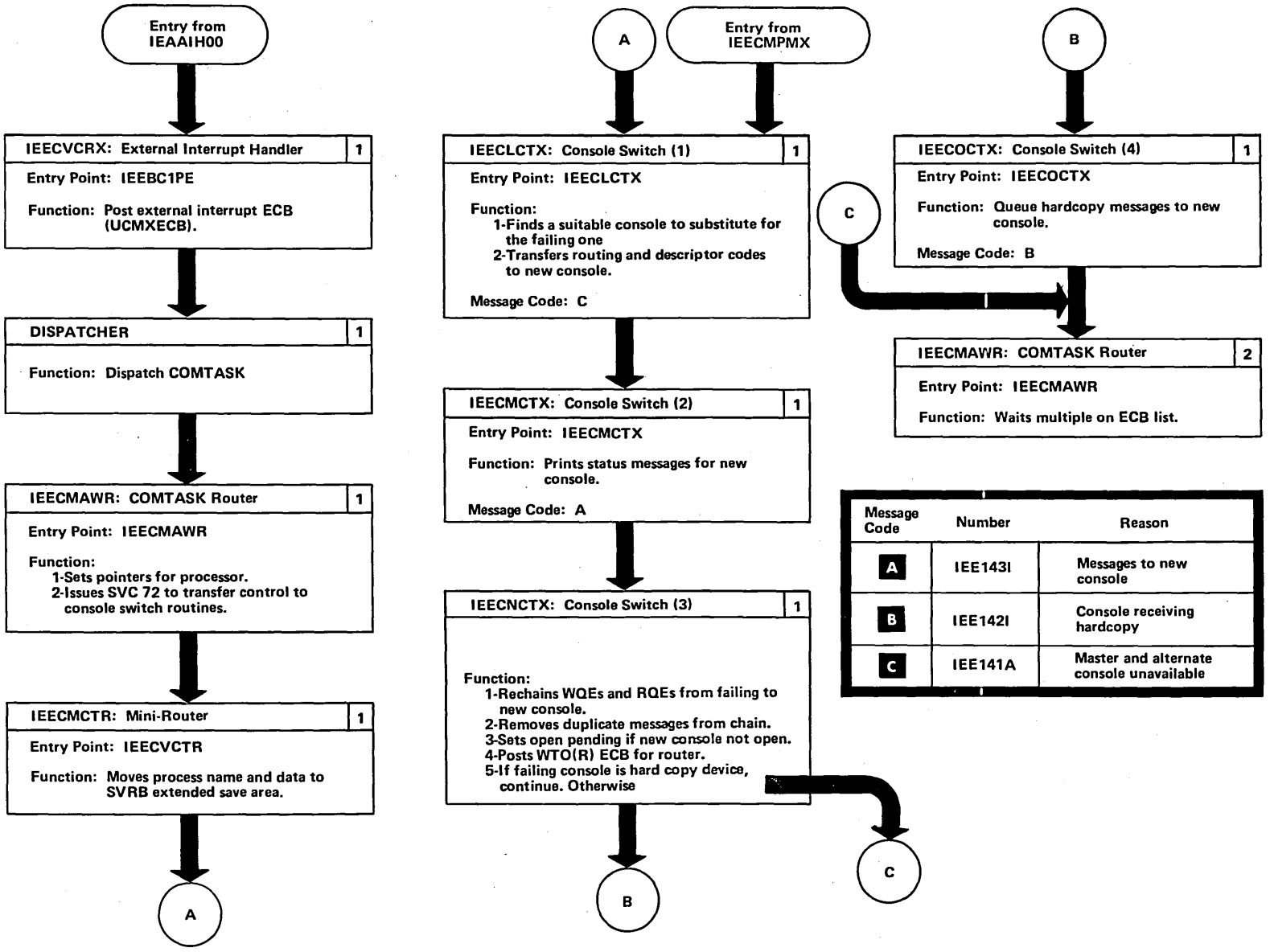
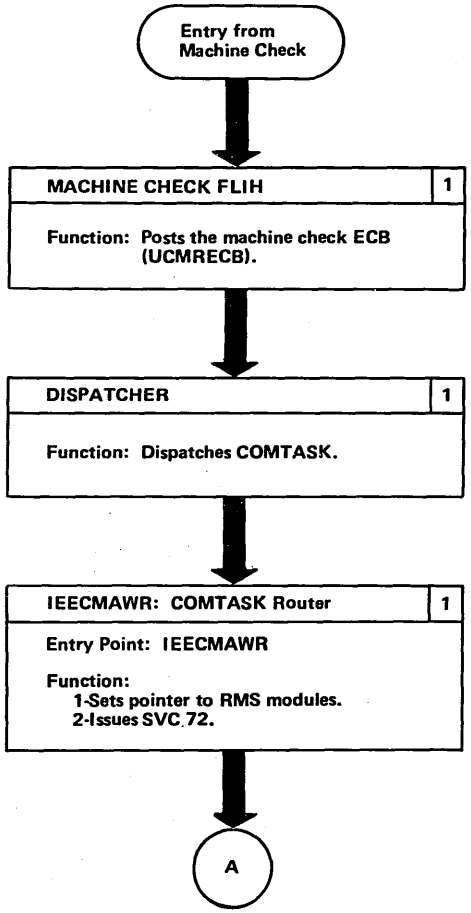


Figure 3-115. Machine Check Processing



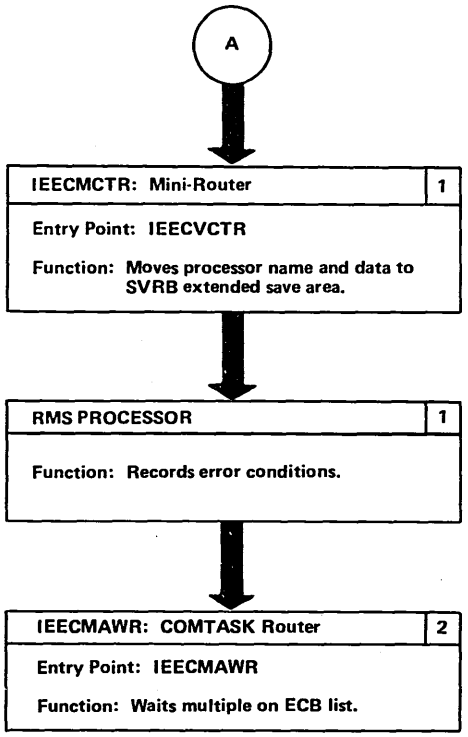


Figure 3-116. WTO(R) Processing (Part 1 of 2)

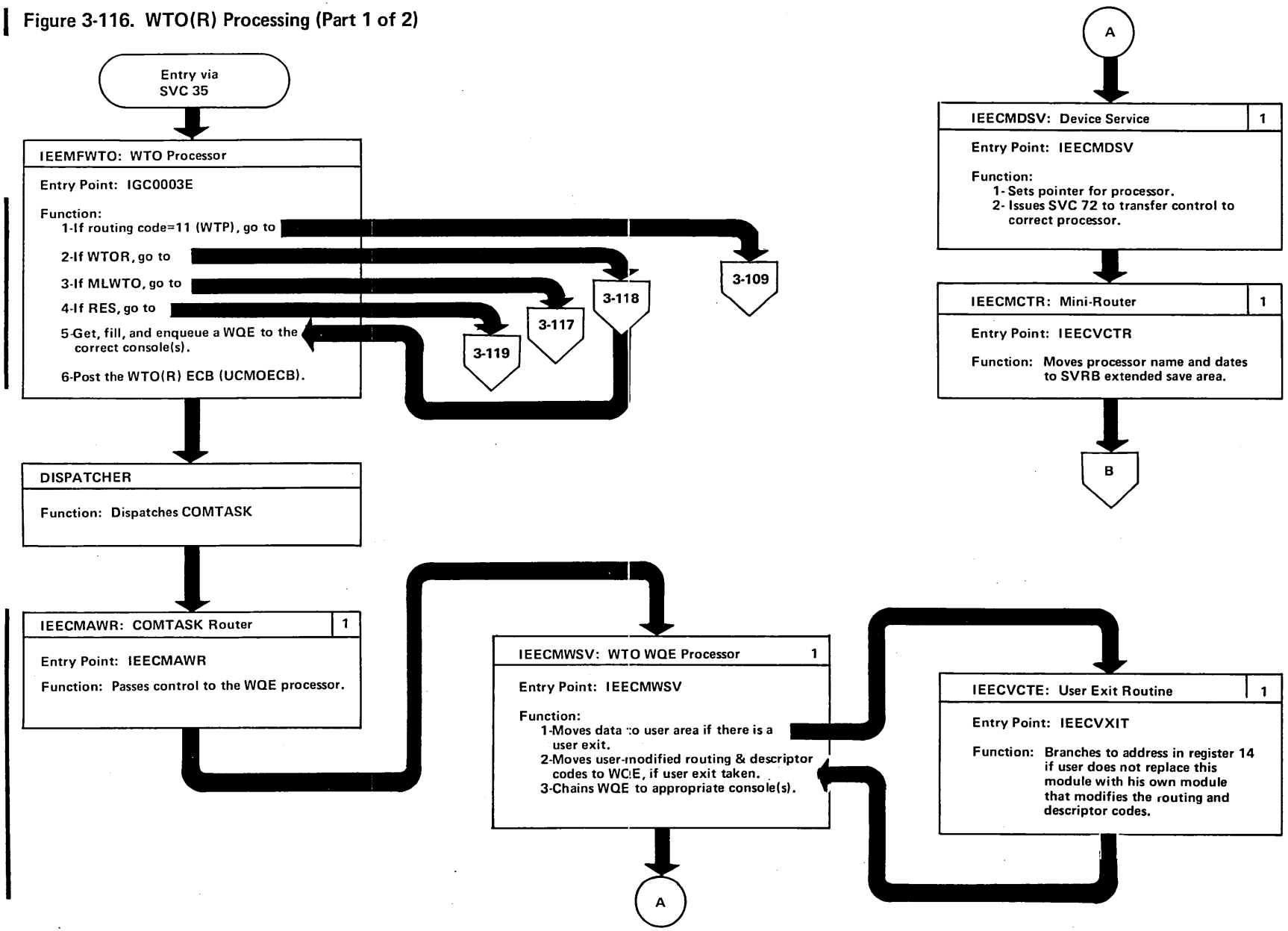


Figure 3-116. WTO(R) Processing (Part 2 of 2)

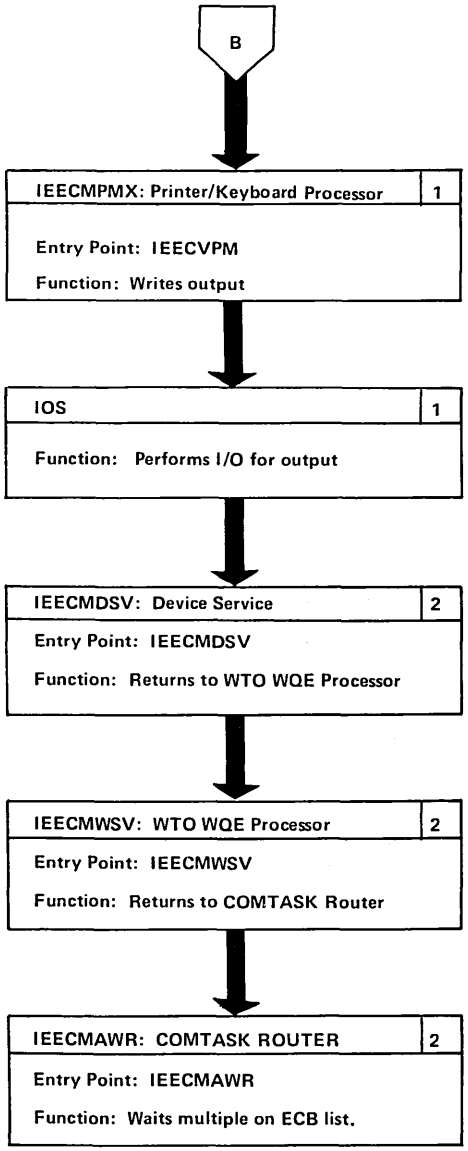


Figure 3-117. MLWTO Processing.

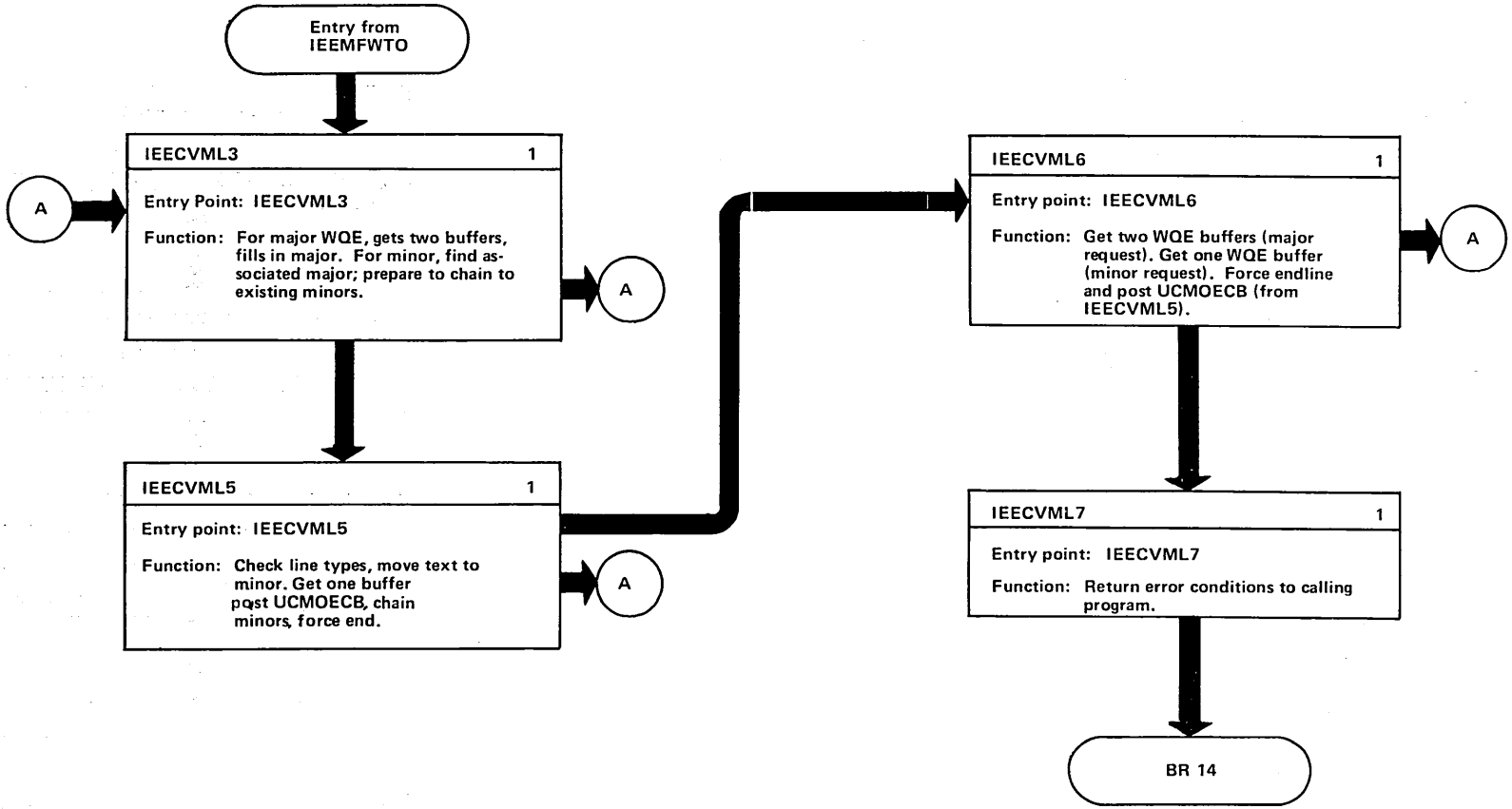


Figure 3-118. WTOR Processing

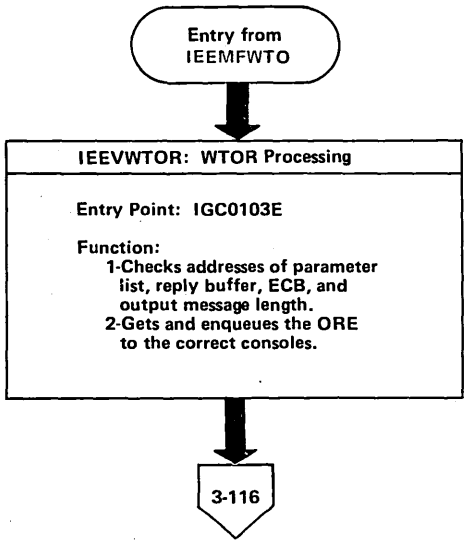


Figure 3-119. RES Processing for WTOR

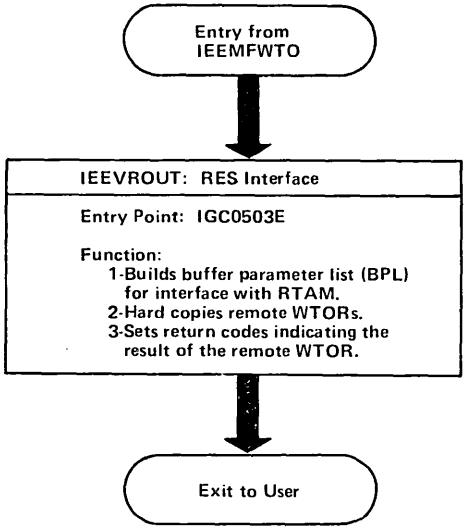


Figure 3-120. I/O Completion for Input

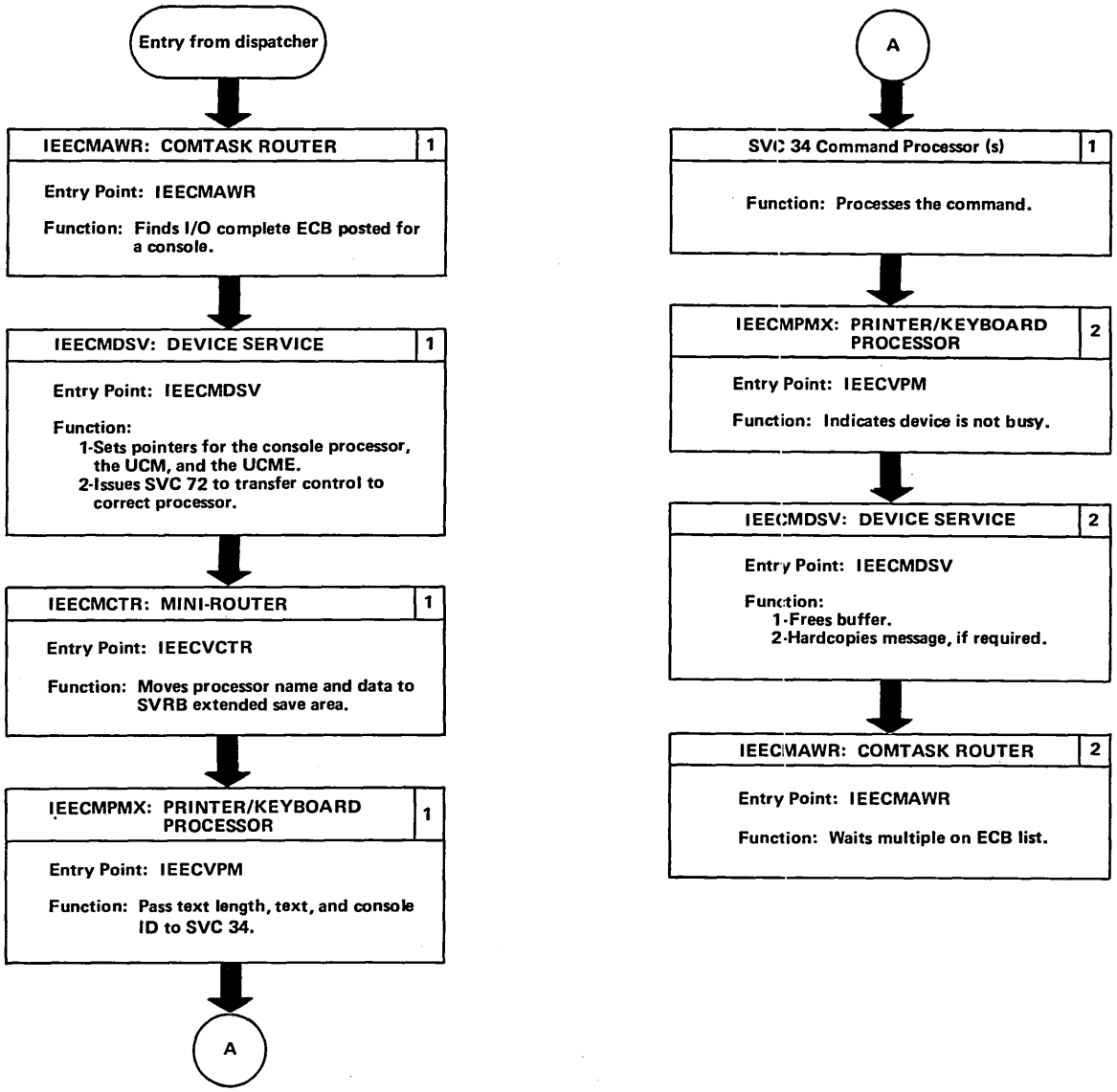


Figure 3-121. I/O Completion for Output

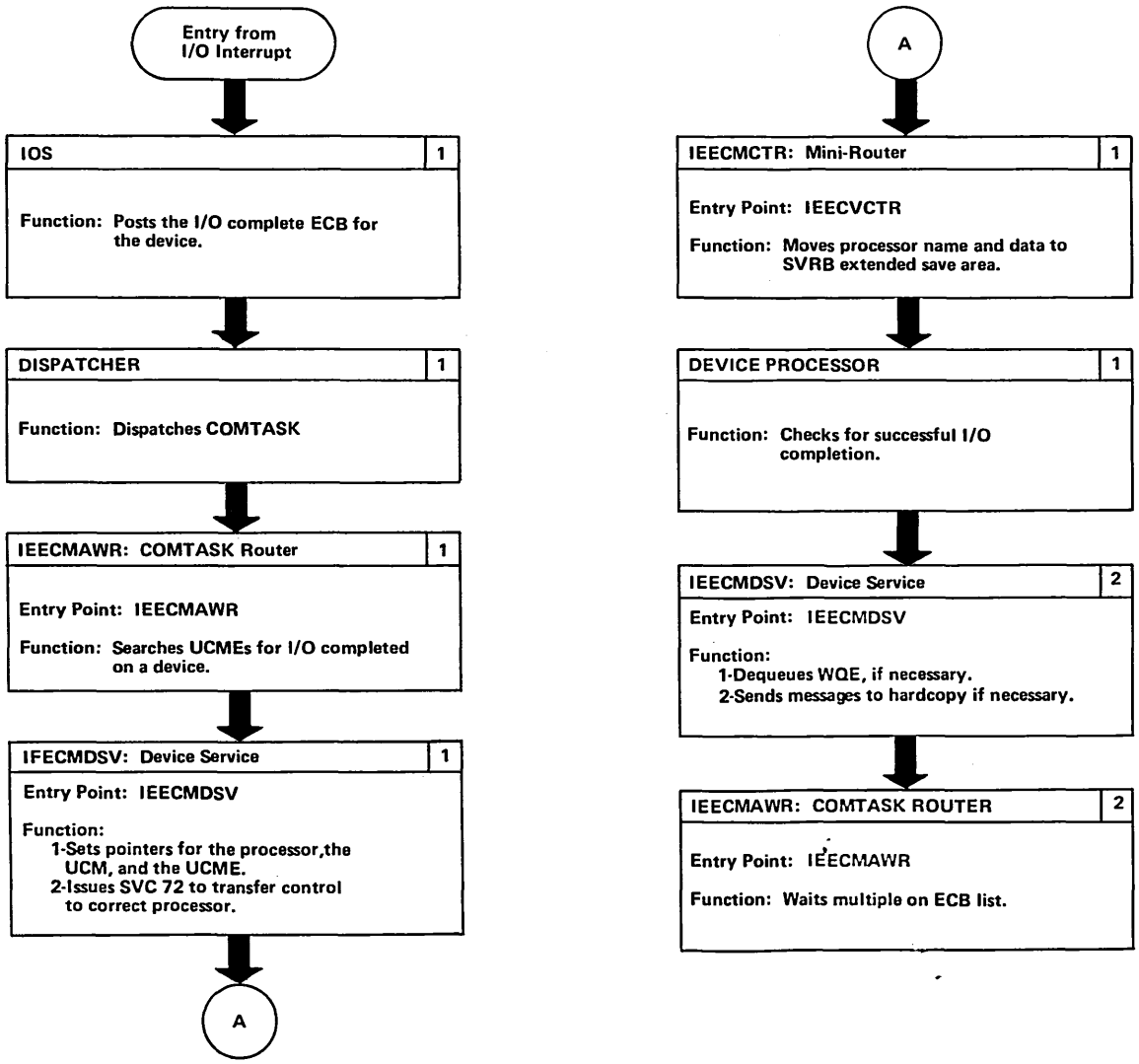
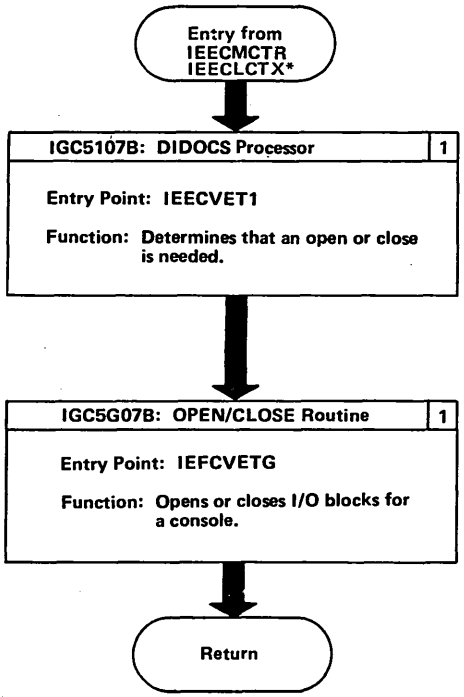


Figure 3-122. DIDOCS OPEN/CLOSE Processing



*For MCS console switch processing

Figure 3-123. ROLL Mode Processing (Part 1 of 2)

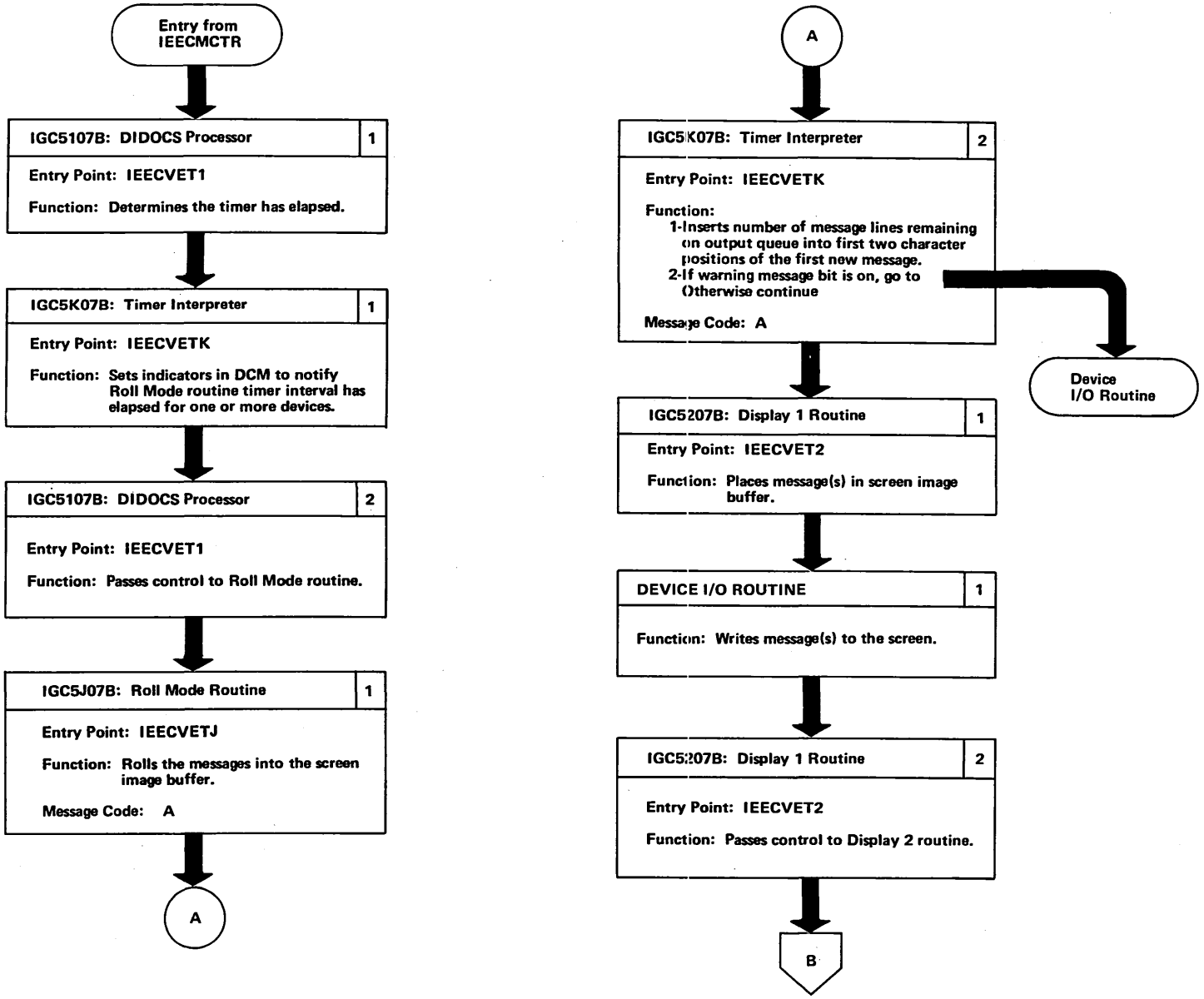
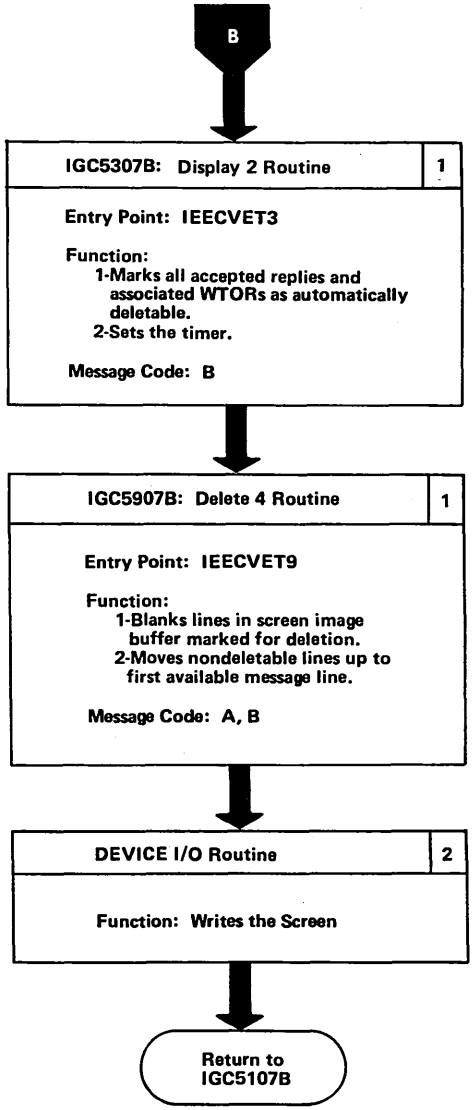
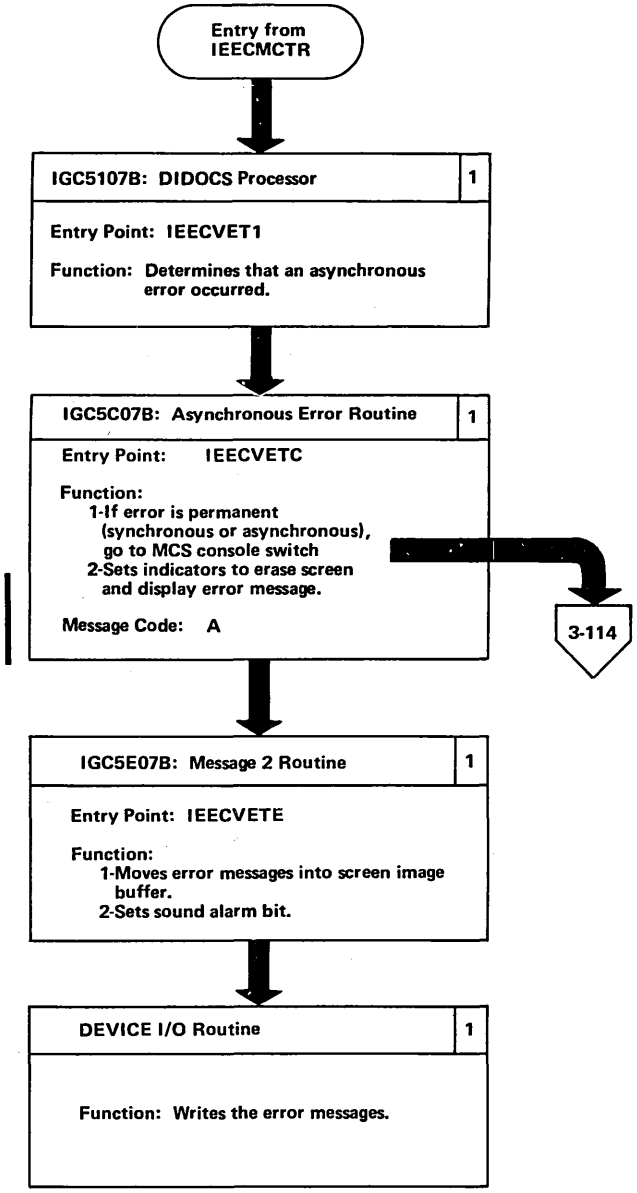


Figure 3-123. ROLL Mode Processing (Part 2 of 2)



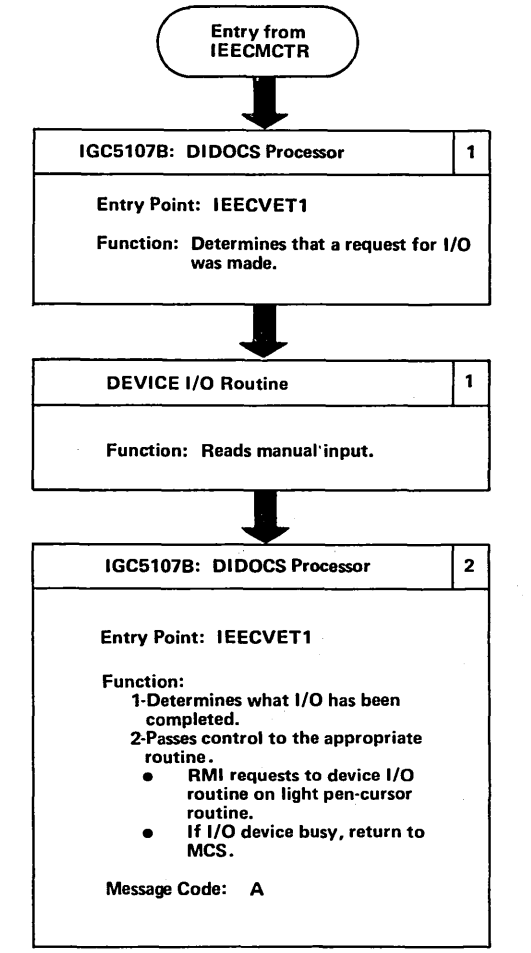
Messages Code	Number	Reason
A	IEE159E	Messages waiting
B	IEE160I	Screen messages replaced by status display

Figure 3-124. DIDOCS Asynchronous Error Processing



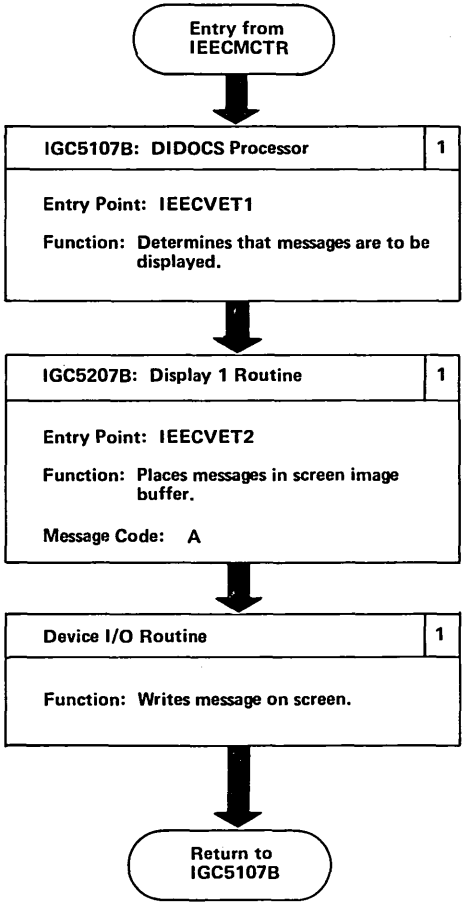
Message Code	Number	Reason
A	IEE170E IEE171I	Recoverable hardware error Hardware error

Figure 3-125. DIDOCS Attention Processing



Message Code	Number	Reason
A	IEE153E	Command too long

Figure 3-126. Displaying System Messages (DIDOCS)



Message Code	Number	Reason
A	IEE159E IEE160I	Messages waiting Screen messages displaced by status display

Figure 3-127. DIDOCS CANCEL Processing

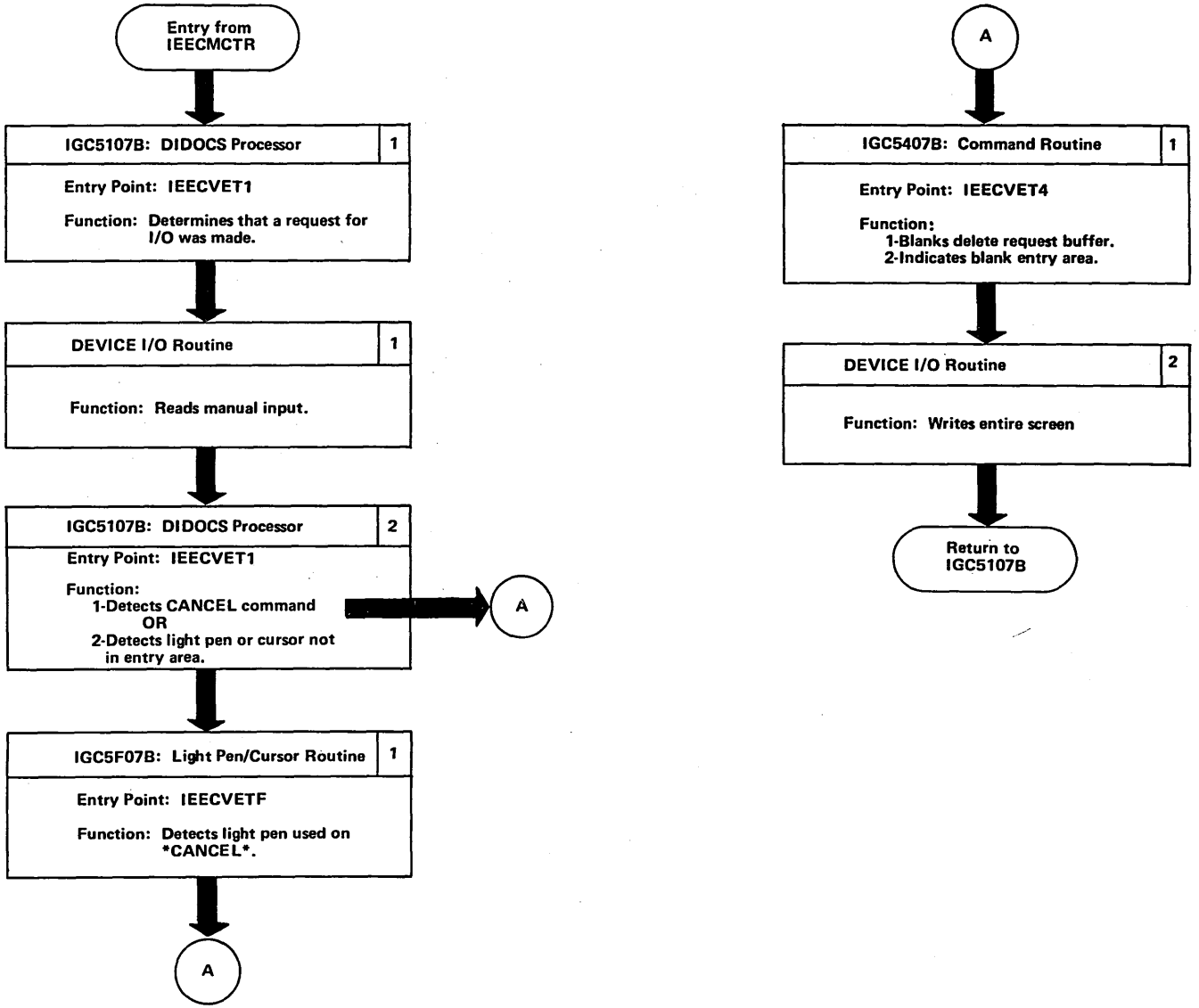


Figure 3-128. Displaying the CONTROL Command Operands (Part 1 of 2)

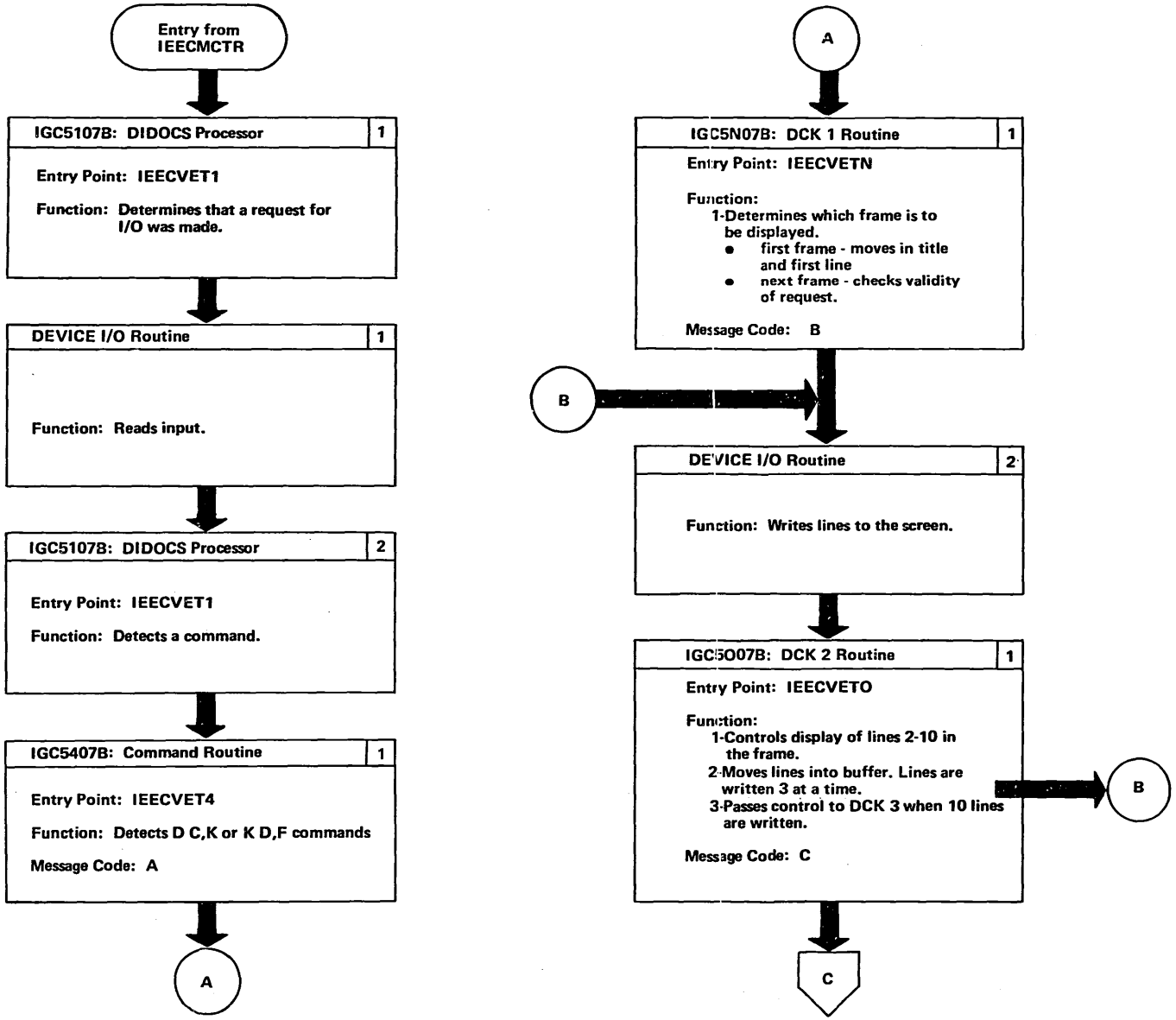
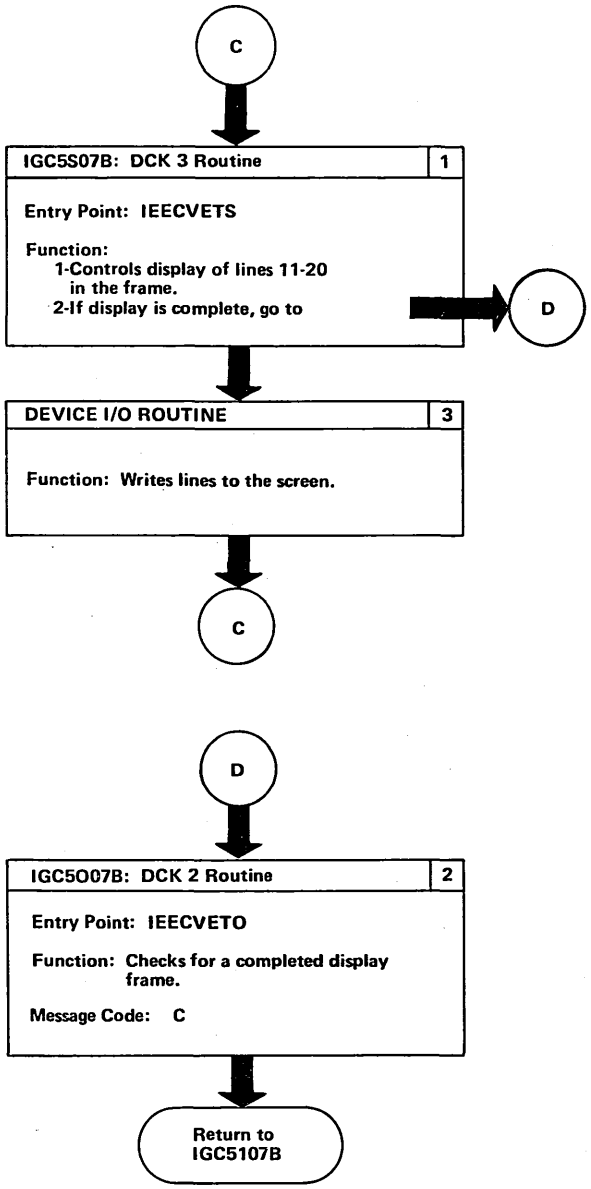
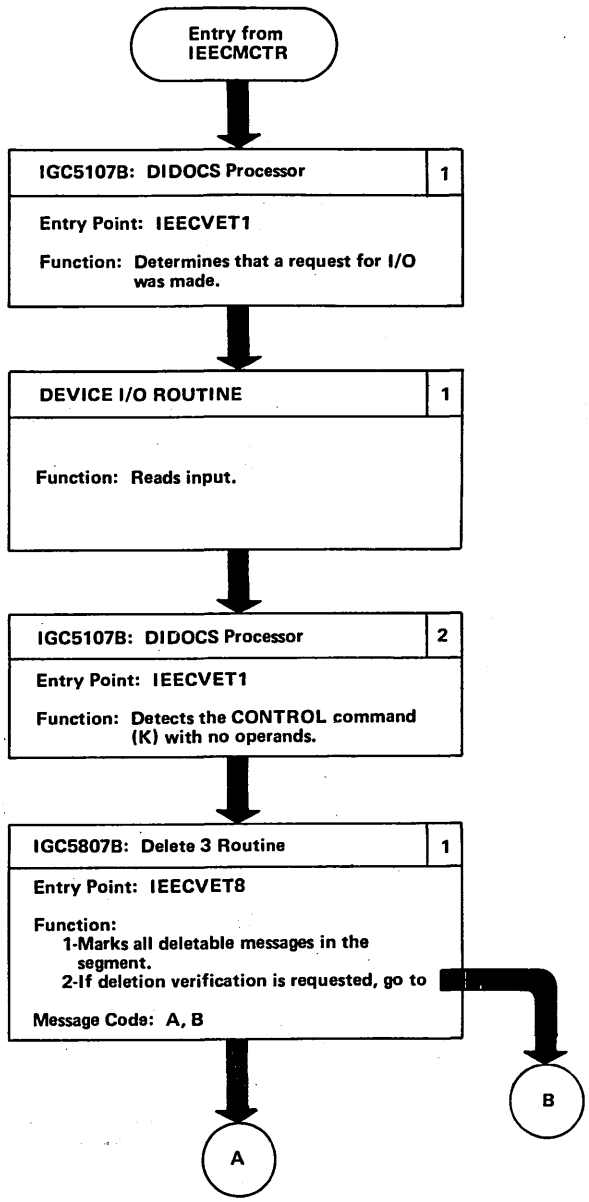


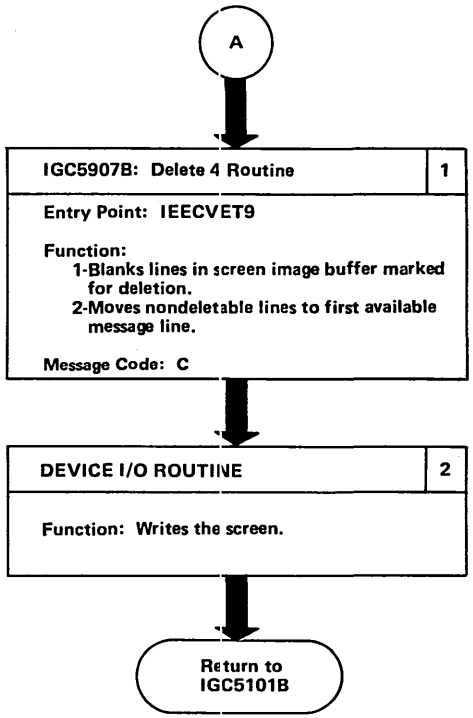
Figure 3-128. Displaying the CONTROL Command Operands (Part 2 of 2)

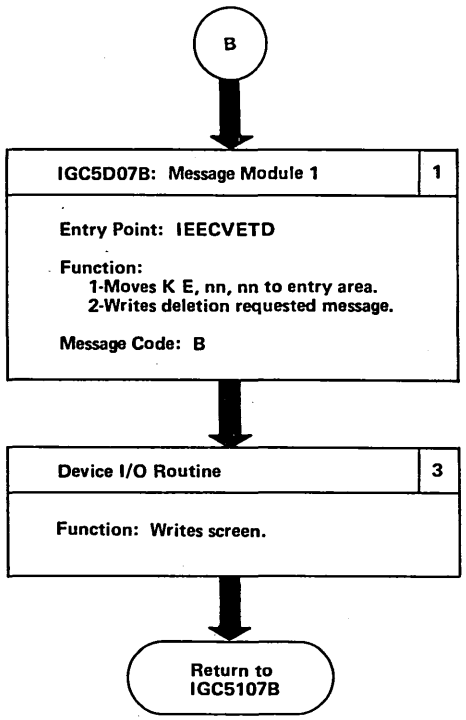


Messages Code	Number	Reason
A	IEE156E	Invalid command
B	IEE151E	Invalid message deletion request
C	IEE160I	Screen messages displaced by status display

Figure 3-129. CONTROL Command Processing with No Operands







Messages Code	Number	Reason
A	IEE151E	Invalid message deletion request
B	IEE157E	Message deletion required
C	IEE159I IEE160I	Messages waiting Screen messages displaced by status display

Figure 3-130. Delete Operator Message (DOM) Processing (Part 1 of 2)

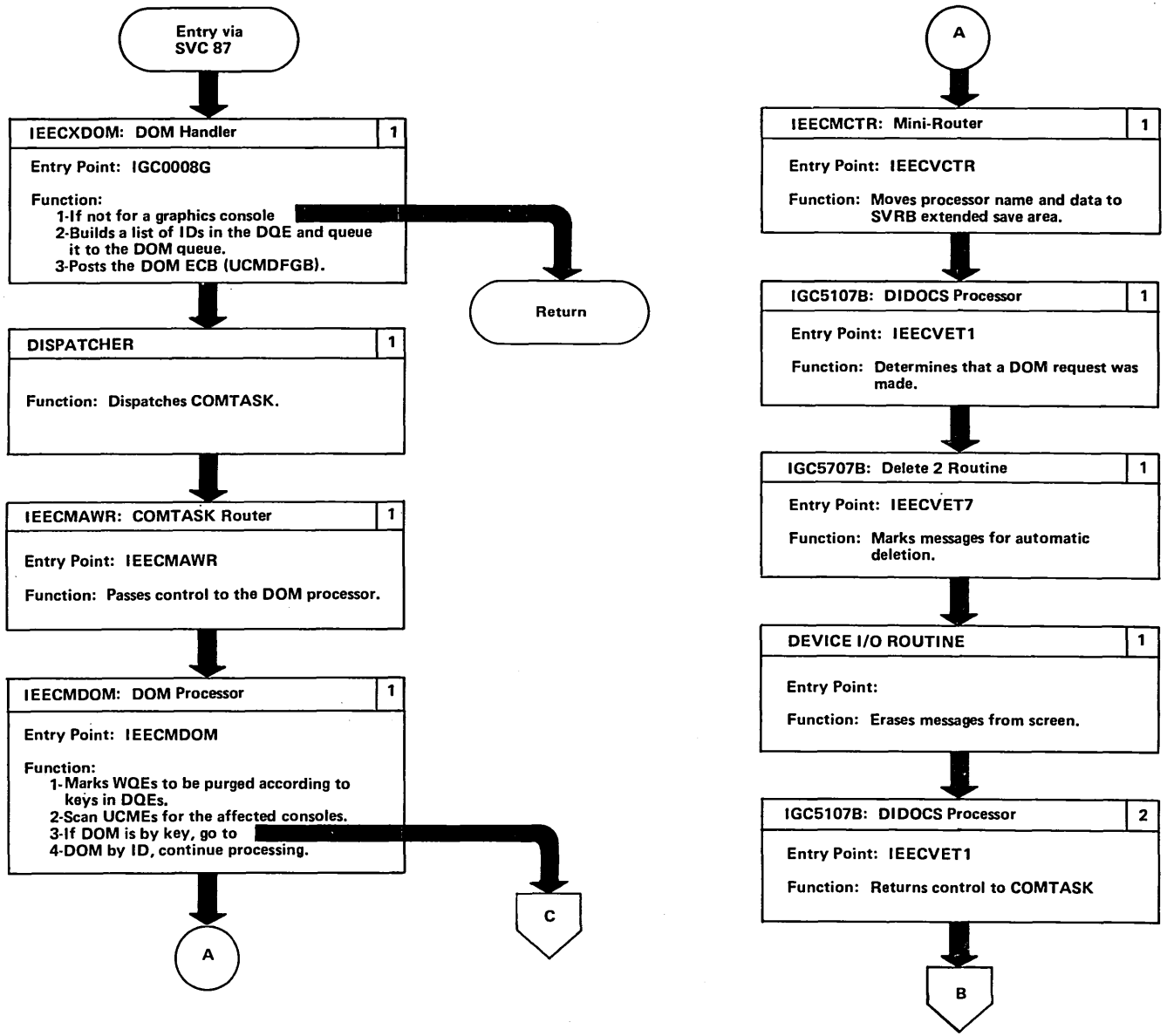


Figure 3-130. Delete Operator Message (DOM) Processing (Part 2 of 2)

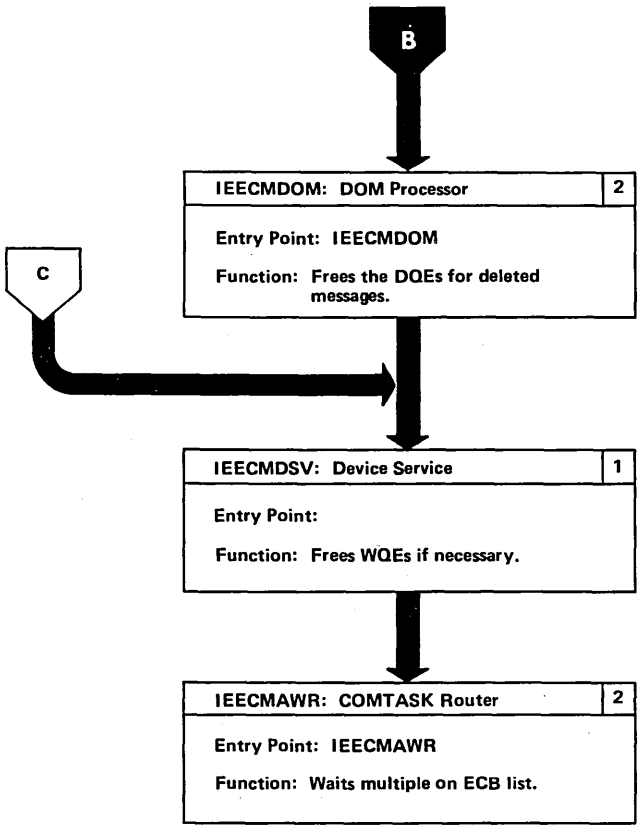


Figure 3-131. 2250 I/O Processing

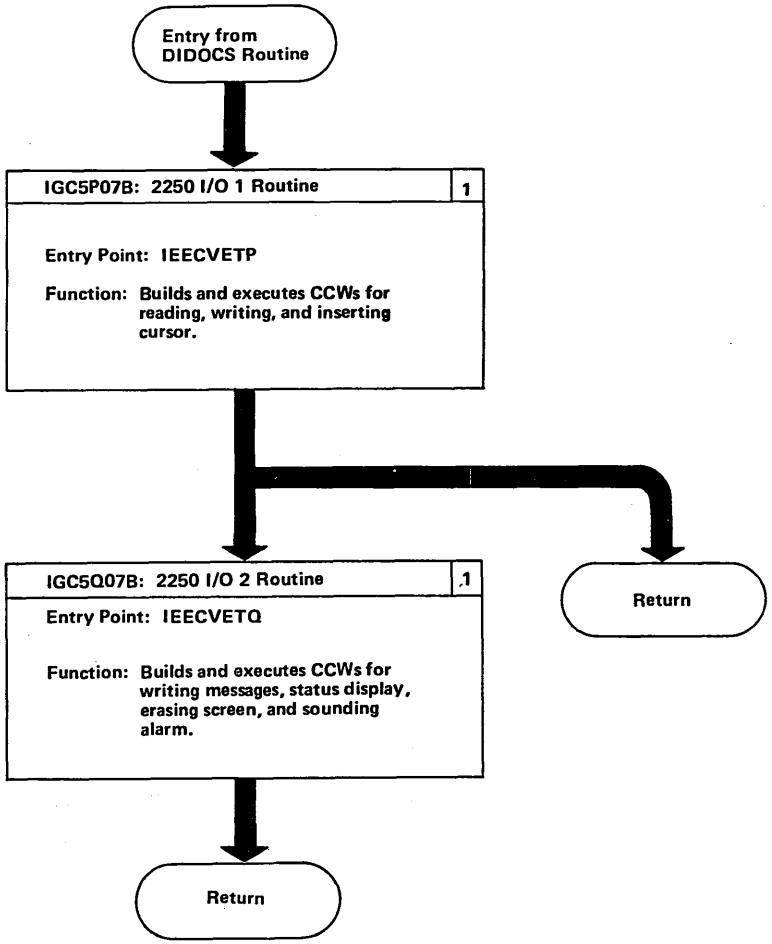


Figure 3-132. 2260 I/O Processing

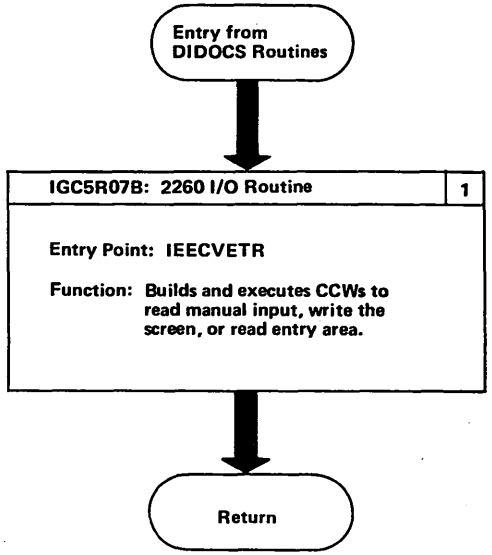


Figure 3-133. SMF Interconnection Module Diagram

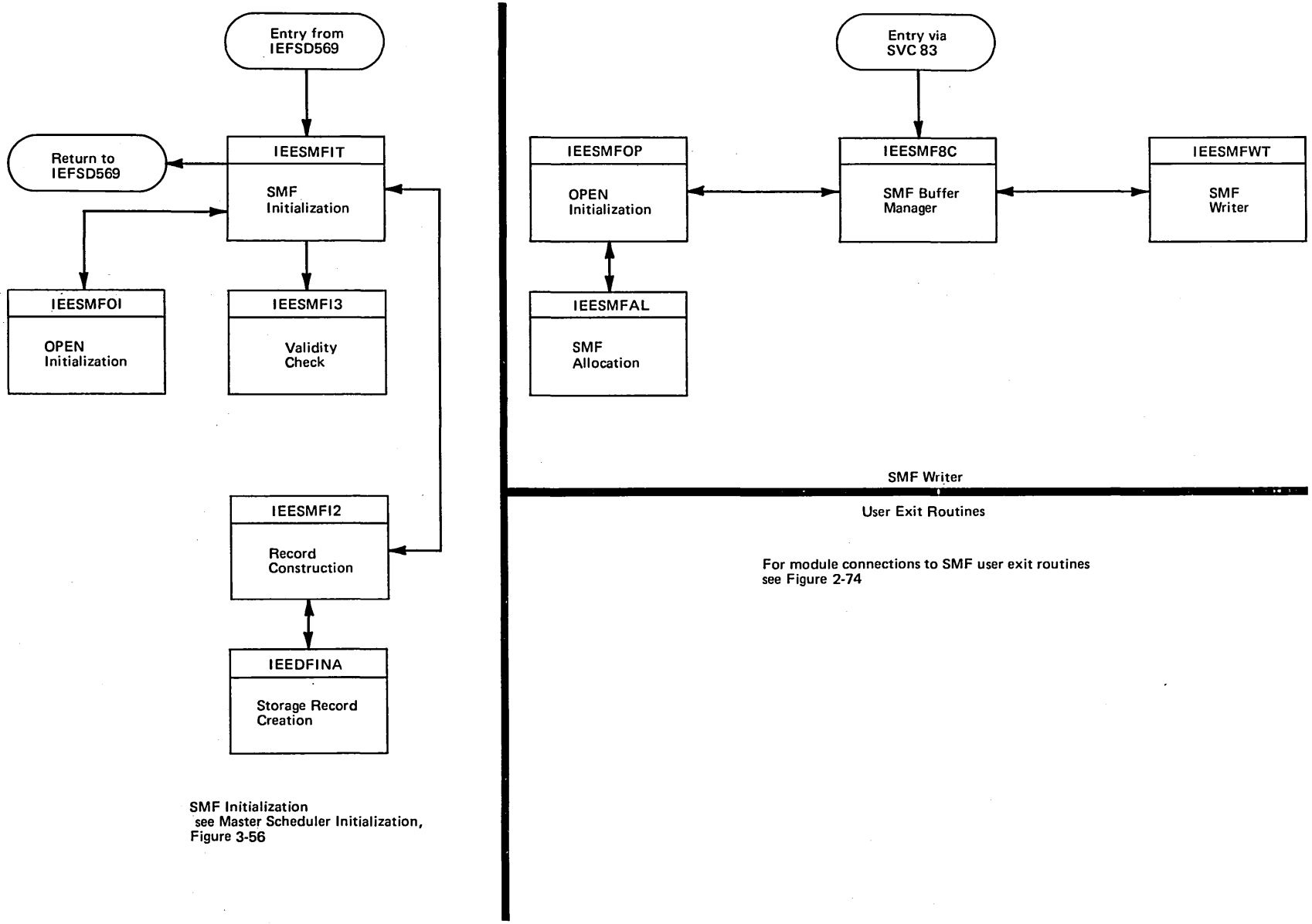
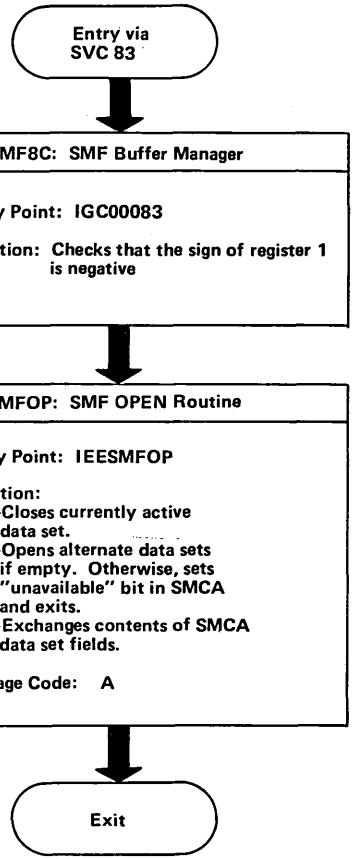


Figure 3-134. Switching SMF Data Sets



Message Code	Number	Reason
A	IEE360I IEE362A	Alternate data set open Dump current data set

Figure 3-135. Writing Non-segmented SMF Records

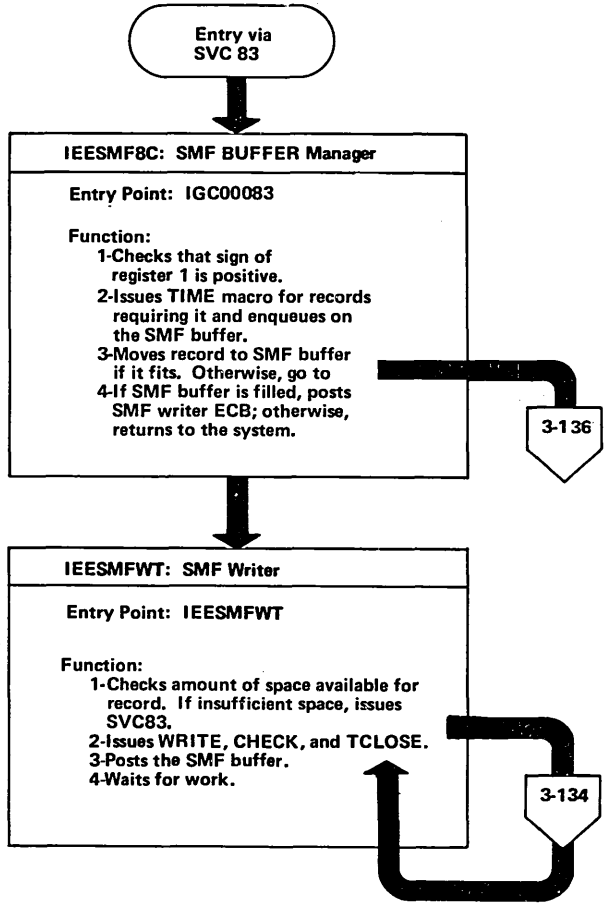
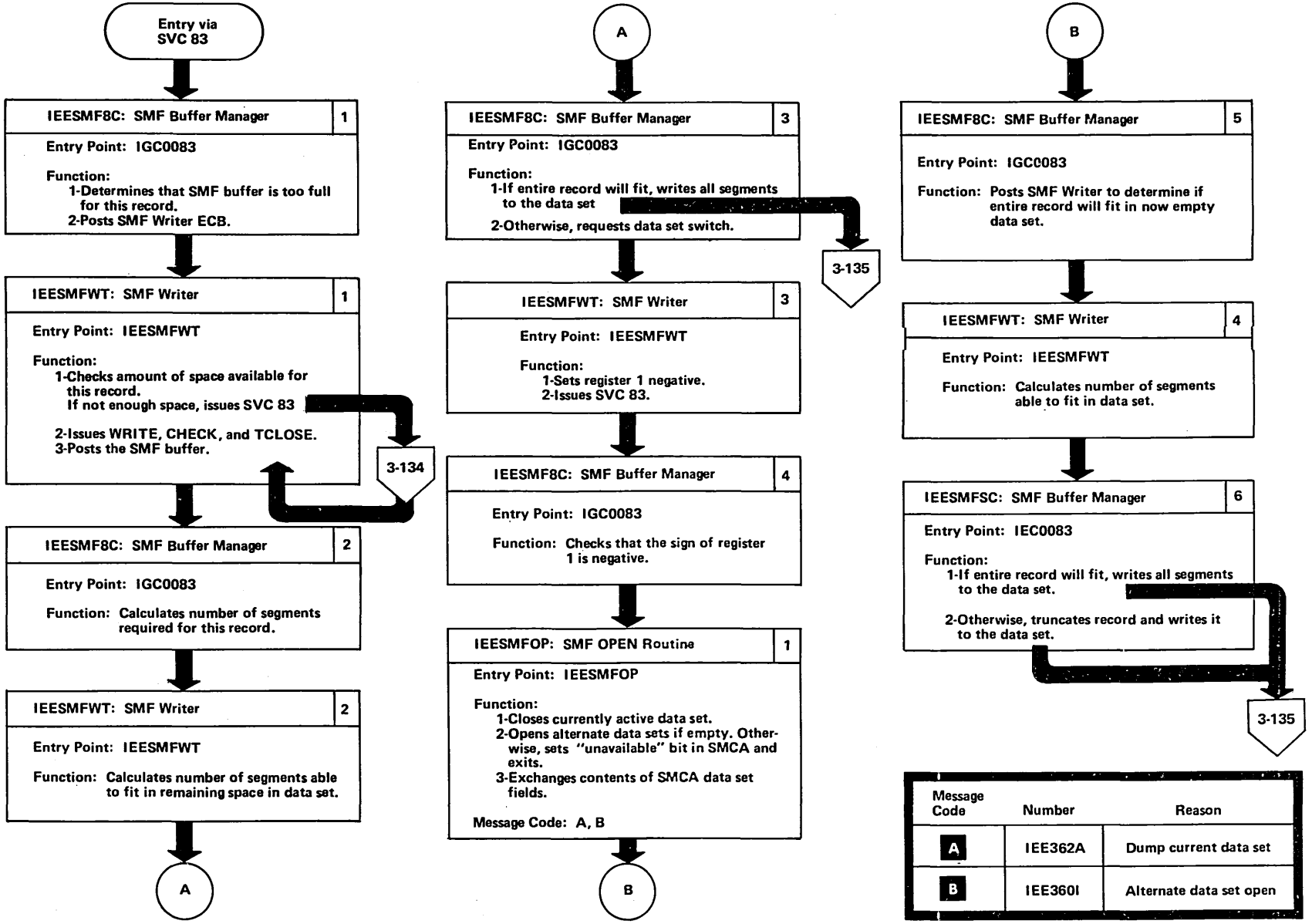
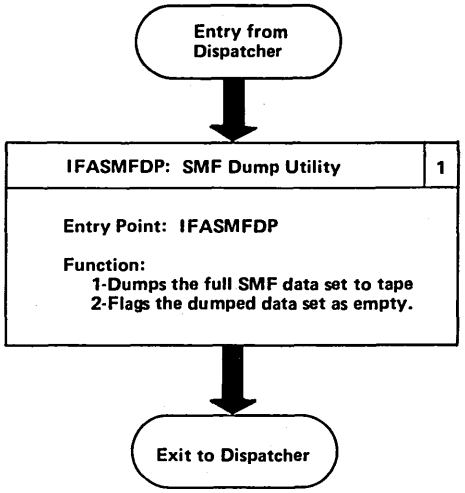


Figure 3-136. Writing Segmented SMF Records



Message Code	Number	Reason
A	IEE362A	Dump current data set
B	IEE360I	Alternate data set open

Figure 3-137. Dumping the SMF Data Set



Section 4. Cross-Reference Directories

This section contains four directories, each organized alphabetically. The directories are arranged according to module, entry point, load module, and module name with figure number reference. The alphabetical list by module also contains the microfiche name when the microfiche name differs from the module name.

CONTENTS

Cross-Reference Directory Alphabetical List

Cross-Reference Directory Alphabetical List — Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IECURAT1	IECURAT1	IEANUCnn	IECURAT1	
IEECIR50	IEECIR50	IEANUCnn	IEECIR50	
IEECIR51	IEECIR51	IEANUCnn	IEECIR51	
IEECMOCP	IEECVOC	IGC2107B	IEECVOC	
IEECMPMC	IEECVPM	IGC1107B	IEECVPM	
IEECMPMP	IEECVPM	IGC2107B	IEECVPM	
IEECMPMX	IEECVPM	IGC0107B	IEECVPM	
IEECMPM1	IEECMPM1	IGC0207B	IEECMPM1	
IEECMWSV	IEECMWA0	IEECMWSV	IEECMWSV	
-	IEECMWSV	-	-	
-	IEECMQCN	-	-	
-	IEECMENQ	-	-	
-	IEECMWA1	-	-	
IEECMWTL	IGC0907B	IGC0907B	IGC0907B	
IEECNCTX	IEECNCTX	IGCXN07B	IEECNCTX	
IEECOCTX	IEECOCTX	IGCXO07B	IEECOCTX	
IEECVCRA	IEEBA1A1	IEEBA1	IEEBA1	
-	IEEBA1A0	-	-	
-	IEEBA1	-	-	
IEECVCRX	IEECJR45	IEEBC1PE	IEEBC1PE	
-	IEEBC1A1	-	-	
-	IEEBC1A0	-	-	
-	IEEBC1PE	-	-	
IEECVCTE	IEECVXIT	IEECVXIT	IEECVXIT	
IEECVDOM	IEECMDOM	IEECMDOM	IEECMDOM	
IEECVML3	IEECVML3	IGC0603E	IEECVML3	
IEECVML5	IEECVML5	IGC0703E	IEECVML5	
IEECVML6	IEECVML6	IGC0803E	IEECVML6	
IEECVML7	IEECVML7	IGC0903E	IEECVML7	
IEECVOCC	IEECVOC	IGC1107B	IEECVOC	
IEECVOCX	IEECVOC	IGC0107B	IEECVOC	
IEECXDOM	IGC0008G	IGC0008G	IGC0008G	
IEEDFINA	IEEDFINA	IGX00003	IEEDFINA	
IEEDFIN1	IEEDFIN1	IGX00000	IEEDFIN1	
IEEDFIN2	IEEDFIN2	IGX01000	IEEDFIN2	
IEEDFIN3	IEEDFIN3	IGX02000	IEEDFIN3	
IEEDFIN4	IEEDFIN4	IGX00001	IEEDFIN4	
IEEDFIN5	IEEDFIN5	IGX00002	IEEDFIN5	
IEEDFIN6	IEEDFIN6	IGX03000	IEEDFIN6	
IEEDFIN7	IEEDFIN7	IGX04000	IEEDFIN7	
IEEDFIN8	IEEDFIN8	IEEDFIN8	IEEDFIN8	
-	-	-	IEEDFIN1	
-	-	-	IEEDFIN2	
-	-	-	IEEDFIN6	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEELGON	IEELGON	IEELGON	IEELGON	
IEELGON1	IEELGON1	IEELGON1	IEELGON1	
IEELGON2	IEELGON2	IEELGON2	IEELGON2	
IEELIST	IEELIST	IEELIST	-	
IEELIST1	IEELIST1	IEELIST1	-	
IEELOGWR	IEELOGUP	IEELWAIT	IEELWAIT	
-	IEELOGFC	-	IEELWAIT	
-	IEELOGWR	-	IEELWAIT	
IEEMB800	IEEMB800	IEFSD569	IEEMB800	
IEEMFWTO	IGC0003E	IGC0003E	IGC0003E	
IEEPSN	IEEPSN	IEEVRCTL	IEEPSN	
IEERTE	IEERTE	IEERTE	IEERTE	
IEERTE1	IEERTE1	IEERTE1	IEERTE1	
IEERTE2	IEERTE2	IEERTE2	IEERTE2	
IEERTE3	IEERTE3	IEERTE3	IEERTE3	
IEESD561	IEESD561	IGX00004	IEESD561	
IEESD562	IEESD562	IGX01004	IEESD562	
IEESD563	IEESD563	IGX02004	IEESD563	
IEESD564	IEESD564	IGX03004	IEESD564	
IEESD565	IEESD565	IGX04004	IEESD565	
IEESD566	IEESD566	IGX00012	IEESD566	
IEESD568	IEEBAFQE	IEANUCnn	IEEMSER	
-	IEEMSLT	-	-	
IEESD571	IGC1803D	IGC1803D	IGC1803D	
IEESD575	IEESD575	IGX05004	IEESD575	
IEESD576	IEESD576	IGX06004	IEESD576	
IEESMFAL	IEESMFAL	IGC0208C	IEESMFAL	
IEESMFIT	IEESMFIT	IEESMFIT	IEESMFIT	
-	IEESMFI4	-	-	
IEESMFI2	IEESMFI2	IEESMFI2	IEESMFI2	
IEESMFI3	IEESMFMS	IEESMFI3	IEESMFI3	
-	IEESMFI3	-	-	
-	IEESMFIO	-	-	
IEESMFOI	IEESMFOI	IEESMFOI	IEESMFOI	
IEESMFOP	IEESMFOP	IGC0108C	IEESMFOP	
IEESMFWT	IEESMFWT	IEANUCNN	IEESMFWT	
-	IEESMFRB	-	-	
IEESMF8C	IGC0083	IGC0008C	IGC0083	
IEEVCTI	IEECVCTI	IEECVCTI	IEECVCTI	
IEECVJCL	IEEVJCL	IEEVMNT1	IEEVJCL	
-	-	IEEVSTAR	-	
IEEVLIN	IEEVLIN	IEFSD569	IEEVLIN	
IEEVLNKT	IEEVLNKT	IEEVRCTL	IEEVLNKT	
IEEVMNT1	IEEVMNT1	IEEVMNT1	IEEVMNT1	
IEEVMNT2	IEEVMNT2	IEEVMNT2	IEEVMNT2	
IEEVPRES	IEBUCBOB	IEEVPRES	IEEVPRES	
IEEVRCTL	IEEVRCTL	IEEVRCTL	IEEVRCTL	
-	IEEVIC	IEVRCTL	IEEVRCTL	
IEEVRFRX	IEEVRFRX	IEECVCTI	IEEVRFRX	
IEEVRJCL	IEEVRJCL	IEEVRRC	IEEVRJCL	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
-	-	IEEVRCTL	-	
IEEVROUT	IGC0503E	IGCO503E	IGC0503E	
IEEVSEND	IEEVSEND	IEEVSEND	IEEVSEND	
-	EXITCODE	IEEVSEND	IEEVSEND	
IEEVSND1	IEEVSND1	IEEVSND1	IEEVSND1	
IEEVSND2	IEEVSND2	IEEVSND2	IEEVSND2	
IEEVSND3	IEEVSND3	IEEVSND3	IEEVSND2	
IEEVSND4	IEEVSND4	IEEVSND4	IEEVSND4	
IEEVSND5	IEEVSND5	IEEVSND2	IEEVSND5	
-	-	IEEVSND3	-	
-	-	IEEVSND8	-	
IEEVSND6	IEEVSND6	IEEVSND4	IEEVSND6	
IEEVSND8	IEEVSND8	IEEVSND8	IEEVSND8	
IEEVSND9	IEEVSND9	IEEVSND9	IEEVSND9	
IEEVSMSG	IEEVSMSG	IEEVRCTL	IEEVSMSG	
-	-	IEEVMNT2	-	
IEEVSTAR	IEEVSTAR	IEEVSTAR	IEEVSTAR	
IEEVWTOR	IGC0103E	IGC0103E	IGC0103E	
IEEXEDNA	IEEXEDNA	IGCXE03D	IEEXEDNA	
IEE00110	IEE00110	IGC00110	IEE00110	
IEE0303D	IEE0303D	IGC0003D	IEE0303D	
IEE0303F	IEE0303F	IGC0003F	IEE0303F	
IEE0403D	IEE0403D	IGC0403D	IEE0403D	
IEE0403F	IEE0403F	IGC0403F	IEE0403F	
IEE0503D	IEE0503D	IGC0503D	IEE0503D	
IEE0603D	IEE0603D	IGC0603D	IEE0603D	
-	-	IEFSD569	-	
IEE1A03D	IEE1A03D	IGC1203D	IEE1A03D	
IEE1B03D	IEE1B03D	IGC1B03D	IEE1B03D	
IEE10110	IEE10110	IGC10110	IEE10110	
IEE1103D	IEE1103D	IGC1103D	IEE1103D	
-	IEE3103D	-	-	
IEE11110	IEE11110	IGC11110	IEE11110	
IEE12110	IEE12110	IGC12110	IEE12110	
IEE1403D	IGC1403D	IGX00011	IGC1403D	
IEE1603D	IEE1603D	IGC1603D	IEE1603D	
IEE1903D	IEE1903D	IGC1903D	IEE1903D	
IEE20110	IEE20110	IGC20110	IEE20110	
IEE21110	IEE21110	IGC21110	IEE21110	
IEE22110	IEE22110	IGC22110	IEE22110	
IEE2303D	IEE2303D	IGC2303D	IEE2303D	
IEE23110	IEE23110	IGC23110	IEE23110	
IEE2903D	IEE2903D	IGC2903D	IEE2903D	
IEE3303D	IEE3303D	IGC3303D	IEE3303D	
IEE3503D	IEE3503D	IGC3503D	IEE3503D	
IEE3703D	IEE3703D	IGC3703D	IEE3703D	
IEE3803D	IEE3803D	IGC3803D	IEE3803D	
IEE40110	IEE40110	IGC40110	IEE40110	
IEE4303D	IEE4303D	IGC4303D	IEE4303D	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEE4403D	IEE4203D	IGC4403D	IEE4403D	
-	IEE4403D	-	-	
IEE4503D	IEE4503D	IGC4503D	IEE4503D	
IEE4603D	IEE4603D	IGC4603D	IEE4603D	
IEE4703D	IEE4703D	IGC4703D	IEE4703D	
IEE4903D	IEE4903D	IGC4903D	IEE4903D	
IEE5603D	IEE5603D	IGC5603D	IEE5603D	
IEE5903D	IEE5903D	IGC5903D	IEE5903D	
IEE60110	IEE60110	IGC60110	IEE60110	
IEE6303D	IEE6303D	IGC6303D	IEE6303D	
IEE6403D	IEE6403D	IGC6403D	IEE6403D	
IEE6503D	IEE6503D	IGC6503D	IEE6503D	
IEE6603D	IEE6603D	IGC6603D	IEE6603D	
IEE6703D	IEE6703D	IGC6703D	IEE6703D	
IEE6803D	IEE6803D	IGC6803D	IEE6803D	
IEE6903D	IEE6903D	IGC6903D	IEE6903D	
IEE7203D	IEE7203D	IGC7203D	IEE7203D	
IEE7303D	IEE7303D	IGC7303D	IEE7303D	
IEE7503D	IEE7503D	IGC7503D	IEE7503D	
IEE7603D	IEE7603D	IGC7603D	IEE7603D	
IEE7703D	IEE7703D	IGC7703D	IEE7703D	
IEE7803D	IEE7803D	IGC7803D	IEE7803D	
IEE7903D	IEE7903D	IGC7903D	IEE7903D	
IEE8703D	IGC8703D	IGC8703D	IGC8703D	
IEE8803D	IGC8803D	-	-	
IEE8903D	IGC8903D	-	-	
IEE9703D	MESSAGES	IEE9803D	IEE9703D	
IEE9803D	IEE9803D	IEE9803D	IEE9803D	
IEE9903D	IEE9903D	IGC9903D	IEE9903D	
IEFAB400	IEFAB400	IEFSD526	IEFAB400	
IEFAB401	IEFAB401	IEFSD526	IEFAB401	
IEFAB402	IEFAB402	IEFSD526	IEFAB402	
IEFAB403	IEALBXCP	IEANUCnn	IEALBXCP	
IEFAB404	IEFAB404	IEFSD569	IEFAB404	
IEFAB405	IEFAB405	IEFW21SD	IEFAB405	
IEFAB406	IEFAB406	IEFW21SD	IEFAB406	
IEFAB407	IEFAB407	IEFW21SD	IEFAB407	
IEFAB408	IEFAB408	IEFW21SD	IEFAB408	
IEFAB416	IEFAB416	IEFAB416	IEFAB416	
IEFAB417	IEFAB417	IEFAB417	IEFAB417	
IEFAB418	IEFAB418	IEFAB418	IEFAB418	
IEFAB420	IEFAB420	IEFAB417	IEFAB420	
-	-	IEFAB418	-	
IEFACTFK	IEFACTFK	IEFW21SD	IEFACTFK	
-	-	IEFJES	-	
-	-	IEFSD161	-	
IEFACTLK	IEFACTLK	IEFW21SD	IEFACTLK	
-	-	IEFJES	-	
-	-	IEFSD161	-	
IEFACTRT	IEFACTRT	IEFSD161	IEFACTRT	
-	-	IEFW21SD	-	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFATECB	IEFQMWR	IEANUCnn	IEFQMWR	
IEFAVFAK	IEFXV001	IEFW21SD	IEFXV001	
IEFBMGET	IEFBMGET	IEFJES	IEFBMGET	
IEFBMINT	IEFBMINT	IEFSD569	IEFBMINT	
IEFBR14	IEFBR14	IEFBR14	IEFBR14	
IEFBMPUR	IEFBMPUR	IEFJES	IEFBMPUR	
IEFBMPUT	IEFBMPUT	IEFJES	IEFBMPUT	
IEFCVFAK	IEFCVOL1	IEFW21SD	IEFCVOL1	
-	IEFCVOL2	IEFVM6LS	-	
-	IEFCVOL3	IEFXVAVR	-	
-	-	IEFSD526	-	
IEFDPOST	IEFDPOST	IEANUCnn	IEFDPOST	IEFSD567
IEFDPOST	IEFDPOST	IEANUCnn	IEFSD567	
IEFDSDRP	IEFDSDRP	IEFDSDRP	IEFDSDRP	
IEFDSLST	IEFDSLST	IEFSD161	IEFDSLST	
IEFDSOAL	IEFDSOAL	IEFDSOAL	IEFDSOAL	
IEFDSOCP	IEFDSOCP	IEFDSO	IEFDSOCP	
IEFDSOFB	IEFDSOFB	IEFDSOFB	IEFDSOFB	
IEFDSOSM	IEFDSOSM	IEFDSOSM	IEFDSOSM	
IEFDSOWR	IEFDSOWR	IEFDSOWR	IEFDSOWR	
IEFDSTBL	IEFDSTBL	IEFSD161	IEFDSTBL	
IEFIDMPM	IEFIDMPM	IEFSD161	IEFIDMPM	
IEFIDUMP	IEFIDUMP	IEFSD161	IEFIDUMP	
IEFIIC	IEFIIC	IEFIIC	IEFIIC	
IEFINTQA	IEFINTQS	IEF160SD	IEFINTQS	
-	IEFINTQM	-	IEFINTQM	
-	INTQMSG1	-	-	
-	INTQMSG2	-	-	
IEFK1MSG	IEFK1MSG	IEEVPRES	IEFK1MSG	
IEFMCVOL	IEFCVOL1	IEFMCVOL	IEFCVOL1	
-	-	-	IEFCVOL2	
-	-	-	IEFCVOL3	
IEFMF102	IEFSD102	IEFSD161	IEFSD102	
IEFMF105	IEFSD105	IEFSD510	IEFSD105	
IEFMF106	IEFMF106	IEFSD510	IEFMF106	
IEFMF263	IEFSD263	IEFSD263	IEFSD263	
IEFMSGJP	IEFMSGJP	IEFJES	IEFMSGJP	
IEFORMAT	IEFORMAT	IEFSQINT	IEFORMAT	
IEFOSC01	IEFOSC01	IEFJES	IEFOSC01	IEFSD082
IEFOSC02	IEFOSC02	IEFJES	IEFOSC02	IEFSD089
IEFOSC03	IEFOSC3B	IEFJES	IEFOSC03	IEFSD083
-	IEFOSC3A	-	-	
-	IEFOSC03	-	-	
-	IEFOSC3C	-	-	
IEFOSC04	IEFOSC04	IEFJES	IEFOSC04	IEFSD084
IEFOSC05	IEFOSC5A	IEFJES	IEFOSC05	IEFSD079
-	IEFOSC05	-	-	
IEFOSC06	IEFOSC06	IEFJES	IEFOSC06	
IEFOSC07	IEFOSC07	IEFJES	IEFOSC07	
IEFOSC08	IEFOSC08	IEFJES	IEFOSC08	IEFSDXXX

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFPARMG	IEFPARMG	IEFSD569	IEFPARMG	
IEFPARMS	IEFPARMS	IEFSD569	IEFPARMS	
IEFPRES	IEFPRES	IEESD569	IEFPRES	
IEFPRTXX	SPRINTER	IEFPRINT	SPRINTER	
IEFQMIFC (See Note 1)				
IEFQMJ01	IEFJESOM	IEFJES	IEFJESQM	
-	IEFQASNM	-	IEFQASNM	
IEFQMJ02	IEFQMJ02	IEFJES	IEFQMJ02	
IEFQMJ03	IEFQMJ03	IEFJES	IEFQMJ03	
IEFQMMAC	IEFQMMAC	IEFJES	IEFQMMAC	
IEFRPREP	IEFRPREP	IEFSD161	IEFRPREP	
IEFRSTRT	IEFRSTRT	IEFRSTRT	IEFRSTRT	
IEFSCAN	IEFSCAN	IEFXVAVR	IEFSCAN	
-	-	IEFW21SD	-	
-	-	IEEVPRES	-	
IEFSDPPT	AUTHSEC	IEFSD161	IEFSDPPT	
-	IEFSDPPT	-	-	
-	LNGRNSEC	-	-	
-	NOCANSEC	-	-	
IEFSD055	IEFSD055	IEFSQINT	IEFSD055	
IEFSD095	IEFSD095	IEFSD095	IEFSD095	
IEFSD096	IEFSD096	IEFSD526	IEFSD096	
-	-	IEFVM6LS	-	
IEFSD097	IEFSD097	IEFSD526	IEFSD097	
IEFSD101	IEFSD062	IEFSD161	IEFSD062	
IEFSD160	IEFSD060	IEF160SD	IEFSD060	
-	-	IEFSD160	-	
IEFSD161	IEFSD061	IEFSD161	IEFSD061	
-	IEFIRET	-	-	
-	IEFSD061	IEF161SD	-	
-	IEFDSOSL	IEFSD161	IEFSDSOSL	
IEFSD162	IEFSD162	IEFSD162	IEFSD062	
-	IEFALRET	-	-	
-	IEFSD62A	-	-	
IEFSD164	IEFSD064	IEFSD161	IEFSD064	
IEFSD165	IEFSD065	IEFSD161	IEFSD065	
IEFSD166	IEFSDA66	IEFSD161	IEFSD066	
-	IEFSD066	-	-	
IEFSD168	IEFSD068	IEFSD161	IEFSD068	
IEFSD180	IEFSD18	IEFW21SD	IEFSD180	
IEFSD195	IEFSD095	IEFVM6LS	IEFSD095	
-	-	IEFSD526	-	
IEFSD21Q	IEFW21SD	IEFW21SD	IEFW21SD	
IEFSD22Q	IEFW22SD	IEFSD161	IEFW22SD	
IEFSD300	IEFSD300	IEFSD300	IEFSD300	
-	IEFSD306	-	IEFSD306	
-	IEFSD300	IEFSQINT	IEFSD300	
-	IEFSD306	-	IEFSD306	
IEFSD301	IEFSD301	IEFSD300	IEFSD301	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFSD302	IEFSD302	IEFSD300	IEFSD302	
IEFSD303	IEFSD303	IEFSD300	IEFSD303	
IEFSD304	IEFSD304	IEFSD304	IEFSD304	
-	IEFSD307	IEFVRRC	IEFSD307	
-	-	IEFVRR1	-	
-	-	IEFVRR3	-	
IEFSD305	IEFSD305	IEFSD304	IEFSD305	
-	IEFSD306	-	IEFSD306	
IEFSD309	IEFSD309	IEFSD300	IEFSD309	
-	-	IEFSD304	-	
IEFSD310	IEFSD310	IEFSD300	IEFSD310	
-	-	IEFSD304	-	
IEFSD31Q	IEFW31SD	IEFSD161	IEFW31SD	
IEFSD41Q	IEFW41SD	IEFSD526	IEFW41SD	
-	-	IEFVM6LS	-	
IEFSD42Q	IEFW42SD	IEFSD161	IEFW42SD	
IEFSD510	IEFSD510	IEFSD510	IEFSD510	
IEFSD515	GO	IEFSD161	IEFSD515	
IEFSD518	IEFSD518	IEFSD518	IEFSD518	
IEFSD519	IEFSD519	IEFSD519	IEFSD519	
IEFSD533	IEFIRC	IEFIRC	IEFIRC	
IEFSD536	IEFVHR	IEFIRC	IEFVHR	
IEFSD551	IEFV15XL	IEFW21SD	IEFV15XL	
-	-	IEFXVAVR	-	
-	-	IEFSD526	-	
IEFSD552	IEFXJX5A	IEFXVAVR	IEFXJX5A	
-	-	IEFSD526	-	
IEFSD567	IEFDPOST	IEANUCnn	IEFDPOST	
IEFSD598	IEFSD598	IEFSD161	IEFSD598	
IEFSMCLD	IEFCLOD	IEFJES	IEFSMCLD	
IEFSMEND	IEFENDS	IEFJES	IEFSMEND	
IEFSMFAT	IEFSMFAT	IEFSD263	IEFSMFAT	
IEFSMFIE	IEFSMFIE	IEFSD162	IEFSMFIE	
IEFSMFLK	IEFACTLK	IEFSD161	IEFACTLK	
IEFSMFSO	IEFSMFSO	IEFJES	IEFSMFSO	
IEFSMFWI	IEFSMFWI	IEFSD161	IEFSMFWI	
IEFSMGET	IEFSMGET	IEFJES	IEFSMGET	
IEFSMIFC	IEFSMIFC	IEFJES	IEFSMIFC	
IEFSMINT	IEFSMINT	IEFSD569	IEFSMINT	
IEFSMODS	IEFSMODS	IEFJES	IEFSMODS	
IEFSMPUT	IEFSMPUT	IEFJES	IEFSMPUT	
IEFSMREP	IEFSMREP	IEFJES	IEFSMREP	
IEFSTDSC	IEFSTDSC	IEFSD161	IEFSTDSC	
IEFUIV	IEFUIV	IEFJES	IEFUIV	
IEFUJI	IEFUJI	IEFSD162	IEFUJI	
IEFUJP	IEFUJP	IEFJES	IEFUJP	
IEFUJV	IEFUJV	IEFUJV	IEJUJV	
IEFUSI	IEFUSI	IEFSD162	IEFUSI	
IEFUSO	IEFUSO	IEFJES	IEFUSO	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFUTL	IEFUTL	IEANUCnn	IEFUTL	
IEFVDA	IEFVDA	IEFIRC	IEFVDA	
IEFVDBSD	IEFVDBSD	IEFIRC	IEFVDBSD	
IEFVEA	IEFVEA	IEFIRC	IEFVEA	
IEFVFA	IEFVFA	IEFIRC	IEFVFA	
IEFVFB	IEFVFB	IEFIRC	IEFVFB	
IEFVGI	IEFVGI	IEFIRC	IEFVGI	
IEFVGK	IEFVGK	IEFIRC	IEFVGK	
IEFVGM	IEFVGM	IEFIRC	IEFVGM	
IEFVGS	IEFVGS	IEFIRC	IEFVGS	
IEFVGT	IEFVGT	IEFIRC	IEFVGT	
IEFVHA	IEFVHA	IEFIRC	IEFVHA	
IEFVHC	IEFVHC	IEFIRC	IEFVHC	
IEFVHCB	IEFVHCB	IEFIRC	IEFVHCB	
IEFVHE	IEFVHE	IEFIRC	IEFVHE	
IEFVHEB	IEFVHEB	IEFIRC	IEFVHEB	
IEFVHEC	IEFVHEC	IEFIRC	IEFVHEC	
IEFVHF	IEFVHF	IEFIRC	IEFVHF	
IEFVHH	IEFVHH	IEFIRC	IEFVHH	
IEFVHL	IEFVHL	IEFIRC	IEFVHL	
IEFVHM	IEHVHM	IEFIRC	IEFVHM	
IEFVHN	IEHVHN	IEFIRC	IEHVHN	
IEFVHQ	IEFVHQ	IEFIRC	IEFVHQ	
IEFVH1	IEFVH1	IEFIRC	IEFVH1	
IEFVINA	IEFVINA	IEFIRC	IEFVINA	
IEFVINB	IEFVINB	IEFIRC	IEFVINB	
IEFVINC	IEFVINC	IEFIRC	IEFVINC	
IEFVINE	IEFVINE	IEFIRC	IEFVINE	
IEFVJA	IEFVJA	IEFIRC	IEFVJA	
IEFVJIMP	IEFVJ	IEFSD161	WTERM030	
-	-	-	IEFVJ	
IEFVJMSG	IEFVJMSG	IEFSD161	IEFVJMSG	
IEFVKIMP	IEFVK	IEFW21SD	IEFVK	
IEFVKMSG	IEFVKMJ1	IEFW21SD	IEFVKMSG	
IEFVMA	IEFVMA	IEFJES	IEFVMA	
IEFVMB	IEFVMB	IEFJES	IEFVMB	
IEFVMC	IEFVMC	IEFJES	IEFVMC	
IEFVMD	IEFVMD	IEFJES	IEFVMD	IEFSD080
IEFVMFAK	IEFVMCVL	IEFMCVOL	IEFVMCVL	
IEFVMLK5	IEFVM6	IEFW21SD	IEFVM6	
IEFVMLS1	IEFVM	IEFW21SD	IEFVM1	
-	VM7000	IEFW215D	IEFVM1	
-	VM7030D	-	-	
-	VM7060	-	-	
-	VM7055	-	-	
-	VM7055AA	-	-	
-	VM7070	-	-	
-	VM7370	-	-	
-	VM7090	-	-	
-	VM7700	-	-	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
-	VM7850	-	-	
-	IEFVMCVL	-	-	
-	VM1BASE	-	-	
-	VM1R1	-	-	
-	VM7750	-	-	
-	VM7130	-	IEFVMVTE	
-	VM7742	-	IEFVMQMI	
-	SWDSFULL	-	-	
-	VM7900	-	IEFVMPDQ	
-	VM7950	-	-	
IEFVMLS6	IEFVM6	IEFVM6LS	IEFVM6	
IEFVMLS7	IEFVM7	IEFVM6LS	IEFVM7	
IEFVMMS1	IEFVM1	IEFMCVOL	IEFVM1	
-	-	IEFVM6LS	-	
-	-	IEFSD526	-	
IEFVM2LS	IEFVM2	IEFW21SD	IEFVM2	
IEFVM3LS	IEFVM3	IEFW21SD	IEFVM3	
IEFVM4LS	IEFVM4	IEFW21SD	IEFVM4	
IEFVM5LS	IEFVM5	IEFW21SD	IEFVM5	
IEFVM76	IEFVM76	IEFW21SD	IEFVM76	
IEFVRRC	IEFVRRC	IEFVRRC	IEFVRRC	
-	IEFVRRCA	IEFVRRC	-	
-	IEFVRRCB	-	-	
IEFVRR1	IEFVRR1	IEFVRR1	IEFVRR1	
IEFVRR2	IEFVRR2	IEFVRR2	IEFVRR2	
IEFVRR3	IEFV3AE	IEFVRR3	IEFVRR3	
-	IEFVRR3	-	-	
IEFVSCDQ	IEFVSCDQ	IEFIRC	IEFVSCDQ	
IEFVSDRA	IEFVSDRA	IEFSD161	IEFVSDRA	
-	-	IEFSD304	-	
IEFVSDRD	IEFVSDRD	IEFSD304	IEFVSDRD	
IEFVSD13	IEFSD090	IEFIRC	IEFSD090	
IEFVSPL	IEFVSPL	IEFIRC	IEFVSPL	
IEFWAALC	IEFWAALC	IEFJES	IEFWAALC	
IEFWAA01	IEFWAA01	IEFJES	IEFWAA01	
IEFWAA02	IEFWAA02	IEFJES	IEFWAA02	
IEFWAA03	IEFWAA03	IEFJES	IEFWAA03	
IEFWAA04	IEFWAA04	IEFJES	IEFWAA04	
IEFWAD	IEFWAD	IEFJES	IEFWAD	
-	-	IEFSD161	-	
-	-	IEFW21SD	-	
IEFWAMAP	IEFWAMAP	IEFWAMAP	IEFWAMAP	
-	IEFWAMP1	-	-	
-	WAMAPVCB	-	-	
IEFWAMGR	IEFWAMGR	IEFJES	IEFWAMGR	
IEFWAMIN	IEFWAMIN	IEFWAMIN	IEFWAMIN	
IEFWAMSG	IEFWAMSG	IEFWARIN	IEFWAMSG	
IEFWARIN	IEFWARIN	IEFWARIN	IEFWARIN	
-	IEFQMRIN	-	-	
IEFWA000	IEFWA000	IEFW21SD	IEFUCBL	
-	-	-	IEFWA002	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
-	-	-	IEFWA7	
IEFWCFAK	IEFWC000	IEFXVAVR	IEFWC000	
IEFWCIMP	IEFWC000	IEFSD526	IEFWC000	
-	-	-	IEFWC002	
IEFWDFAK	IEFWD000	IEFW21SD	IEFWD000	
IEFWD000	IEFWD000	IEFSD526	IEFWDMSG	
-	-	-	IEFWD002	
-	-	-	IEFWD000	
-	-	IEFXVAVR	IEFWDMSG	
-	-	-	IEFWD002	
-	-	-	IEFWD000	
IEFWD001	IEFWD001	IEFXVAVR	IEFWD001	
IEFWEXTA	IEFWEXTA	IEFSD526	IEFWEXTA	
IEFWSTRT	WTERM050	IEFW21SD	IEFWSTRT	
-	WTERM040	-	-	
IEFWSWIN	IEFWSWIT	IEFW21SD	IEFWSWIT	
IEFWTERM	IEFWTERM	IEFSD161	IEFWTERM	
IEFWTP00	IGC0203E	IEFWTO00	IGC0203E	
IEFXAFAK	IEFXA	IEFVM6LS	IEFXA	
IEFXAMSG	IEFXAMSG	IEFW21SD	IEFXAMSG	
IEFXCSSS	IEFXA	IEFW21SD	IEFXA	
-	-	-	IEFXAB00	
IEFXDPTH	IEFXDPTH	IEFW21SD	IEFXDPTH	
-	-	IEFSD526	-	
-	-	IEFXVAVR	-	
IEFXH000	IEFXH000	IEFSD526	IEFXH000	
-	-	IDFXVAVR	-	
IEFXJFAK	IEFXJ000	IEFW21SD	IEFXJ000	
IEFXJIMP	IEFXJ000	IEFVM6LS	IEFXJ000	
IEFXJMSG	IEFXJMSG	IEFVM6LS	IEFXJMSG	
IEFXKFAK	IEFXK000	IEFSD526	IEFXK000	
IEFXKIMP	IEFXK000	IEFVM6LS	IEFXK000	
IEFXKMSG	IEFXKMSG	IEFVM6LS	IEFXKMSG	
IEFXTDMY	IEFXTDMY	IEFSD526	IEFXTDMY	
IEFXTMSG	IEFXTMSG	IEFSD526	IEFXTMSG	
IEFXT00D	IEFXT000	IEFSD526	IEFXT000	
IEFXT002	IEFXT002	IEFSD526	IEFXT002	
IEFXT003	IEFXT003	IEFSD526	IEFXT003	
IEFXVMSG	IEFXVMSG	IEFXVAVR	IEFXVMSG	
IEFXVNSL	IEFXVNSL	IEFXVAVR	IEFXVNSL	
IEFXV001	IEFXV001	IEFXVAVR	IEFXV001	
IEFXV002	IEFXV002	IEFXVAVR	IEFXV002	
IEFXV003	IEFXV003	IEFXVAVR	IEFXV003	
IEFX300A	IEFX3000	IEFSD526	IEFX3000	
-	-	IEFW21SD	-	
-	-	IEFXVAVR	-	
IEFX5FAK	IEFX5000	IEFW21SD	IEFX5000	
IEFX5000	IEFX5000	IEFSD526	IEFX5000	
-	-	IEFXVAVR	-	

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFYNIMP	IEFYN	IEFSD161	WTERN024	
-	-	-	IEFYN	
IEFYNMSG	IEFYNMSG	IEFSD161	IEFYNMSG	
IEFYPJB3	IEFYP	IEFSD161	IEFYP	
IEFYPMSG	YPPMSG4	IEFSD161	IEFYPMSG	
-	YPPMSG3	-	-	
IEFYSVMS	IEFYSVMS	IEFMCVOL	IEFYSVMS	
-	-	IEFSD526	-	
-	-	IEFVM6LS	-	
-	-	IEFW21SD	-	
-	-	IEFSD161	-	
IEFYTVMS	IEFYT	IEFSD161	IEFYT	
IEFZAJB3	IEFZA	IEFSD161	IEFZA	
IEFZGJB1	IEFZGJ	IEFSD161	IEFZGJ	
IEFZGMSG	IEFZGMSG	IEFSD161	IEFZGMSG	
IEFZGST1	IEFZG	IEFSD161	IEFZG	
IEFZGST2	IEFZG2	IEFSD161	IEFZG2	
IEFZHMSG	IEFZH	IEFSD161	IEFZH	
IEF065FK	IEFSD065	IEFSD162	IEFSD065	
IEF160DM	IEFSD060	IEF161SD	IEFSD060	
-	-	IEFIIC	-	
IEF160FK	IEFSD060	IEFSD161	IEFSD060	
IEF161DM	IEFSD061	IEF160SD	IEFSD061	
IEF161FK	IEFSD061	IEFSD160	IEFSD061	
IEF263FK	IEFSD063	IEFSD162	IEFSD063	
IEF300SD	IEFSD300	IEFSQINT	IEFSD300	
IEF304SD	IEFSD304	IEFSQINT	IEFSD304	
IEF41FAK	IEFW41SD	IEFMCVOL	IEFW41SD	
-	-	IEFW21SD	-	
IFSPREIN	IFSPREIN	-	-	
IGCU103D	IEEUNIT1	IGCU103D	IEEUNIT1	IEEUNIT1
IGCU203D	IEEUNIT2	IGCU203D	IEEUNIT2	IEEUNIT2
IGCU303D	IEEUNIT3	IGCU303D	IEEUNIT3	IEEUNIT3
IGCU403D	IEEUNIT4	IGCU403D	IEEUNIT4	IEEUNIT4
IGC5A07B	IEECVETA	IGC5A07B	IEECVETA	IEECVETA
IGC5C07B	IEECVETC	IGC5C07B	IEECVETC	IEECVETC
IGC5D07B	IEECVETD	IGC5D07B	IEECVETD	IEECVETD
IGC5E07B	IEECVETE	IGC5E07B	IEECVETE	IEECVETE
IGC5F07B	IEECVETF	IGC5F07B	IEECVETF	IEECVETF
IGC5G07B	IEECVETG	IGC5G07B	IEECVETG	IEECVETG
IGC5H07B	IEECVETH	IGC5H07B	IEECVETH	IEECVETH
IGC5J07B	IEECVETJ	IGC5J07B	IEECVETJ	IEECVETJ
IGC5K07B	IEECVETK	IGC5K07B	IEECVETK	IEECVETK
IGC5L07B	IEECVETL	IGC5L07B	IEECVETL	IEECVETL
IGC5M07B	IEECVETM	IGC5M07B	IEECVETM	IEECVETM
IGC5N07B	IEECVETN	IGC5N07B	IEECVETN	IEECVETN
IGC5O07B	IEECVETO	IGC5O07B	IEECVETO	IEECVETO
IGC5P07B	IEECVETP	IGC5P07B	IEECVETP	IEECVETP

Cross-Reference Directory Alphabetical List — By Module

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IGC5Q07B	IEECVETQ	IGC5Q07B	IEECVETQ	IEECVETQ
IGC5R07B	IEECVETR	IGC5R07B	IEECVETR	IEECVETR
IGC5S07B	IEECVETS	IGC5S07B	IEECVETS	IEECVETS
IGC5T07B	IEECVETT	IGC5T07B	IEECVETT	IEECVETT
IGC5107B	IEECVET1	IGC5107B	IEECVET1	IEECVET1
IGC5207B	IEECVET2	IGC5207B	IEECVET2	IEECVET2
IGC5307B	IEECVET3	IGC5307B	IEECVET3	IEECVET3
IGC5407B	IEECVET4	IGC5407B	IEECVET4	IEECVET4
IGC5607B	IEECVET6	IGC5607B	IEECVET6	IEECVET6
IGC5707B	IEECVET7	IGC5707B	IEECVET7	IEECVET7
IGC5807B	IEECVET8	IGC5807B	IEECVET8	IEECVET8
IGC5907B	IEECVET9	IGC5907B	IEECVET9	IEECVET9
IGF2403D	IGF2403D	IGF2403D	IGF2403D	
IGFVMCD4	IGFVMCD4	IGC2603D	IGFVMCD4	
IGG019DF	IGG019DF	IGG019DF	IGG019DF	
IGG019DG	IGG019DG	IGG019DG	IGG019DG	
IGG019DH	IGG019DH+8	IGG019DH	IGG019DH	
-	IGG019DH+4	-	-	
-	IGG019DH	-	-	
IGG019DL	IGG019DL	IGG019DL	IGG019DL	
IGG019DM	IGG019DM	IGG019DM	IGG019DM	
IGG019FM	IGG019FM	IGG019FM	IFF019FM	
IGG019MA	IGG019MA	IGG019MA	IEE019MA	
IGG019MB	IGG019MB	IGG019MB	IGG019MB	
IGG019MR	IGG019MR	IGG019MR	IGG019MR	
IGG0196U	IGG0196U	IGG0196U	IGG0196U	
IGG0196V	IGG0196V	IGG0196V	IGG0196V	
IGG0196W	IGG0196W	IGG0196W	IGG0196W	
IGG0196X	IGG0196X	IGG0196X	IGG0196X	
IGG0196Y	IGG0196Y	IGG0196Y	IGG0196Y	
IGG0196Z	IGG0196Z	IGG0196Z	IGG0196Z	
IGG0201M	IGG0201M	IGG0201M	IGG0201M	
IGG0201N	IGG0201N	IGG0201N	IGG0201N	

Note 1: Module IEFQMIFC contains one CSECT, IEFQMIFC. The module has these entry points:

CNVRT
IEFCNVRT
IEFLOCDQ
IEFQAGST
IEFQASGN
IEFQDELE
IEFQMDQ2
IEFQMIFC
IEFQMNQ2
IEFQMRAW
IEFQMSSS
IEFQMUNC
IEFRDWRT

Module Name	Entry Point	Load Module	CSECT	Microfiche Name
IEFSD514				
LOC				
LOCCAN				
LOCDQ				
RD				
WRT				

These load modules contain module IEFQMIFC:

IEFDSDRP
IEFDSO
IEFDSOSM
IEFDSOWR
IEFIIC
IEFMCVOL
IEFQMIFC
IEFSD161
IEFSD162
IEFSD300
IEFSD304
IEFSD510
IEFSD518
IEFSD519
IEFSD526
IEFVM6LS
IEFVRRC
IEFVRR1
IEFVRR3
IEFW21SD
IEFXVAVR
IEF161SD

Cross Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
AUTHSEC	IEFSDPPT	IEFSD161	IEFSDPPT
CNVRT (See Note 2)			
EXITCODE	IEEVSEND	IEEVSEND	IEEVSEND
GO	IEFSD515	IEFSD161	IEFSD515
IEALBXCP	IEFAB403	IEANUNnn	IEALBXCP
IEBUCBOB	IEEVPRES	IEEVPRES	IEEVPRES
IECURAT1	IECURAT1	IEANUCnn	IECURAT1
IEEBAFQE	IEESD568	IEANUCnn	IEEMSER
IEEBA1A0	IEECVCRA	IEEBA1	IEEBA1
IEEBA1A1	IEECVCRA	IEEBA1	IEEBA1
IEEBC1A0	IEECVCRX	IEEBC1PE	IEEBC1PE
IEEBC1A1	IEECVCRX	IEEBC1PE	IEEBC1PE
IEEBC1PE	IEECVCRX	IEECB1PE	IEEBC1PE
IEECIRUP	IEECMAWR	IEECMAWR	IEECIRFC
IEECIR50	IEECIR50	IEANUCnn	IEECIR50
IEECIR51	IEECIR51	IEANUCnn	IEECIR51
IEECJR45	IEECVCRX	IEEBC1PE	IEEBC1PE
IEECLCTX	IEECLCTX	IGCXL07B	IEECLCTX
IEECMAA1	IEECMAWR	IEECMAWR	IEECIRFC
IEECMAWR	IEECMAWR	IEECMAWR	IEECMAWR
IEECMA00	IEECMAWR	IEECMAWR	IEECIRFC
IEECMCTX	IEECMCTX	IGXM07B	IEECMCTX
IEECMDA0	IEECMDSV	IEECMDSV	IEECMDSV
IEECMDA1	IEECMDSV	IEECMDSV	IEECMDSV
IEECMDOM	IEECMDOM	IEECMDOM	IEECMDOM
-	IEECVDOM	-	-
IEECMDSV	IEECMDSV	IEECMDSV	IEECMDSV
IEECMENQ	IEECMWSV	IEECMWSV	IEECMWSV
IEECMPM1	IEECMPM1	IGC0207B	IEECMPM1
IEECMQCN	IEECMWSV	IEECMWSV	IEECMWSV
IEECMWA0	IEECMWSV	IEECMWSV	IEECMWSV
IEECMWA1	IEECMWSV	IEECMWSV	IEECMWSV
IEECMWSV	IEECMWSV	IEECMWSV	IEECMWSV
IEECNCTX	IEECNCTX	IGCXN07B	IEECNCTX
IEECOCTX	IEECOCTX	IGCXO07B	IEECOCTX
IEECVCTI	IEEVCTI	IEECVCTI	IEECVCTI
IEECVCTR	IEECMCTR	IGC0007B	IEECVCTR
IEECVETA	IGC5A07B	IGC5A07B	IEECVETA
IEECVETC	IGC5C07B	IGC5C07B	IEECVETC
IEECVETD	IGC5D07B	IGC5D07B	IEECVETD
IEECVETE	IGC5E07B	IGC5E07B	IEECVETE
IEECVETF	IGC5F07B	IGC5F07B	IEECVETF
IEECVETG	IGC5G07B	IGC5G07B	IEECVETG
IEECVETH	IGC5H07B	IGC5H07B	IEECVETH
IEECVETJ	IGC5J07B	IGC5J07B	IEECVETJ
IEECVETK	IGC5K07B	IGC5K07B	IEECVETK
IEECVETL	IGC5L07B	IGC5L07B	IEECVETL
IEECVETM	IGC5M07B	IGC5M07B	IEECVETM
IEECVETN	IGC5N07B	IGC5N07B	IEECVETN

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEECVETO	IGC5O07B	IGC5O07B	IEECVETO
IEECVETP	IGC5P07B	IGC5P07B	IEECVETP
IEECVETQ	IGC5Q07B	IGC5Q07B	IEECVETQ
IEECVETR	IGC5R07B	IGC5R07B	IEECVETR
IEECVETS	IGC5S07B	IGC5S07B	IEECVETS
IEECVETT	IGC5T07B	IGC5T07B	IEECVETT
IEECVET1	IGC5107B	IGC5107B	IEECVET1
IEECVET2	IGC5207B	IGC5207B	IEECVET2
IEECVET3	IGC5307B	IGC5307B	IEECVET3
IEECVET4	IGC5407B	IGC5407B	IEECVET4
IEECVET6	IGC5607B	IGC5607B	IEECVET6
IEECVET7	IGC5707B	IGC5707B	IEECVET7
IEECVET8	IGC5807B	IGC5807B	IEECVET8
IEECVET9	IGC5907B	IGC5907B	IEECVET9
IEECVOC	IEECVOCX	IGC0I07B	IEECVOC
-	IEECMOCP	IGC2I07B	-
-	IEECVOCC	IGCI107B	-
IEECVML3	IEECVML3	IGC0603E	IEECVML3
IEECVML5	IEECVML5	IGC0703E	IEECVML5
IEECVML6	IEECVML6	IGC0803E	IEECVML6
IEECVML7	IEECVML7	IGC0903E	IEECVML7
IEECVPM	IEECMPMX	IGC0107B	IEECVPM
-	IEECMPMP	IGC2107B	-
-	IEECMPMC	IGC1107B	-
IEECVXIT	IEECVCTE	IEECVXIT	IEECVXIT
IEEDFINA	IEEDFINA	IGX00003	IEEDFINA
IEEDFIN1	IEEDFIN1	IGX00000	IEEDFIN1
IEEDFIN2	IEEDFIN2	IGX01000	IEEDFIN2
IEEDFIN3	IEEDFIN3	IGX02000	IEEDFIN3
IEEDFIN4	IEEDFIN4	IGX00001	IEEDFIN4
IEEDFIN5	IEEDFIN5	IGX00002	IEEDFIN5
IEEDFIN6	IEEDFIN6	IGX03000	IEEDFIN6
IEEDFIN7	IEEDFIN7	IGX04000	IEEDFIN7
IEEDFIN8	IEEDFIN8	IEEDFIN8	IEEDFIN8
-	-	-	IEEDFIN1
-	-	-	IEEDFIN2
-	-	-	IEEDFIN6
IEELGON	IEELGON	IEELGON	IEELGON
IEELGON1	IEELGON1	IEELGON1	IEELGON1
IEELGON2	IEELGON2	IEELGON2	IEELGON2
IEELIST	IEELIST	IEELIST	-
IEELIST1	IEELIST1	IEELIST1	-
IEELOGFC	IEELOGWR	IEELWAIT	IEELWAIT
IEELOGUP	IEELOGWR	IEELWAIT	IEELWAIT
IEELOGWR	IEELOGWR	IEELWAIT	IEELWAIT
IEEMB800	IEEMB800	IEESD569	IEEMB800
IEEMSLT	IEESD568	IEANUCnn	IEEMSER
IEEPSN	IEEPSN	IEEVRCTL	IEEPSN
IEERTE	IEERTE	IEERTE	IEERTE

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEERTE1	IEERTE1	IEERTE1	IEERTE1
IEERTE2	IEERTE2	IEERTE2	IEERTE2
IEERTE3	IEERTE3	IEERTE3	IEERTE3
IEESD561	IEESD561	IGX00004	IEESD561
IEESD562	IEESD562	IGX01004	IEESD562
IEESD563	IEESD563	IGX02004	IEESD563
IEESD564	IEESD564	IGX03004	IEESD564
IEESD565	IEESD565	IGX04004	IEESD565
IEESD566	IEESD566	IGX00012	IEESD566
IEESD575	IEESD575	IGX05004	IEESD575
IEESD576	IEESD576	IGX06004	IEESD576
IEESMFAL	IEESMFAL	IGC0208C	IEESMFAL
IEESMFIO	IEESMFI3	IEESMFI3	IEESMFI3
IEESMFIT	IEESMFIT	IEESMFIT	IEESMFIT
IEESMFI2	IEESMFI2	IEESMFI2	IEESMFI2
IEESMFI3	IEESMFI3	IEESMFI3	IEESMFI3
IEESMFI4	IEESMFIT	IEESMFIT	IEESMFIT
IEESMFMS	IEESMFI3	IEESMFI3	IEESMFI3
IEESMFOI	IEESMFOI	IEESMFOI	IEESMFOI
IEESMFOP	IEESMFOP	IGC0108C	IEESMFOP
IEESMFRB	IEESMFWT	IEANUCnn	IEESMFWT
IEESMFWT	IEESMFWT	IEANUCnn	IEESMFWT
IEEUNIT1	IGCU103D	IGCU103D	IEEUNIT1
IEEUNIT2	IGCU203D	IGCU203D	IEEUNIT2
IEEUNIT3	IGCU303D	IGCU303D	IEEUNIT3
IEEUNIT4	IGCU403D	IGCU403D	IEEUNIT4
IEEVIC	IEEVRCTL	IEEVRCTL	IEEVRCTL
IEEVJCL	IEEVJCL	IEEVSTAR	IEEVJCL
-	-	IEEVMNT1	-
IEEVLIN	IEEVLIN	IEFSD569	IEEVLIN
IEEVLNKT	IEEVLNKT	IEEVRCTL	IEEVLINKT
IEEVMNT1	IEEVMNT1	IEEVMNT1	IEEVMNT1
IEEVMNT2	IEEVMNT2	IEEVMNT2	IEEVMNT2
IEEVRCTL	IEEVRCTL	IEEVRCTL	IEEVRCTL
IEEVRFRX	IEEVRFRX	IEECVCTI	IEEVRFRX
IEEVRJCL	IEEVRJCL	IEEVRRC	IEEVRJCL
-	-	IEEVRCTL	-
IEEVSEND	IEEVSEND	IEEVSEND	IEEVSEND
IEEVSMSG	IEEVSMSG	IEEVRCTL	IEEVSMSG
-	-	IEEVMNT2	-
IEEVSND1	IEEVSND1	IEEVSND1	IEEVSND1
IEEVSND2	IEEVSND2	IEEVSND2	IEEVSND2
IEEVSND3	IEEVSND3	IEEVSND3	IEEVSND3
IEEVSND4	IEEVSND4	IEEVSND4	IEEVSND4
IEEVSND5	IEEVSND5	IEEVSND2	IEEVSND5
-	-	IEEVSND3	-
-	-	IEEVSND8	-
IEEVSND6	IEEVSND6	IEEVSND4	IEEVSND6
IEEVSND8	IEEVSND8	IEEVSND8	IEEVSND8

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEEVSND9	IEEVSND9	IEEVSND9	IEEVSND9
IEEVSTAR	IEEVSTAR	IEEVSTAR	IEEVSTAR
IEEXEDNA	IEEXEDNA	IGCXE03D	IEEXEDNA
IEE00110	IEE00110	IGC00110	IEE00110
IEE0303D	IEE0303D	IGC0003D	IEE0303D
IEE0303F	IEE0303F	IGC0303F	IEE0303F
IEE0403D	IEE0403D	IGC0403D	IEE0403D
IEE0403F	IEE0403F	IGC0403F	IEE0403F
IEE0503D	IEE0503D	IGC0503D	IEE0503D
IEE0603D	IEE0603D	IGC0603D	IEE0603D
IEE0903D	IEE0903D	IGC0903D	IEE0903D
IEE1A03D	IEE1A03D	IGC1203D	IEE1A03D
IEE1B03D	IEE1B03D	IGC1B03D	IEE1B03D
IEE10110	IEE10110	IGC10110	IEE10110
IEE1103D	IEE1103D	IGC1103D	IEE1103D
IEE11110	IEE11110	IGC11110	IEE11110
IEE12110	IEE12110	IGC12110	IEE12110
IEE1603D	IEE1603D	IGC1603D	IEE1603D
IEE1903D	IEE1903D	IGC1903D	IEE1903D
IEE20110	IEE20110	IGC20110	IEE20110
IEE21110	IEE21110	IGC21110	IEE21110
IEE22110	IEE22110	IGC22110	IEE22110
IEE2303D	IEE2303D	IGC2303D	IEE2303D
IEE23110	IEE23110	IGC23110	IEE23110
IEE2903D	IEE2903D	IGC2903D	IEE2903D
IEE3103D	IEE1103D	IGC1103D	IEE1103D
IEE3203D	IEE3203D	IGC3203D	IEE3203D
IEE3303D	IEE3303D	IGC3303D	IEE3303D
IEE3503D	IEE3503D	IGC3503D	IEE3503D
IEE3703D	IEE3703D	IGC3703D	IEE3703D
IEE40110	IEE40110	IGC40110	IEE40110
IEE4203D	IEE4403D	IGC4403D	IEE4403D
IEE4303D	IEE4303D	IGC4303D	IEE4303D
IEE4403D	IEE4403D	IGC4403D	IEE4403D
IEE4503D	IEE4503D	IGC4503D	IEE4503D
IEE4603D	IEE4603D	IGC4603D	IEE4603D
IEE4703D	IEE4703D	IGC4703D	IEE4703D
IEE4903D	IEE4903D	IGC4903D	IEE4903D
-	-	IEFSD569	-
IEE5603D	IEE5603D	IGC5603D	IEE5603D
IEE5903D	IEE5903D	IGC5903D	IEE5903D
IEE60110	IEE60110	IGC60110	IEE60110
IEE6303D	IEE6303D	IGC6303D	IEE6303D
IEE6403D	IEE6403D	IGC6403D	IEE6403D
IEE6503D	IEE6503D	IGC6503D	IEE6503D
IEE6603D	IEE6603D	IGC6603D	IEE6603D
IEE6703D	IEE6703D	IGC6703D	IEE6703D
IEE6803D	IEE6803D	IGC6803D	IEE6803D
IEE6903D	IEE6903D	IGC6903D	IEE6903D

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEE7203D	IEE7203D	IGC7203D	IEE7203D
IEE7303D	IEE7303D	IGC7303D	IEE7303D
IEE7503D	IEE7503D	IGC7503D	IEE7503D
IEE7603D	IEE7603D	IGC7603D	IEE7603D
IEE7703D	IEE7703D	IGC7703D	IEE7703D
IEE7803D	IEE7803D	IGC7803D	IEE7803D
IEE7903D	IEE7903D	IGC7903D	IEE7903D
IEE9803D	IEE9803D	IEE9803D	IEE9803D
IEE9903D	IEE9903D	IGC9903D	IEE9903D
IEFAB400	IEFAB400	IEFSD526	IEFAB400
IEFAB401	IEFAB401	IEFSD526	IEFAB401
IEFAB402	IEFAB402	IEFSD526	IEFAB402
IEFAB404	IEFAB404	IEFSD569	IEFAB404
IEFAB405	IEFAB405	IEFW21SD	IEFAB405
IEFAB406	IEFAB406	IEFW21SD	IEFAB406
IEFAB407	IEFAB407	IEFW21SD	IEFAB407
IEFAB408	IEFAB408	IEFW21SD	IEFAB408
IEFAB416	IEFAB416	IEFAB416	IEFAB416
IEFAB417	IEFAB417	IEFAB417	IEFAB417
IEFAB418	IEFAB418	IEFAB418	IEFAB418
IEFAB420	IEFAB420	IEFAB417	IEFAB420
-	-	IEFAB418	-
IEFACTFK	IEFACTFK	IEFW21SD	IEFACTFK
-	-	IEFJES	-
-	-	IEFSD161	-
IEFACTLK	IEFACTLK	IEFW21SD	IEFACTLK
-	-	IEFJES	-
-	-	IEFSD161	-
-	IEFSMFLK	IEFSD161	-
-	-	IEFSD161	-
IEFACTRT	IEFACTRT	IEFSD161	IEFACTRT
-	-	IEFW21SD	-
IEFALRET	IEFSD162	IEFSD162	IEFSD062
IEFBMGET	IEFBMGET	IEFJES	IEFBMGET
IEFBMINT	IEFBMINT	IEFSD569	IEFMINT
IEFBMPUR	IEFBMPUR	IEFJES	IEFBMPUR
IEFBMPUT	IEFBMPUT	IEFJES	IEFBMPUT
IEFBR14	IEFBR14	IEFBR14	IEFBR14
IEFCLOD	IEFSMCLD	IEFJES	IEFSMCLD
IEFCNVRT (See Note 2)			
IEFCVOL1	IEFMCVOL	IEFMCVOL	IEFCVOL1
-	IEFCVFAK	IEFW21SD	IEFCVOL3
-	-	-	IEFCVOL2
IEFCVOL2	IEFCVFAK	IEFVM6LS	IEFCVOL1
IEFCVOL3	IEFCVFAK	IEFSD526	IEFCVOL1
-	-	IEFXVAVR	-
IEFDPOST	IEFDPOST	IEANUCnn	IEFDPOST
IEFDSDRP	IEFDSDRP	IEFDSDRP	IEFDSDRP
IEFDSLST	IEFDSLST	IEFSD161	IEFDSLST
IEFDSOAL	IEFDSOAL	IEFDSOAL	IEFDSOAL
IEFDSOCP	IEFDSOCP	IEFDSO	IEFDSOCP

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEFDSOFB	IEFDSOFB	IEFDSOFB	IEFDSOFB
IEFDSOSM	IEFDSOSM	IEFDSOSM	IEFDSOSM
IEFDSOWR	IEFDSOWR	IEFDSOWR	IEFDSOWR
IEFDSTBL	IEFDSTBL	IEFSD161	IEFDSTBL
IEFENDS	IEFSMEND	IEFJES	IEFSMEND
IEFIDMPM	IEFIDMPM	IEFSD161	IEFIDMPM
IEFIDUMP	IEFIDUMP	IEFSD161	IEFIDUMP
IEFIIC	IEFIIC	IEFIIC	IEFIIC
IEFINTQM	IEFINTQA	IEF160SD	IEFINTQM
IEFINTQS	IEFINTQA	IEF160SD	IEFINTQS
IEFIRC	IEFSD533	IEFIRC	IEFIRC
IEFIRET	IEFSD161	IEFSD161	IEFSD061
IEFJESQM	IEFQMJ01	IEFJES	IEFJESQM
IEFK1MSG	IEFK1MSG	IEEVPRES	IEFK2MSG
IEFLOCDQ (See Note 2)			
IEFMF106	IEFMF106	IEFSD510	IEFMF106
IEFMSGJP	IEFMSGJP	IEFJES	IEFMSGJP
IEFORMAT	IEFORMAT	IEFSQINT	IEFORMAT
IEFOSC01	IEFOSC01	IEFJES	IEFOSC01
IEFOSC02	IEFOSC02	IEFJES	IEFOSC02
IEFOSC03	IEFOSC03	IEFJES	IEFOSC03
IEFOSC04	IEFOSC04	IEFJES	IEFOSC04
IEFOSC05	IEFOSC05	IEFJES	IEFOSC05
IEFOSC06	IEFOSC06	IEFJES	IEFOSC06
IEFOSC07	IEFOSC07	IEFJES	IEFOSC07
IEFOSC08	IEFOSC08	IEFJES	IEFOSC08
IEFOSC3A	IEFOSC03	IEFJES	IEFOSC03
IEFOSC3B	IEFOSC03	IEFJES	IEFOSC03
IEFOSC5A	IEFOSC05	IEFJES	IEFOSC05
IEFPARMG	IEFPARMG	IEFSD569	IEFPARMG
IEFPARMS	IEFPARMS	IEFSD569	IEFPARMS
IEFPRES	IEFPRES	IEFSD569	IEFPRES
IEFQAGST (See Note 2)			
IEFQASGN (See Note 2)			
IEFQASNM (See Note 2)			
IEFQDELE (See Note 2)			
IEFQMDQ2 (See Note 2)			
IEFQMIFC (See Note 2)			
IEFQMJ02	IEFQMJ02	IEFJES	IEFQMJ02
IEFQMJ03	IEFQMJ03	IEFJES	IEFQMJ03
IEFQMMAC (See Note 2)			
IEFQMNQ2 (See Note 2)			
IEFQMRAW (See Note 2)			
IEFQMSSS (See Note 2)			
IEFQMUNC (See Note 2)			
IEFQMWR	IEFATECB	IEANUCnn	IEFQMWR
IEFRDWRT (See Note 2)			
IEFRPREP	IEFRPREP	IEFSD161	IEFRPREP
IEFRSTRT	IEFRSTRT	IEFRSTRT	IEFRSTRT

Entry Point	Module Name	Load Module	CSECT
IEFSCAN	IEFSCAN	IEFXVAVR	IEFSCAN
IEFSCAN	IEFSCAN	IEFW21SD	IEFSCAN
IEFSCAN	IEFSCAN	IEEVPRES	IEFSCAN
IEFSDA66	IEFSD166	IEFSD161	IEFSD066
IEFSDPPT	IEFSDPPT	IEFSD161	IEFSDPPT
IEFSD055	IEFSD055	IEFSQINT	IEFSD055
IEFSD060	IEFSD160	IEF160SD	IEFSD060
-	-	IEFSD160	-
-	IEF160FK	IEFIIC	-
-	-	IEF161SD	-
-	-	IEFSD161	-
-	IEF160DM	IEFIIC	-
-	-	IEF161SD	-
-	-	IEFSD161	-
IEFSD061	IEF161DM	IEF160SD	IEFSD061
-	IEF161FK	IEFSD160	-
-	IEFSD161	IEFSD161	-
-	-	IEF161SD	-
IEFSD062	IEFSD101	IEFSD161	IEFSD062
-	IEFSD162	IEFSD162	IEFSD062
IEFSD063	IEF263FK	IEFSD162	IEFSD063
IEFSD064	IEFSD164	IEFSD161	IEFSD064
IEFSD065	IEFSD165	IEFSD161	IEFSD065
-	IEF065FK	-	-
IEFSD066	IEFSD166	IEFSD161	IEFSD066
IEFSD068	IEFSD168	IEFSD161	IDFSD068
IEFSD090	IEFVSD13	IEFIRC	IEFSD090
IEFSD095	IEFSD195	IEFVM6LS	IEFSD095
-	-	IEFSD526	-
-	IEFSD095	IEFSD095	IEFSD095
IEFSD096	IEFSD096	IEFSD526	IEFSD096
-	-	IEFVM6LS	-
IEFSD097	IEFSD097	IEFSD526	IEFSD097
IEFSD102	IEFMF102	IEFSD161	IEFSD102
IEFSD105	IEFMF105	IEFSD510	IEFSD105
IEFSD162	IEFSD162	IEFSD162	IEFSD062
IEFSD180	IEFSD180	IEFW21SD	IEFSD180
IEFSD263	IEFMF263	IEFSD263	IEFSD263
IEFSD300	IEFSD300	IEFSD300	IEFSD300
-	IEF300SD	IEFSQINT	-
IEFSD301	IEFSD301	IEFSD300	IEFSD301
IEFSD302	IEFSD302	IEFSD300	IEFSD302
IEFSD303	IEFSD303	IEFSD300	IEFSD303
IEFSD304	IEFSD304	IEFSD304	IEFSD304
-	IEF304SD	IEFSQINT	-
-	-	IEFVRRC	-
-	-	IEFVRR1	-
-	-	IEFVRR3	-
IEFSD305	IEFSD305	IEFSD305	IEFSD305

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEFSD306	IEFSD305	IEFSD305	IEFSD306
-	IEFSD300	IEFSD300	-
-	-	IEFSQINT	-
IEFSD307	IEFSD304	IEFSD307	IEFSD307
-	-	IEFSD304	-
-	-	IEFVRRC	-
-	-	IEFVRR1	-
-	-	IEFVRR3	-
IEFSD309	IEFSD309	IEFSD300	IEFSD309
-	-	IEFSD304	-
IEFSD310	IEFSD310	IEFSD300	IEFSD310
-	-	IEFSD304	-
IEFSD510	IEFSD510	IEFSD510	IEFSD510
IEFSD514 (See Note 2)			
IEFSD518	IEFSD518	IEFSD518	IEFSD518
IEFSD519	IEFSD519	IEFSD519	IEFSD519
IEFSD569	-	IEFSD569	IEFSD569
IEFSD598	IEFSD598	IEFSD161	IEFSD598
IEFSD62A	IEFSD162	IEFSD162	IEFSD062
IEFSIDSC	IEFSTDSC	IEFSD161	IEFSTDSC
IEFSMFAT	IEFSMFAT	IEFSD263	IEFSMFAT
IEFSMFIE	IEFSMFIE	IEFSD162	IEFSMFIE
IEFSMFSO	IEFSMFSO	IEFJES	IEFSMFSO
IEFSMFWI	IEFSMFWI	IEFSD161	IEFSMFWI
IEFSMGET	IEFSMGET	IEFJES	IEFSMGET
IEFSMIFC	IEFSMIFC	IEFJES	IEFSMIFC
IEFSMINT	IEFSMINT	IEFSD569	IEFSMINT
IEFSMODS	IEFSMODS	IEFJES	IEFSMODS
IEFSMPUT	IEFSMPUT	IEFJES	IEFSMPUT
IEFSMREP	IEFSMREP	IEFJES	IEFSMREP
IEFUIV	IEFUIV	IEFJES	IEFUIV
IEFUJI	IEFUJI	IEFSD162	IEFUJI
IEFUJP	IEFUJP	IEFJES	IEFUJP
IEFUJV	IEFUJV	IEFUJV	IEJUJV
IEFUSI	IEFUSI	IEFSD162	IEFUSI
IEFUSO	IEFUSO	IEFJES	IEFUSO
IEFUTL	IEFUTL	IEANUCnn	IEFUTL
IEFV3AE	IEFSD309	IEFVRR3	IEFVRR3
-	IEFVRR3	-	-
IEFVDA	IEFVDA	IEFIRC	IEFVDA
IEFVDBSD	IEFVDBSD	IEFIRC	IEFVDBSD
IEFVEA	IEFVEA	IEFIRC	IEFVEA
IEFVFA	IEFVFA	IEFIRC	IEFVFA
IEFVFB	IEFVFB	IEFIRC	IEFVFB
IEFVGI	IEFVGI	IEFIRC	IEFVGI
IEFVGK	IEFVGK	IEFIRC	IEFVGK
IEFVGM	IEFVGM	IEFIRC	IEFVGM
IEFVGS	IEFVGS	IEFIRC	IEFVGS

Entry Point	Module Name	Load Module	CSECT
IEFVGT	IEFVGT	IEFIRC	IEFVGT
IEFVHA	IEFVHA	IEFIRC	IEFVHA
IEFVHC	IEFVHC	IEFIRC	IEFVHC
IEFVHCB	IEFVHCB	IEFIRC	IEFVHCB
IEFVHE	IEFVHE	IEFIRC	IEFVHE
IEFVHEB	IEFVHEB	IEFIRC	IEFVHEB
IEFVHEC	IEFVHEC	IEFIRC	IEFVHEC
IEFVHF	IEFVHF	IEFIRC	IEFVHF
IEFVHH	IEFVHH	IEFIRC	IEFVHH
IEFVHL	IEFVHL	IEFIRC	IEFVHL
IEFVHM	IEFVHM	IEFIRC	IEFVHM
IEFVHN	IEFVHN	IEFIRC	IEFVHN
IEFVHQ	IEFVHQ	IEFIRC	IEFVHQ
IEFVHR	IEFSD536	IEFIRC	IEFVHR
IEFVH1	IEFVH1	IEFIRC	IEFVH1
IEFVINA	IEFVINA	IEFIRC	IEFVINA
IEFVINB	IEFVINB	IEFIRC	IEFVINB
IEFVINC	IEFVINC	IEFIRC	IEFVINC
IEFVINE	IEFVINE	IEFIRC	IEFVINE
IEFVJ	IEFVJIMP	IEFSD161	WTERM030
IEFVJ	IEFVJIMP	IEFSD161	IEFVJ
IEFVJA	IEFVJA	IEFIRC	IEFVJA
IEFVJMSG	IEFVJMSG	IEFSD161	IEFVJMSG
IEFVK	IEFVKIMP	IEFW21SD	IEFVK
IEFVKMJ1	IEFVKMSG	IEFW21SD	IEFVKMSG
IEFVM	IEFVMLS1	IEFW21SD	IEFVM1
IEFVMA	IEFVMLS1	IEFW21SD	IEFVM1
IEFVMA	IEFVMA	IEFJES	IEFVMA
IEFVMB	IEFVMB	IEFJES	IEFVMB
IEFVMC	IEFVMC	IEFJES	IEFVMC
IEFVMCVL	IEFVMLS1	IEFW21SD	IEFVM1
-	IEFVMFAK	IEFMCVOL	IEFVMCVL
IEFVMD	IEFVMD	IEFJES	IEFVMD
IEFVM1	IEFVMMS1	IEFMCVOL	IEFVM1
-	-	IEFSD526	-
-	-	IEFVM6LS	-
IEFVM2	IEFVM2LS	IEFW21SD	IEFVM2
IEFVM3	IEFVM3LS	IEFW21SD	IEFVM3
IEFVM4	IEFVM4LS	IEFW21SD	IEFVM4
IEFVM5	IEFVM5LS	IEFW21SD	IEFVM5
IEFVM6	IEFVMLS6	IEFVM6LS	IEFVM6
-	IEFVMLK5	IEFW21SD	-
IEFVM7	IEFVMLS7	IEFVM6LS	IEFVM7
IEFVM76	IEFVM76	IEFW21SD	IEFVM76
IEFVRRC	IEFVRRC	IEFVRRC	IEFVRRC
IEFVRRCA	IEFVRRC	IEFVRRC	IEFVRRC
IEFVRRCB	IEFVRRC	IEFVRRC	IEFVRRC
IEFVRR1	IEFVRR1	IEFVRR1	IEFVRR1
IEFVRR2	IEFVRR2	IEFVRR2	IEFVRR2

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEFVRR3	IEFVRR3	IEFVRR3	IEFVRR3
IEFVSCDQ	IEFVSCDQ	IEFIRC	IEFVSCDQ
IEFVSDRD	IEFVSDRD	IEFSD304	IEFVSDRD
IEFVSPL	IEFVSPL	IEFIRC	IEFVSPL
IEFV15XL	IEFSD551	IEFSD526	IEFV15XL
-	-	IEFW21SD	-
-	-	IEFXVAVR	-
IEFWAALC	IEFWAALC	IEFJES	IEFWAALC
IEFWAA01	IEFWAA01	IEFJES	IEFWAA01
IEFWAA02	IEFWAA02	IEFJES	IEFWAA02
IEFWAA03	IEFWAA03	IEFJES	IEFWAA03
IEFWAA04	IEFWAA04	IEFJES	IEFWAA04
IEFWAD	IEFWAD	IEFJES	IEFWAD
-	-	IEFSD161	-
-	-	IEFW21SD	-
IEFWAMGR	IEFWAMGR	IEFJES	IEFWAMGR
IEFWAMSG	IEFWAMSG	IEFWAMIN	IEFWAMSG
IEFWAMIN	IEFWAMIN	IEFWAMIN	IEFWAMIN
IEFWARIN	IEFWARIN	IEFWARIN	IEFWARIN
IEFWA000	IEFWA000	IEFW21SD	IEFUCBL
-	-	-	IEFWA002
-	-	-	IEFWA7
IEFWC000	IEFWCFAK	IEFXVAVR	IEFWC000
-	IEFWCIMP	IEFSD526	IEFWC002
-	IEFWCFAK	-	-
IEFWD000	IEFWD000	IEFSD526	IEFWDMSG
-	-	-	IEFWD002
-	IEFWDFAK	IEFW21SD	IEFWD000
-	-	IEFXVAVR	IEFWDMSG
-	-	-	IEFWD002
-	-	-	IEFWD000
IEFWD001	IEFWD001	IEFXVAVR	IEFWD001
IEFWEXTA	IEFWEXTA	IEFSD526	IEFWEXTA
IEFWSWIT	IEFWSWIN	IEFW21SD	IEFWSWIT
IEFWTERM	IEFWTERM	IEFSD161	IEFWTERM
IEFW21SD	IEFSD21Q	IEFW21SD	IEFW21SD
IEFW22SD	IEFSD22Q	IEFSD161	IEFW22SD
IEFW31SD	IEFSD31Q	IEFSD161	IEFW31SD
IEFW41SD	IEFSD41Q	IEFSD526	IEFW41SD
-	IEF41FAK	IEFMCVOL	IEFW41SD
-	-	IEFW21SD	-
-	-	IEFVM6LS	-
IEFW42SD	IEFSD42Q	IEFSD161	IEFW42SD
IEFXA	IEFXCSSS	IEFW21SD	IEFXA
-	IEFXAFAK	IEFVM6LS	-
IEFXAMSG	IEFXAMSG	IEFW21SD	IEFXAMSG
IEFXDPTH	IEFXDPTH	IEFW21SD	IEFXDPTH
-	-	IEFSD526	-
-	-	IEFXVAVR	-

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IEFXH000	IEFXH000	IEFXVAVR	IEFXH000
IEFXJMSG	IEFXJMSG	IEFVM6LS	IEFXJMSG
IEFXJX5A	IEFSD552	IEFSD526	IEFXJX5A
-	-	IEFSVAVR	-
IEFXJ000	IEFXJIMP	IEFVM6LS	IEFXJ000
-	IEFXJFAK	IEFW21SD	-
IEFXKMSG	IEFXKMSG	IEFVM6LS	IEFXKMSG
IEFXK000	IEFXKIMP	IEFVM6LS	IEFXK000
-	IEFXKFAK	IEFSD526	-
IEFXTDMY	IEFXTDMY	IEFSD526	IEFXTDMY
IEFXTMSG	IEFXTMSG	IEFSD526	IEFXTMSG
IEFXT000	IEFXT00D	IEFSD526	IEFXT000
IEFXT002	IEFXT002	IEFSD526	IEFXT002
IEFXT003	IEFXT003	IEFSD526	IEFXT003
IEFXVMSG	IEFXVMSG	IEFXVAVR	IEFXVMSG
IEFXVNSL	IEFXVNSL	IEFXVAVR	IEFSVNSL
IEFXV001	IEFXV001	IEFXVAVR	IEFXV001
-	IEFAVFAK	IEFW21SD	-
IEFXV002	IEFXV002	IEFXVAVR	IEFXV002
IEFXV003	IEFXV003	IEFXVAVR	IEFXV003
IEFX3000	IEFX300A	IEFSD526	IEFX3000
-	-	IEFW21SD	-
-	-	IEFXVAVR	-
IEFX5000	IEFX5000	IEFSD526	IEFX5000
-	-	IEFXVAVR	-
-	IEFX5FAK	IEFW21SD	IEFX5000
IEFYN	IEFYNIMP	IEFSD161	WTERN024
-	-	-	IEFYN
IEFYNMSG	IEFYNMSG	IEFSD161	IEFYNMSG
IEFYP	IEFYPJB3	IEFSD161	IEFYP
IEFYSVMS	IEFYSVMS	IEFMCVOL	IEFYSVMS
-	-	IEFSD526	-
-	-	IEFVM6LS	-
-	-	IEFW21SD	-
-	-	IEFSD161	-
IEFYT	IEFYTVMS	IEFSD161	IEFYT
IEFZA	IEFZAJB3	IEFSD161	IEFZA
IEFZG	IEFZGST1	IEFSD161	IEFZG
IEFZGJ	IEFZGJB1	IEFSD161	IEFZGJ
IEFZGMSG	IEFZGMSG	IEFSD161	IEFZGMSG
IEFZG2	IEFZGST2	IEFSD161	IEFZG2
IEFZH	IEFZHMSG	IEFSD161	IEFZH
IFSPREIN	IFSPREIN	-	-
IGC0003E	IEEMFWTO	IGC0003E	IGC0003E
IGC0008G	IEECXDOM	IGC0008G	IGC0008G
IGC0083	IEESMF8C	IGC0008C	IGC0083
IGC0103E	IEEVWTOR	IGC0103E	IGC0103E
IGC0203E	IEFWTP00	IEFWTO00	IGC0203E
IGC0503E	IEEVROUT	IGC0503E	IGC0503E

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
IGC0907B	IEECMWTL	IGC0907B	IGC0907B
IGC1403D	IEE1403D	IGX00011	IGC1403D
IGC1803D	IEESD571	IGC1803D	IGC1803D
IGC8703	IEE8703	IGC8703	IGC8703
IGFVMCD4	IGF2603D	IGF2603D	IGFVMCD4
IGF2403D	IGF2403D	IGF2403D	IGF2403D
IGG019DF	IGG019DF	IGG019DF	IGG019DF
IGG019DG	IGG019DG	IGG019DG	IGG019DG
IGG019DH+8	IGG019DH	IGG019DH	IGG019DH
IGG019DH+4	IGG019DH	IGG019DH	IGG019DH
IGG019DH	IGG019DH	IGG019DH	IGG019DH
IGG019DL	IGG019DL	IGG019DL	IGG019DL
IGG019DM	IGG019DM	IGG019DM	IGG019DM
IGG019FM	IGG019FM	IGG019FM	IFF019M4
IGG019MA	IGG019MA	IGG019MA	IEE019MA
IGG019MB	IGG019MB	IGG019MB	IGG019MB
IGG019MR	IGG019MR	IGG019MR	IGG019MR
IGG0196U	IGG0196U	IGG0196U	IGG0196U
IGG0196V	IGG0196V	IGG0196V	IGG0196V
IGG0196W	IGG0196W	IGG0196W	IGG0196W
IGG0196X	IGG0196X	IGG0196X	IGG0196X
IGG0196Y	IGG0196Y	IGG0196Y	IGG0196Y
IGG0196Z	IGG0196Z	IGG0196Z	IGG0196Z
IGG0201M	IGG0201M	IGG0201M	IGG0201M
IGG0201N	IGG0201N	IGG0201N	IGG0201N
INTQMSG1	IEFINTQA	IEF160SD	IEFINTQM
INTQMSG2	IEFINTQA	IEF160SD	IEFINTQM
LNGRNSEC	IEFSDPPT	IEFSD161	IEFSDPPT
LOC	(See Note 2)		
LOCCAN	(See Note 2)		
LOCDQ	(See Note 2)		
MESSAGES	IEE9703D	IEE9803D	IEE9703D
NOCANSEC	IEFSDPPT	IEFSD161	IEFSDPPT
RD	(See Note 2)		
SPRINTER	IEFPRTXX	IEFPRINT	SPRINTER
VMAPOST	IEFVMA	IEFJES	IEFVMA
VMA085	IEFVMA	IEFJES	IEFVMA
VMA103	IEFVMA	IEFJES	IEFVMA
VM1BASE	IEFVMLS1	IEFW21SD	IEFVM1
VM1R1	IEFVMLS1	IEFW21SD	IEFVM1
VM1R2	IEFVMLS1	IEFW21SD	IEFVM1
VM7000	IEFVMLS1	IEFW21SD	IEFVM1
VM7055	IEFVMLS1	IEFW21SD	IEFVM1
VM7060	IEFVMLS1	IEFW21SD	IEFVM1
VM7090	IEFVMLS1	IEFW21SD	IEFVM1
VM7130	IEFVMLS1	IEFW21SD	IEFVMVTE
VM7370	IEFVMLS1	IEFW21SD	IEFVM1
VM7700	IEFVMLS1	IEFW21SD	IEFVM1
VM7742	IEFVMLS1	IEFW21SD	IEFVMQMI

Cross-Reference Directory Alphabetical List — By Entry Point

Entry Point	Module Name	Load Module	CSECT
VM7750	IEFVMLS1	IEFW21SD	IEFVM1
VM7850	IEFVMLS1	IEFW21SD	IEFVM1
VM7900	IEFVMLS1	IEFW21SD	IEFVMPDQ
VM7950	IEFVMLS1	IEFW21SD	IEFVMPDQ
VM7030D	IEFVMLS1	IEFW21SD	IEFVM1
VM7055AA	IEFVMLS1	IEFW21SD	IEFVM1
WRABXLE	IEECMAWR	IEECMAWR	IEECIRFC
WRT	(See Note 2)		
WTERM040	IEFWSTRT	IEFW21SD	IEFWSTRT
WTERM050	IEFWSTRT	IEFW21SD	IEFWSTRT
YPPMSG3	IEFYPMSG	IEFSD161	IEFYPMSG
YPPMSG4	IEFYPMSG	IEFSD161	IEFYPMSG

Note 2: The following entry points all appear in module IEFQMIFC:

CNVRT	IEFQMUNC
IEFCNVRT	IEFRDWRT
IEFLOCDQ	IEFSD514
IEFQAGST	LOC
IEFQASGN	LOCCAN
IEFQDELE	LOCDQ
IEFQMDQ2	WRT
IEFQMIFC	RD
IEFQMNQ2	
IEFQMRAW	
IEFQMSSS	

Module IEFQMIFC appears in these load modules:

IEFDSDRP	IEFSD510
IEFDSO	IEFSD518
IEFDSOSM	IEFSD519
IEFDSOWR	IEFSD526
IEFIIC	IEFVM6LS
IEFMCVOL	IEFVRRC
IEFQMIFC	IEFVRR1
IEFSD161	IEFVRR3
IEFSD162	IEFW21SD
IEFSD300	IEFXVAVR
IEFSD304	IEF161SD

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
IEANUCnn	IEFAB403	IEALBXCP	IEALBXCP
-	IEFSD567	IEFDPOST	IEFDPOST
-	IEFUTL	IEFUTL	IEFUTL
-	-	IEESMFRB	-
-	-	IEESMFWT	-
-	IECURAT1	IECURAT1	IECURAT1
-	IEECIR50	IEECIR50	IEECIR50
-	IEECIR51	IEECIR51	IEECIR51
-	IEESD568	IEEMSLT	IEEMSER
-	-	IEEBAFQE	-
-	IEFATECB	IEFQMWR	IEFQMWR
-	IEFSD567	IEFDPOST	IEFDPOST
IEEBA1	IEECVCRA	IEEBA1A1	IEEBA1
-	-	IEEBA1A0	-
IEEBC1PE	IEECVCRX	IEECJR45	IEEBC1PE
-	-	IEEBC1A1	-
-	-	IEEBC1A0	-
-	-	IEEBC1PE	-
IEECMAWR	IEECMAWR	IEECIRUP	IEECIRFC
-	-	IEECMAA1	-
-	-	IEECMA00	-
-	-	WRABXLE	-
-	-	IEECMAWR	IEECMAWR
IEECMDOM	IEECVDOM	IEECMDOM	IEECMDOM
-	IEECMDOM	-	-
IEECMDSV	IEECMDSV	IEECMDA1	IEECMDSV
-	-	IEECMDA0	-
-	-	IEECMDSV	-
IEECMWSV	IEECMWSV	IEECMQCN	IEECMWSV
-	-	IEECMENQ	-
-	-	IEECMWA1	-
-	-	IEECMWA0	-
-	-	IEECMWSV	-
IEECVCTI	IEEVRFRX	IEEVRFRX	IEEVRFRX
-	IEEVCTI	IEECVCTI	IEECVCTI
IEECVXIT	IEECVCTE	IEECVXIT	IEECVXIT
IEEDFIN8	IEEDFIN8	IEEDFIN8	IEEDFIN8
-	-	-	IEEDFIN1
-	-	-	IEEDFIN2
-	-	-	IEEDFIN6
IEELGON	IEELGON	IEELGON	IEELGON
IEELGON1	IEELGON1	IEELGON1	IEELGON1
IEELGON2	IEELGON2	IEELGON2	IEELGON2
IEELIST	IEELIST	IEELIST	-
IEELIST1	IEELIST1	IEELIST1	-
IEELWAIT	IEELOGWR	IEELOGUP	IEELWAIT
-	-	IEELOGFC	-
-	-	IEELOGWR	-
IEERTE	IEERTE	IEERTE	IEERTE
IEERTE1	IEERTE1	IEERTE1	IEERTE1

Load Module	Module Name	Entry Point	CSECT
IEERTE2	IEERTE2	IEERTE2	IEERTE2
IEERTE3	IEERTE3	IEERTE3	IEERTE3
IEESMFIT	IEESMFIT	IEESMFIT	IEESMFIT
-	-	IEESMFI4	-
IEESMFI2	IEESMFI2	IEESMFI2	IEESMFI2
IEESMFI3	IEESMFI3	IEESMFMS	IEESMFI3
-	-	IEESMFI3	-
-	-	IEESMFIO	-
IEESMFOI	IEESMFOI	IEESMFOI	IEESMFOI
IEEVLIN	IEEVLIN	IEEVLIN	IEEVLIN
IEEVLNKT	IEEVLNKT	IEEVLNKT	IEEVLNKT
IEEVMNT1	IEEVMNT1	IEEVMNT1	IEEVMNT1
IEEVMNT2	IEEVMNT2	IEEVMNT2	IEEVMNT2
IEEVPRES	IEFSCAN	IEFSCAN	IEFSCAN
-	IEFK1MSG	IEFK1MSG	IEFK2MSG
-	IEEVPRES	IEBUCBOB	IEEVPRES
IEEVRCTL	IEEVRJCL	IEEVRJCL	IEEVRJCL
-	IEEPSN	IEEPSN	IEEPSN
-	IEEVSMSG	IEEVSMSG	IEEVSMSG
-	IEEVRCTL	IEEVIC	IEEVRCTL
IEEVRRC	IEEVRJCL	IEEVRJCL	IEEVRJCL
IEEVSEND	IEEVSEND	IEEVSEND	IEEVSEND
-	IEEVSEND	EXITCODE	IEEVSEND
-	IEEVSND1	IEEVSND1	IEEVSND1
IEEVSND2	IEEVSND2	IEEVSND2	IEEVSND2
-	IEEVSND5	IEEVSND5	IEEVSND5
IEEVSND3	IEEVSND3	IEEVSND3	IEEDSND3
-	IEEVSND5	IEEVSND5	IEEVSND5
IEEVSND4	IEEVSND4	IEEVSND4	IEEVSND4
-	IEEVSND6	IEEVSND6	IEEVSND6
IEEVSND8	IEEVSND8	IEEVSND8	IEEVSND8
-	IEEVSND5	IEEVSND5	IEEVSND5
IEEVSND9	IEEVSND9	IEEVSND9	IEEVSND9
IEEVSTAR	IEEVJCL	IEEVJCL	IEEVJCLY
-	IEEVSTAR	IEEVSTAR	IEEVSTAR
IEE9803D	IEE9703D	MESSAGES	IEE9703D
-	IEE9803D	IEE9803D	IEE9803D
IEFAB416	IEFAB416	IEFAB416	IEFAB416
IEFAB417	IEFAB417	IEFAB417	IEFAB417
-	IEFAB420	IEFAB420	IEFAB420
IEFAB418	IEFAB418	IEFAB418	IEFAB418
-	IEFAB420	IEFAB420	IEFAB420
IEFBR14	IEFBR14	IEFBR14	IEFBR14
IEFDSDRP	IEFDSDRP (See Note 3)	IEFDSDRP	IEFDSDRP
IEFDSO	IEFDSOCP (See Note 3)	IEFDSOCP	IEFDSOCP
IEFDSOAL	IEFDSOAL	IEFDSOAL	IEFDSOAL
IEFDSOFB	IEFDSOFB	IEFDSOFB	IEFDSOFB
IEFDSOSM	IEFDSOSM (See Note 3)	IEFDSOSM	IEFDSOSM
IEFDSOWR	IEFDSOWR (See Note 3)	IEFDSOWR	IEFDSOWR
IEFIIC (See Note 3)			
IEFIRC	IEFVSPL	IEFVSPL	IEFVSPL

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	IEFVSCDQ	IEFVSCDQ	IEFVSCDQ
-	IEFVSD13	IEFSD090	IEFSD090
-	IEFVJA	IEFVJA	IEFVJA
-	IEFVINE	IEFVINE	IEFVINE
-	IEFVINC	IEFVINC	IEFVINC
-	IEFVINB	IEFVINB	IEFVINB
-	IEFVINA	IEFVINA	IEFVINA
-	IEFVH1	IEFVH1	IEFVH1
-	IEFVHQ	IEFVHQ	IEFVHQ
-	IEFVHN	IEFVHN	IEFVHN
-	IEFVFB	IEFVFB	IEFVFB
-	IEFVFA	IEFVFA	IEFVFA
-	IEFVEA	IEFVEA	IEFVEA
-	IEFVDBSD	IEFVDBSD	IEFVDBSD
-	IEFVDA	IEFVDA	IEFVDA
-	IEFSD536	IEFVHR	IEFVHR
-	IEFSD533	IEFIRC	IEFIRC
-	IEFVHM	IEFVHM	IEFVHM
-	IEFVHL	IEFVHL	IEFVHL
-	IEFVHH	IEFVHH	IEFVHH
-	IEFVHF	IEFVHF	IEFVHF
-	IEFVHEC	IEFVHEC	IEFVHEC
-	IEFVHEB	IEFVHEB	IEFVHEB
-	IEFVHE	IEFVHE	IEFVHE
-	IEFVHCB	IEFVHCB	IEFVHCB
-	IEFVHC	IEFVHC	IEFVHC
-	IEFVHA	IEFVHA	IEFVHA
-	IEFVGM	IEFVGM	IEFVGM
-	IEFVGK	IEFVGK	IEFVGK
-	IEFVGI	IEFVGI	IEFVGI
-	IEFVGT	IEFVGT	IEFVGT
-	IEFVGS	IEFVGS	IEFVGS
IEFJES	IEFUSO	IEFUSO	IEFUSO
-	IEFSMGET	IEFSMGET	IEFSMGET
-	IEFSMEND	IEFENDS	IEFSMEND
-	IEFSMCLD	IEFCLOD	IEFSMCLD
-	IEFWAALC	IEFWAALC	IEFWAALC
-	IEFSMREP	IEFSMREP	IEFSMREP
-	IEFSMPUT	IEFSMPUT	IEFSMPUT
-	IEFSMODS	IEFSMODS	IEFSMODS
-	IEFSMIFC	IEFSMIFC	IEFSMIFC
-	IEFBMPUT	IEFBMPUT	IEFBMPUT
-	IEFBMPUR	IEFBMPUR	IEFBMPUR
-	IEFBMGET	IEFBMGET	IEFBMGET
-	IEFQMJ01	IEFJESQM	IEFJESQM
-	-	IEFQASNM	IEFQASNM
-	IEFQMJ02	IEFQMJ02	IEFQMJ02
-	IEFQMJ03	IEFQMJ03	IEFQMJ03
-	IEFQMMAC	IEFQMMAC	IEFQMMAC
-	IEFWAA01	IEFWAA01	IEFWAA01
-	IEFWAA02	IEFWAA02	IEFWAA02
-	IEFWAA03	IEFWAA03	IEFWAA03

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	IEFWAA04	IEFWAA04	IEFWAA04
-	IEFWAD	IEFWAD	IEFWAD
-	IEFWAMGR	IEFWAMGR	IEFWAMGR
-	IEFUJP	IEFUJP	IEFUJP
-	IEFUIV	IEFUIV	IEFUIV
-	IEFOSC08	IEFOSC08	IEFOSC08
-	IEFMSGJP	IEFMSGJP	IEFMSGJP
-	IEFOSC02	IEFOSC02	IEFOSC02
-	IEFOSC04	IEFOSC04	IEFOSC04
-	IEFOSC03	IEFOSC3B	IEFOSC03
-	-	IEFOSC3A	-
-	-	IEFOSC03	-
-	IEFOSC01	IEFOSC01	IEFOSC01
-	IEFVMD	IEFVMD	IEFVMD
-	IEFOSC05	IEFOSC5A	IEFOSC05
-	-	IEFOSC05	IEFOSC05
-	IEFOSC07	IEFOSC07	IEFOSC07
-	IEFOSC06	IEFOSC06	IEFOSC06
-	IEFVMA	VMA085	IEFVMA
-	-	VMA103	-
-	-	VMAPOST	-
-	-	IEFVMA	-
-	IEFVMB	IEFVMB	IEFVMB
-	IEFVMC	IEFVMC	IEFVMC
-	IEFACTLK	IEFACTLK	IEFACTLK
-	IEFACTFK	IEFACTFK	IEFACTFK
IEFMCVOL	IEFYSVMS	IEFYSVMS	IEFYSVMS
-	IEFMCVOL	IEFCVOL1	IEFCVOL1
-	-	-	IEFCVOL2
-	-	-	IEFCVOL3
-	IEFVMFAK	IEFVMCVL	IEFVMCVL
-	IEFVMMS1	IEFVM1	IEFVM1
-	IEF41FAK	IEFW41SD	IEFW41SD
-	IEFVMLS6	IEFVM6	IEFVM6
-	IEFVMLS7 (See Note 3)	IEFVM7	IEFVM7
IEFPRINT	IEFPRTXX	SPRINTER	SPRINTER
IEFQMIFC (See Note 3)			
IEFRSTRT	IEFRSTRT	IEFRSTRT	IEFRSTRT
IEFSD095	IEFSD095	IEFSD095	IEFSD095
IEFSD160	IEFSD160	IEFSD060	IEFSD060
-	IEF161FK	IEFSD061	IEFSD061
IEFSD161	IEFSD161	IEFSD061	IEFSD061
-	IEFSD165	IEFSD065	IEFSD065
-	IEFSD164	IEFSD064	IEFSD064
-	IEFSD161	IEFIRET	IEFSD061
-	IEFSDPPT	AUTHSEC	IEFSDPPT
-	-	IEFSDPPT	-
-	-	LNGRNSEC	-
-	-	NOCANSEC	-
-	IEFMF102	IEFSD102	IEFSD102
-	IEFDSLST	IEFDSLST	IEFDSLST

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	IEFDSTBL	IEFDSTBL	IEFDSTBL
-	IEFSD101	IEFSD062	IEFSD062
-	IEFSD515	GO	IEFSD515
-	IEFSTDSC	IEFSTDSC	IEFSTDSC
-	IEFSD168	IEFSD068	IEFSD068
-	IEFSD166	IEFSDA66	IEFSD066
-	-	IEFSD066	-
-	IEFYPMSG	YPPMSG4	IEFYPMSG
-	-	YPPMSG3	-
-	IEFYPJB3	IEFYP	IEFYP
-	IEFIDMPM	IEFIDMPM	IEFIDMPM
-	IEFYNMSG	IEFYNMSG	IEFYNMSG
-	IEFYNIMP	IEFYN	WTERN024
-	-	-	IEFYN
-	IEFSD42Q	IEFW42SD	IEFW42SD
-	IEFSD31Q	IEFW31SD	IEFW31SD
-	IEFSD22Q	IEFW22SD	IEFW22SD
-	IEFRPREP	IEFRPREP	IEFRPREP
-	IEFIDUMP	IEFIDUMP	IEFIDUMP
-	IEFVJMSG	IEFVJMSG	IEFVJMSG
-	IEFWTERM	IEFWTERM	IEFWTERM
-	IEFVJIMP	IEFVJ	WTERM030
-	-	-	IEFVJ
-	IEFZHMSG	IEFZH	IEFZH
-	IEFZGST2	IEFZG2	IEFZG2
-	IEFZGST1	IEFZG	IEFZG
-	IEFZGMSG	IEFZGMSG	IEFZGMSG
-	IEFZGJB1	IEFZGJ	IEFZGJ
-	IEFZAJB3	IEFZA	IEFZA
-	IEFYTVMS	IEFYT	IEFYT
-	IEFYSVMS	IEFYSVMS	IEFYSVMS
-	IEFACTFK	IEFACTFK	IEFACTFK
-	IEFSMFLK	IEFACTLK	IEFACTLK
-	IEFWAD	IEFWAD	IEFWAD
-	IEFACTRT	IEFACTRT	IEFACTRT
-	IEFACTLK	IEFACTLK	IEFACTLK
-	IEFSMFWI	IEFSMFWI	IEFSMFWI
-	IEF065FK	IEFSD065	IEFSD065
-	IEF160FK	IEFSD060	IEFSD060
-	IEFSD598 (See Note 3)	IEFSD598	IEFSD598
IEFSD162	IEFUJI	IEFUJI	IEFUJI
-	IEFSMFIE	IEFSMFIE	IEFSMFIE
-	IEFUSI	IEFUSI	IEFUSI
-	IEF263FK	IEFSD063	IEFSD063
-	IEFSD162	IEFSD162	IEFSD062
-	-	IEFALRET	-
-	-	IEFSD62A	-
-	IEF065FK (See Note 3)	IEFSD065	IEFSD065
IEFSD263	IEFMF263	IEFSD263	IEFSD263
-	IEFSMFAT (See Note 3)	IEFSMFAT	IEFSMFAT

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
IEFSD300	IEFSD300	IEFSD300	IEFSD300
-	-	IEFSD306	IEFSD306
-	IEFSD301	IEFSD301	IEFSD301
-	IEFSD302	IEFSD302	IEFSD302
-	IEFSD303	IEFSD303	IEFSD303
-	IEFSD309	IEFSD309	IEFSD309
IEFSD304	IEFSD304	IEFSD304	IEFSD304
IEFSD510	IEFMF106	IEFMF106	IEFMF106
-	IEFMF105	IEFSD105	IEFSD105
-	IEFSD510 (See Note 3)	IEFSD510	IEFSD510
IEFSD518	IEFSD518 (See Note 3)	IEFSD518	IEFSD518
IEFSD519	IEFSD519 (See Note 3)	IEFSD519	IEFSD519
IEFSD526	IEFAB400	IEFAB400	IEFAB400
-	IEFAB401	IEFAB401	IEFAB401
-	IEFAB402	IEFAB402	IEFAB402
-	IEFYSVMS	IEFYSVMS	IEFYSVMS
-	IEFSD097	IEFSD097	IEFSD097
-	IEFSD41Q	IEFW41SD	IEFW41SD
-	IEFSD195	IEFSD095	IEFSD095
-	IEFSD096	IEFSD096	IEFSD096
-	IEFXTDMY	IEFXTDMY	IEFXTDMY
-	IEFXT00D	IEFXT000	IEFXT000
-	IEFXT003	IEFXT003	IEFXT003
-	IEFXT002	IEFXT002	IEFXT002
-	IEFXDPTH	IEFXDPTH	IEFXDPTH
-	IEFX5000	IEFX5000	IEFX5000
-	IEFX300A	IEFX3000	IEFX3000
-	IEFWEXTA	IEFWEXTA	IEFWEXTA
-	IEFWD000	IEFWD000	IEFWDMSG
-	-	-	IEFWD002
-	-	-	IEFWD000
-	IEFCVFAK	IEFCVOL3	IEFCVOL1
-	IEFVMMS1	IEFVM1	IEFVM1
-	IEFSD551	IEFV15XL	IEFV15XL
-	IEFSD552	IEFXJX5A	IEFXJX5A
-	IEFXKFAK	IEFXK000	IEFXK000
-	IEFWCIMP	IEFWC000	IEFWC002
-	-	-	IEFWC000
-	IEFXTMSG	IEFXTMSG	IEFXTMSG
-	IEFXH000	IEFXH000	IEFXH000
IEFSD569	-	IEFSD569	IEFSD569
-	IEEMB800	IEEMB800	IEEMB800
-	IEEVLIN	IEEVLIN	IEEVLIN
-	IEE0603D	IEE0603D	IEE0603D
-	IEFAB404	IEFAB404	IEFAB404
-	IEFBMINT	IEFBMINT	IEFBMINT
-	IEFPARMG	IEFPARMG	IEFPARMG
-	IEFPARMS	IEFPARMS	IEFPARMS

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	IEFPRES	IEFPRES	IEFPRES
-	IEFSMINT	IEFSMINT	IEFSMINT
IEFSQINT	IEFORMAT	IEFORMAT	IEFORMAT
-	IEFSD055	IEFSD055	IEFSD055
-	IEF300SD	IEFSD300	IEFSD300
-	IEF304SD	IEFSD304	IEFSD304
-	-	IEFQMSGV	IEFQMSGV
IEFVM6LS	IEFVMLS6	IEFVM6	IEFVM6
-	IEFXKIMP	IEFXK000	IEFXK000
-	IEFXJIMP	IEFXJ000	IEFXJ000
-	IEFSD096	IEFSD096	IEFSD096
-	IEFXKMSG	IEFXKMSG	IEFXKMSG
-	IEFCVFAK	IEFCVOL2	IEFCVOL1
-	IEFVMMS1	IEFVM1	IEFVM1
-	IEFXAFAK	IEFXA	IEFXA
-	IEFSD41Q	IEFW41SD	IEFW41SD
-	IEFSD195	IEFSD095	IEFSD095
-	IEFVMLS7	IEFVM7	IEFVM7
-	IEFXJMSG	IEFXJMSG	IEFXJMSG
-	IEFYSVMS (See Note 3)	IEFYSVMS	IEFYSVMS
IEFVRRC	IEFVRRC	IEFVRRC	IEFVRRC
-	-	IEFVRRCA	-
-	-	IEFVRRCB	-
-	IEFSD304	IEFSD304	IEFSD304
-	- (See Note 3)	IEFSD307	IEFSD307
IEFVRR1	IEFVRR1 (See Note 3)	IEFVRR1	IEFVRR1
-	IEFSD304	IEFSD304	IEFSD304
-	-	IEFSD307	IEFSD307
IEFVRR2	IEFVRR2	IEFVRR2	IEFVRR2
IEFVRR3	IEFSD309	IEFVRR3	IEFVRR3
-	-	IEFV3AE	-
-	IEFSD304	IEFSD304	IEFSD304
-	- (See Note 3)	IEFSD307	IEFSD307
IEFWAMAP	IEFWAMAP	IEFWAMAP	IEFWAMAP
IEFWAMIN	IEFWAMIN	IEFWAMIN	IEFWAMIN
-	IEFWAMSG	IEFWAMSG	IEFWAMSG
IEFWARIN	IEFWARIN	IEFWARIN	IEFWARIN
IEFWTO00	IEFWTP00	IGC0203E	IGC0203E
IEFWTP00	-	-	-
-	IEFACTLK	IEFACTLK	IEFACTLK
IEFW21SD	IEFAB405	IEFAB405	IEFAB405
-	IEFAB406	IEFAB406	IEFAB406
-	IEFAB407	IEFAB407	IEFAB407
-	IEFAB408	IEFAB408	IEFAB408
-	IEFACTFK	IEFACTFK	IEFACTFK
-	IEFCVFAK	IEFCVOL1	IEFCVOL1
-	IEFVMLK5	IEFVM6	IEFVM6

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	IEFSD551	IEFV15XL	IEFV15XL
-	IEFWDFAK	IEFWD000	IEFWD000
-	IEF41FAK	IEFW41SD	IEFW41SD
-	IEFXJFAK	IEFXJ000	IEFXJ000
-	IEFAVFAK	IEFXV001	IEFXV001
-	IEFX5FAK	IEFX5000	IEFX5000
-	IEFYSVMS	IEFYSVMS	IEFYSVMS
-	IEFSD180	IEFSD18	IEFSD180
-	IEFSD21Q	IEFW21SD	IEFW21SD
-	IEFSCAN	IEFSCAN	IEFSCAN
-	IEFXAMSG	IEFXAMSG	IEFXAMSG
-	IEFVMLS1	IEFVM	IEFVM1
-	IEFVM76	IEFVM76	IEFVM76
-	IEFVM5LS	IEFVM5	IEFVM5
-	IEFWA000	IEFWA000	IEFUCBL
-	-	-	IEFWA002
-	-	-	IEFWA7
-	IEFVM4LS	IEFVM4	IEFVM4
-	IEFVM3LS	IEFVM3	IEFVM3
-	IEFVM2LS	IEFVM2	IEFVM2
-	IEFVKIMP	IEFVK	IEFVK
-	IEFX300A	IEFX3000	IEFX3000
-	IEFWSWIN	IEFWSWIT	IEFWSWIT
-	IEFXDPTH	IEFXDPTH	IEFXDPTH
-	IEFXCSSS	IEFXA	IEFXA
-	IEFVKMSG	IEFVKMJ1	IEFVKMSG
-	IEFWSTRT	WTERM050	IEFWSTRT
-	-	WTERM040	-
-	IEFACTRT	IEFACTRT	IEFACTRT
-	IEFWAD (See Note 3)	IEFWAD	IEFWAD
IEFXH000	IEFXH000	IEFXH000	IEFXH000
IEFXVAVR	IEFCVFAK	IEFCVOL3	IEFCVOL1
-	IEFSD551	IEFV15XL	IEFV15XL
-	IEFWCFAK	IEFWC000	IEFWC000
-	IEFWD001	IEFWD001	IEFWD001
-	IEFSD552	IEFXJX5A	IEFXJX5A
-	IEFX5000	IEFX5000	IEFX5000
-	IEFX300A	IEFX3000	IEFX3000
-	IEFWD000	IEFWD000	IEFWDMSG
-	-	-	IEFWD002
-	-	-	IEFWD000
-	IEFXDPTH	IEFXDPTH	IEFXDPTH
-	IEFXV002	IEFXV002	IEFXV002
-	IEFXV001	IEFXV001	IEFXV001
-	IEFXVMSG	IEFXVMSG	IEFXVMSG
-	IEFXVNSL	IEFXVNSL	IEFXVNSL
-	IEFSCAN	IEFSCAN	IEFSCAN
-	IEFXV003	IEFXV003	IEFXV003
IEFQMIFC (See Note 3)			
IEF160SD	IEFINTQA	IEFINTQS	IEFINTQS
-	-	IEFINTQM	IEFINTQM

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
-	-	INTQMSG1	-
-	-	INTQMSG2	-
-	IEF161DM	IEFSD060	IEFSD060
IEF161SD	IEFSD161	IEFIRET	IEFSD061
-	-	IEFSD061	-
-	IEFSD160	IEFSD060	IEFSD060
IEF160DM	-	-	-
IEFQMIFC (See Note 3)			
IGCI107B	IEECVOCC	IEECVOC	IEECVOC
IGCU103D	IGCU103D	IEEUNIT1	IEEUNIT1
IGCU203D	IGCU203D	IEEUNIT2	IEEUNIT2
IGCU303D	IGCU303D	IEEUNIT3	IEEUNIT3
IGCU403D	IGCU403D	IEEUNIT4	IEEUNIT4
IGCXE03D	IEEXEDNA	IEEXEDNA	IEEXEDNA
IGCXL07B	IEECLCTX	IEECLCTX	IEECLCTX
IGCXN07B	IEECNCTX	IEECNCTX	IEECNCTX
IGCXO07B	IEECOCTX	IEECOCTX	IEECOCTX
IGC0003D	IEE0303D	IEE0303D	IEE0303D
IGC0003E	IEEMFWTO	IGC0003E	IGC0003E
IGC0003F	IEE0303F	IEE0303F	IEE0303F
IGC0007B	IEECMCTR	IEECVCTR	IEECVCTR
IGC0008C	IEESMF8C	IGC0083	IGC0083
IGC00110	IEE00110	IEE00110	IEE00110
IGC0008G	IEECXDOM	IGC0008G	IGC0008G
IGC0103E	IEEVWTOR	IGC0103E	IGC0103E
IGC0107B	IEECMPMX	IEECVPM	IEECVPM
-	IEECVOCX	IEECVOC	IEECVOC
IGC0108C	IEESMFOP	IEESMFOP	IEESMFOP
IGC0207B	IEECMPM1	IEECMPM1	IEECMPM1
IGC0208C	IEESMFAL	IEESMFAL	IEESMFAL
IGC0503E	IEEVROUT	IGC0503E	IGC0503E
IGC0603E	IEECVML3	IEECVML3	IEECVML3
IGC0703E	IEECVML5	IEECVML5	IEECVML5
IGC0803E	IEECVML6	IEECVML6	IEECVML6
IGC0903E	IEECVML7	IEECVML7	IEECVML7
IGC0907B	I3EECMWTL	IGC0907B	IGC0907B
IGC1B03D	IEE1B03D	IEE1B03D	IEE1B03D
IGC10110	IEE10110	IEE10110	IEE10110
IGC1103D	IEE1103D	IEE1103D	IEE1103D
-	-	IEE3103D	-
IGC1107B	IEECMPMC	IEECVPM	IEECVPM
IGC11110	IEE11110	IEE11110	IEE11110
IGC1203D	IEE1A03D	IEE1A03D	IEE1A03D
IGC12110	IEE12110	IEE12110	IEE12110
IGC1803D	IEESD571	IGC1803D	IGC1803D
IGC1903D	IEE1903D	IEE1903D	IEE1903D
IGC20110	IEE20110	IEE20110	IEE20110
IGC2107B	IEECMOCP	IEECVOC	IEECVOC
-	IEECMPMP	IEECVPM	IEECVPM
IGC21110	IEE21110	IEE21110	IEE21110
IGC22110	IEE22110	IEE22110	IEE22110

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
IGC2303D	IEE2303D	IEE2303D	IEE2303D
IGC23110	IEE23110	IEE23110	IEE23110
IGC2903D	IEE2903D	IEE2903D	IEE2903D
IGC3203D	IEE3203D	IEE3203D	IEE3203D
IGC3303D	IEE3303D	IEE3303D	IEE3303D
IGC3503D	IEE3503D	IEE3503D	IEE3503D
IGC3703D	IEE3703D	IEE3703D	IEE3703D
IGC3803D	IEE3803D	IEE3803D	IEE3803D
IGC40110	IEE40110	IEE40110	IEE40110
IGC4303D	IEE4303D	IEE4303D	IEE4303D
IGC4403D	IEE4403D	IEE4203D	IEE4403D
-	-	IEE4403D	-
IGC4503D	IEE4503D	IEE4503D	IEE4503D
IGC4603D	IEE4603D	IEE4603D	IEE4603D
IGC4703D	IEE4703D	IEE4703D	IEE4703D
IGC4903D	IEE4903D	IEE4903D	IEE4903D
IGC5A07B	IGC5A07B	IEECVETA	IEECVETA
IGC5C07B	IGC5C07B	IEECVETC	IEECVETC
IGC5D07B	IGC5D07B	IEECVETD	IEECVETD
IGC5E07B	IGC5E07B	IEECVETE	IEECVETE
IGC5F07B	IGC5F07B	IEECVETF	IEECVETF
IGC5G07B	IGC5G07B	IEECVETG	IEECVETG
IGC5H07B	IGC5H07B	IEECVETH	IEECVETH
IGC5J07B	IGC5J07B	IEECVETJ	IEECVETJ
IGC5K07B	IGC5K07B	IEECVETK	IEECVETK
IGC5L07B	IGC5L07B	IEECVETL	IEECVETL
IGC5M07B	IGC5M07B	IEECVETM	IEECVETM
IGC5N07B	IGC5N07B	IEECVETN	IEECVETN
IGC5O07B	IGC5O07B	IEECVETO	IEECVETO
IGC5P07B	IGC5P07B	IEECVETP	IEECVETP
IGC5Q07B	IGC5Q07B	IEECVETQ	IEECVETQ
IGC5R07B	IGC5R07B	IEECVETR	IEECVETR
IGC5S07B	IGC5S07B	IEECVETS	IEECVETS
IGC5T07B	IGC5T07B	IEECVETT	IEECVETT
IGC5107B	IGC5107B	IEECVET1	IEECVET1
IGC5207B	IGC5207B	IEECVET2	IEECVET2
IGC5307B	IGC5307B	IEECVET3	IEECVET3
IGC5407B	IGC5407B	IEECVET4	IEECVET4
IGC5603D	IEE5603D	IEE5603D	IEE5603D
IGC5607B	IGC5607B	IEECVET6	IEECVET6
IGC5707B	IGC5707B	IEECVET7	IEECVET7
IGC5807B	IGC5807B	IEECVET8	IEECVET8
IGC5903D	IEE5903D	IEE5903D	IEE5903D
IGC5907B	IGC5907B	IEECVET9	IEECVET9
IGC60110	IEE60110	IEE60110	IEE60110
IGC6303D	IEE6303D	IEE6303D	IEE6303D
IGC6403D	IEE6403D	IEE6403D	IEE6403D
IGC6503D	IEE6503D	IEE6503D	IEE6503D
IGC6603D	IEE6603D	IEE6603D	IEE6603D

Cross-Reference Directory Alphabetical List — By Load Module

Load Module	Module Name	Entry Point	CSECT
IGC6703D	IEE6703D	IEE6703D	IEE6703D
IGC6803D	IEE6803D	IEE6803D	IEE6803D
IGC6903D	IEE6903D	IEE6903D	IEE6903D
IGC7203D	IEE7203D	IEE7203D	IEE7203D
IGC7303D	IEE7303D	IEE7303D	IEE7303D
IGC7503D	IEE7503D	IEE7503D	IEE7503D
IGC7603D	IEE7603D	IEE7603D	IEE7603D
IGC7703D	IEE7703D	IEE7703D	IEE7703D
IGC7803D	IEE7803D	IEE7803D	IEE7803D
IGC7903D	IEE7903D	IEE7903D	IEE7903D
IGC8703D	IEF8703D	IGC8703	IGC8703
IGC8803D	IEF8803D	-	-
IGC8903D	IEF8903D	-	-
IGC9903D	IEE9903D	IEE9903D	IEE9903D
IGF2403D	IGF2403D	IGF2403D	IGF2403D
IGF2603D	IGF2603D	IGFVMCD4	IGFVMCD4
IGG019DF	IGG019DF	IGG019DF	IGG019DF
IGG019DG	IGG019DG	IGG019DG	IGG019DG
IGG019DH	IGG019DH	IGG019DH+8	IGG019DH
-	-	IGG019DH+4	-
-	-	IGG019DH	-
IGG019DL	IGG019DL	IGG019DL	IGG019DL
IGG019DM	IGG019DM	IGG019DM	IGG019DM
IGG019FM	IGG019FM	IGG019FM	IGG019FM
IGG019MA	IGG019MA	IGG019MA	IGG019MA
IGG019MB	IGG019MB	IGG019MB	IGG019MB
IGG019MR	IGG019MR	IGG019MR	IGG019MR
IGG0196U	IGG0196U	IGG0196U	IGG0196U
IGG0196V	IGG0196V	IGG0196V	IGG0196V
IGG0196W	IGG0196W	IGG0196W	IGG0196W
IGG0196X	IGG0196X	IGG0196X	IGG0196X
IGG0196Y	IGG0196Y	IGG0196Y	IGG0196Y
IGG0196Z	IGG0196Z	IGG0196Z	IGG0196Z
IGG0201M	IGG0201M	IGG0201M	IGG0201M
IGG0201N	IGG0201N	IGG0201N	IGG0201N
IGXM07B	IEECMCTX	IEECMCTX	IEECMCTX
IGX00000	IEEDFIN1	IEEDFIN1	IEEDFIN1
IGX00001	IEEDFIN4	IEEDFIN4	IEEDFIN4
IGX00002	IEEDFIN5	IEEDFIN5	IEEDFIN5
IGX00003	IEEDFINA	IEEDFINA	IEEDFINA
IGX00004	IEESD561	IEESD561	IEESD561
IGX00011	IEE1403D	IGC1403D	IGC1403D
IGX00012	IEESD566	IEESD566	IEESD566
IGX01000	IEEDFIN2	IEEDFIN2	IEEDFIN2
IGX01004	IEESD562	IEESD562	IEESD562
IGX02000	IEEDFIN3	IEEDFIN3	IEEDFIN3
IGX02004	IEESD563	IEESD563	IEESD563
IGX03000	IEEDFIN6	IEEDFIN6	IEEDFIN6
IGX03004	IEESD564	IEESD564	IEESD564
IGX04000	IEEDFIN7	IEEDFIN7	IEEDFIN7
IGX04004	IEESD565	IEESD565	IEESD565

Load Module	Module Name	Entry Point	CSECT
IGX05004	IEESD575	IEESD575	IEESD575
IGX06004	IEESD576	IEESD576	IEESD576

Note 3: The following load modules all contain module IEFQMIFC:

IEFDSDRP
IEFDSO
IEFDSOSM
IEFDSOWR
IEFIIC
IEFMCVOL
IEFQMIFC
IEFSD161
IEFSD162
IEFSD300
IEFSD304
IEFSD510
IEFSD518
IEFSD519
IEFSD526
IEFVM6LS
IEFVRRC
IEFVRR1
IEFVRR3
IEFW21SD
IEFXVAVR
IEF161SD

The entry points in module IEFQMIFC are:

CNVRT
IEFCNVRT
IEFLOCDQ
IEFQAGST
IEFQASGN
IEFQDELE
IEFQMDQ2
IEFQMIFC
IEFQMNQ2
IEFQMRAW
IEFQMSSS
IEFQMUNC
IEFRDWRT
IEFSD514
LOC
LOCCAN
LOCDQ
WRT
RD

Module/Figure Relationship

Module Name	Figure Number Section 3-	Function
IEEAAD00	67	SVC dump
IEECIR51	56, 61, 64, 65, 69, 79, 81	master partition resident wait
IEECLCTX	114	console switch (1)
IEECMAWR	112-116, 120, 121, 130	COMTASK router
IEECMCTR	112-116, 120, 121, 130	mini-router
IEECMCTX	114	console switch (2)
IEECMDOM	130	DOM processor
IEECMDSV	113, 116, 120, 121, 130	device service
IEECMPMX	113, 116, 120	printer/keyboard processor
IEECMPM1	113	printer/keyboard processor - MLWTOs
IEECMWSV	116	WTO WQE processor
IEECMWTL	112	WTL processor
IEECNCTX	114	console switch (3)
IEECOCTX	114	console switch (4)
IEECVCRA	113	attention interrupt handler
IEECVCRX	114	external interrupt handler
IEECVML3	117	MLWTO processing
IEECVML5	117	MLWTO processing
IEECVML6	117	MLWTO processing
IEECVML7	117	MLWTO processing
IEECVOCX	113	open routine
IEECXDOM	130	DOM handler
IEEDFINA	56, 61	SMF record creation
IEEDFIN1	56, 61	DEFINE initialization
IEEDFIN2	56, 61	syntax check routine
IEEDFIN3	56, 61	validity check routine
IEEDFIN4	56, 61	partition control block build
IEEDFIN5	56, 61	partition format build
IEEDFIN6	56,61	DEFINE message module
IEEDFIN7	61	DEFINE list processing
IEEDFIN8	56	DEFINE member processing
IEELGON	73	LOGON command processor
IEELGON1	73	LOGON command processor
IEELGON2	73	LOGON command processor
IEELIST	70	LISTBC command processor
IEELIST1	70	LISTBC command processor
IEELOGWR	102, 103	system log resident wait router routine
IEEMFWTO	116	WTO processor
IEEPSN	40	scan routine
IEEQID	56	QID initialization
IEERTE	82	ROUTE command processor
IEERTE1	82	ROUTE syntax scan
IEERTE2	82	ROUTE update route table
IEERTE3	82	ROUTE print messages
IEESD561	59, 64	queue command processor
IEESD562	59	syntax check

Module Name	Figure Number Section 3-	Function
IEESD563	59	queue search routine
IEESD566	76	MONITOR ACTIVE command processor
IEESD571	61	DEFINE command processor
IEESD575	59	CANCEL job on queue
IEESD576	59	CANCEL restarted job
IEESMFAL	56	SMF allocation
IEESMFIT	56	SMF initialization
IEESMFI2	56	SMF record construction
IEESMFI3	56	validity check routine
IEESMFOI	56	SMF OPEN initialization
IEESMFOP	56, 134, 136	SMF OPEN routine
IEESMFWT	56, 135, 136	SMF writer
IEESMF8C	56, 134, 135, 136	SMF buffer manager
IEEVCTI	56, 112	COMTASK initialization
IEEVLIN	112	system log initialization
IEEVMNT1	12	MOUNT command processor
IEEVMNT2	12	MOUNT task
IEEVRCTL	2, 3, 6, 9, 10-12	JES reader interface
IEEVROUT	119	RES interface
IEEVSEND	83-87	SEND command processor
IEEVSND1	85-87	SEND syntax checker
IEEVSND2	86	SEND notice handler
IEEVSND3	84	SEND list handler
IEEVSND4	83-87	SEND cleanup and exit routines
IEEVSND5	83-86	SEND input/output operations
IEEVSND8	83, 85	SEND notice handler
IEEVSND9	83-86	SEND input/output operations
IEEVSTAR	2, 3, 9, 11	START command processor
IEEVWTOR	118	WTOR processing
IEEXEDNA	63	DISPLAY consoles command processor
IEE0303F	71, 102, 103, 108	system log transient writer
IEE0403D	41	SVC 34 command router
IEE0403F	102, 103	system log data set processor
IEE0503D	104, 105	message assembly
IEE0603D	88	SET command processor
IEE1A03D	80	REPLY command processor
IEE1B03D	80	REPLY message routine
IEE1103D	95-97	UNLOAD command processor
IEE1403D	68, 94	SWITCH command processor
IEE1603D	71, 102, 103	WRITELOG command processor
IEE1903D	77, 89	START command processor
IEE2303D	96, 97	VARY SMF record handler
IEE3303D	96, 100	syntax scan routine
IEE3503D	62-66, 69, 79, 81	DISPLAY command router
IEE3703D	58	CANCEL command processor
IEE3803D	66	DISPLAY USER processor
IEE4303D	98	VARY MSTCONS command
IEE4503D	75, 90, 91	STOP command processor
IEE4603D	96	online/offline
IEE4703D	101	VARY HARDCOPY processor
IEE4903D	100	CONSOLE operand processor
IEE60110	67	DUMP command processor
IEE6303D	78	MSGRT command processor

Module Name	Figure Number Section 3-	Function
IEE6403D	78	MSGRT command processor
IEE6503D	88	SET TOD processor
IEE6703D	60	CONTROL command handler
IEE6803D	60	CONTROL command handler
IEE6903D	60	CONTROL command handler
IEE7103D	76	MONITOR command processor
IEE7203D	101	VARY HARDCOPY operand processor
IEE7503D	60, 62, 63, 91	routing location routine
IEE7603D	62, 63, 76, 91	routing location routine
IEE7703D	60	CONTROL command handler
IEE7803D	60	CONTROL command handler
IEE7903D	67	command message module
IEE8503D	64	DISPLAY SQA command processor
IEE8703D	72	LOGOFF command processor
IEE8803D	63	DISPLAY RT command processor
IEE8903D	63	DISPLAY RT command processor
IEE9803D	56	SET spool parameter processor
IEE9903D	104	WRITER command processor
IEFAB400	29	load balance initialization
IEFAB401	29	load balance selection
IEFAB405	24	locate driver
IEFAB406	24	SIOT completion
IEFAB407	24	SIOT creation
IEFAB408	24	queue manager interface
IEFAB418	32	cleanup routine
IEFAB420	32	hook/unhook routine
IEFACTLK	32	accounting interface
IEFBMGET	50, 53	GET buffer
IEFBMINT	56	buffer manager initialization
IEFBMPUT	52	PUT buffer
IEFDSOAL	24	DSO determination
IEFDSOCP	9	DSO control program
IEFDSOSM	10	DSO stop modify
IEFDSOWR	5	DSO writer
IEFDSLST	7	ENQ list construction routine
IEFSDTBL	7	DSNAME tabe construction routine
IEFIIC	2	initiator IEL builder
IEFIDUMP	32	indicative dump
IEFMCVOL	24	mount control volume
IEFORMAT	35	queue format
IEFOSC01	43	printer main logic
IEFOSC02	43	punch/tape main logic
IEFOSC03	43	command processor
IEFOSC04	43	wait routine
IEFOSC05	43	job deletion routine
IEFOSC06	43	job separator
IEFOSC07	43	user written routine
IEFOSC08	43	spanned record processing
IEFPARMS	56	JES option selection
IEFPRES	56	set volume attributes routine
IEFRPREP	32	restart preparation
IEFSCAN	25, 26	UCB bit pattern scan
IEFSD055	35	queue initialization

Module Name	Figure Number Section 3-	Function
IEFSD097	28	wait for space decision
IEFSD101	8	program authorization
IEFSD160	2, 3, 9, 11, 12	initiator initialization
IEFSD161	2-7, 9-12	job selection
IEFSD162	2-5, 9, 11, 12	allocation interface
IEFSD180	24	dedication determination
IEFSD195	27, 28	wait for deallocation
IEFSD21Q	23	allocation entry
IEFSD22Q	32	step termination exit
IEFSD263	2-5, 9, 11, 12, 37	system task/problem program interface
IEFSD300	35	table construction
IEFSD302	35	table processing
IEFSD303	35	free track assign
IEFSD304	35, 37	open SWADS
IEFSD305	35	purge queue processing
IEFSD309	35	MCM-JCM update routine
IEFSD31Q	33	job termination exit
IEFSD41Q	23	allocation exit
IEFSD42Q	32, 35	termination entry
IEFSD510	2-6, 9-12	partition controller
IEFSD514	35	table breakup
IEFSD569	35, 36, 112	master scheduler initialization
IEFSMCLD	51	close open dictionary
IEFSMEND	52	end data set
IEFSMFSO	50	OUTLIM handler
IEFSMGET	49, 53	GET record
IEFSMIFC	35, 40, 41, 43, 47	JECS request router
IEFSMINT	56	spool manager initialization
IEFSMODS	48, 53	OPEN data set
IEFSMPUT	50	PUT record
IEFSMREP	53	reposition data set
IEFSQINT	56	queue manager initialization
IEFUIV	41	optional user exit
IEFUSO	50	user routine
IEFVDA	17, 19	DD statement processor
IEFVEA	16, 19	EXEC statement processor
IEFVFA	15-17, 19	scan routine
IEFVFB	19	symbolic parameter processing
IEFVHA	14-21	interpreter GET routine
IEFVHC	14-21	continuation check
IEFVHCB	14-21	verb identification
IEFVHE	15-19	router routine
IEFVHEB	15-17, 19	pre-scan routine
IEFVHEC	18	validity check
IEFVHF	15-17, 19	post-scan routine
IEFVHH	16, 18	job and job step enqueue
IEFVHL	18	NULL statement processor
IEFVHM	14	command processor
IEFVHN	18	interpreter termination
IEFVINA	20	in-stream procedure router
IEFVINB	20	in-stream procedure directory search
IEFVINC	20	procedure directory build
IEFVINE	20	syntax check routine

Module Name	Figure Number Section 3-	Function
IEFVJA	15	job processor
IEFVKIMP	23	EXEC statement checking
IEFVMA	37, 39	JEPS monitor/reader initialization
IEFVMB	40, 41	reader main logic
IEFVMC	40, 41	reader auxiliary logic
IEFVMD	39	writer initialization
IEFVMF	40	in-core reader interface
IEFVMLS1	23	JFCB housekeeping control
IEFVMLS6	24	message module
IEFVM2LS	24	fetch DCB processing
IEFVM3LS	24	generation data group single processing
IEFVM4LS	24	generation data group all processing
IEFVM5LS	24	patterning DSCB
IEFVM76	24	unique volume ID
IEFVRJCL	40	in-core reader
IEFVRRC	37	restart reader control
IEFVRR1	37	dequeue by job
IEFVRR2	37	table merge routine
IEFVRR3	37	reinterpretation delete/enqueue routine
IEFWAALC	43, 48, 50, 51	work area allocation router
IEFWAA01	48, 50	work area allocate
IEFWAA02	43, 50, 51	work area allocation I/O router
IEFWAMGR	48-53	work area manager
IEFWAMIN	56	work area manager initialization
IEFWARIN	56	work area manager initilization
IEFWA000	23	demand allocation
IEFWCIMP	23	TIOT construction
IEFWD000	23, 26	external action
IEFWEXTA	23	extended external action
IEFWSWIN	23	linkage module
IEFWTP00	109	write-to-programmer
IEFXCSSS	23	allocation control
IEFXDPTH	25, 27	data path calculator
IEFXH000	25, 27	separator strikeout
IEFXJIMP	27, 32	allocation recovery
IEFXKIMP	28	non-recovery error
IEFXTDMY	28	queue management error
IEFXT00D	23	space allocation
IEFXT002	23	TIOT compression
IEFXT003	28	DADSM error recovery
IEFXVNSL	26	AVR nonstandard label processing
IEFXV001	26	automatic volume recognition
IEFXV002	26	AVR label processing
IEFXV003	26	AVR tape processing
IEFX300A	25-27	device strikeout
IEFX5000	23	decision allocation
IEFYNIMP	32	step termination control
IEFYPJB3	32	step termination data set driver
IEFYTVMS	32	DSB processing
IEFZAJB3	32, 33	job termination control
IEFZGJB1	33	job disposition and deallocation
IEFZGST1	32	step termination disposition
IEFZGST2	32	step termination deallocation

Module Name	Figure Number Section 3-	Function
IEFZHMSG	32	disposition and deallocation message
IEF160SD	2	initiator initialization
IEF161SD	2	partition control interface
IFASMFDP	137	SMF dump utility
IFSPREIN	56	RTAM parameter table values
IGCU103D	64	UNIT syntax check routine
IGCU203D	64	DISPLAY line builder
IGCU303D	64	message module
IGCU403D	64	DISPLAY line writer
IGC5C07B	124	asynchronous error routine
IGC5D07B	129	message module
IGC5E07B	124	message module
IGC5F07B	127	light pen/cursor routine
IGC5G07B	122	OPEN/CLOSE routine
IGC5J07B	123	roll mode routine
IGC5K07B	123	timer interpreter
IGC5N07B	128	DCK routine
IGC5O07B	128	DCK routine
IGC5P07B	131	2250 I/O routine
IGC5Q07B	131	2250 I/O routine
IGC5R07B	132	2260 I/O routine
IGC5S07B	128	DCK routine
IGC5107B	122-130	DIDOCS processor
IGC5307B	123	DISPLAY routine
IGC5407B	127, 128	command routine
IGC5707B	130	delete routine
IGC5807B	129	delete routine
IGC5907B	123, 129	delete routine
IGF2403D	99	VARY PATH processor
IGF2503D	92, 93	SWAP command processor
IGF2603D	74	MODE command processor
IGG019DF	41	JAM GET module
IGG019DG	43	JAM PUT module
IGG019DH	41, 43	RCI GET/PUT intercept
IGG019DM	41	asynchronous exit routine
IGG019FM	44	DCB exit routine
IGG0196U	45	JAM OPEN executor
IGG0196V	45	JAM OPEN executor
IGG0196W	45	JAM OPEN executor
IGG0196X	45	JAM OPEN executor
IGG0196Y	45	OPEN executor
IGG0196Z	44	RCI OPEN intercept executor
IGG0197A	44	OPEN routines
IGG0201M	46	JAM CLOSE executor
IGG0201N	46	RCI CLOSE executor

Section 5: Data Areas

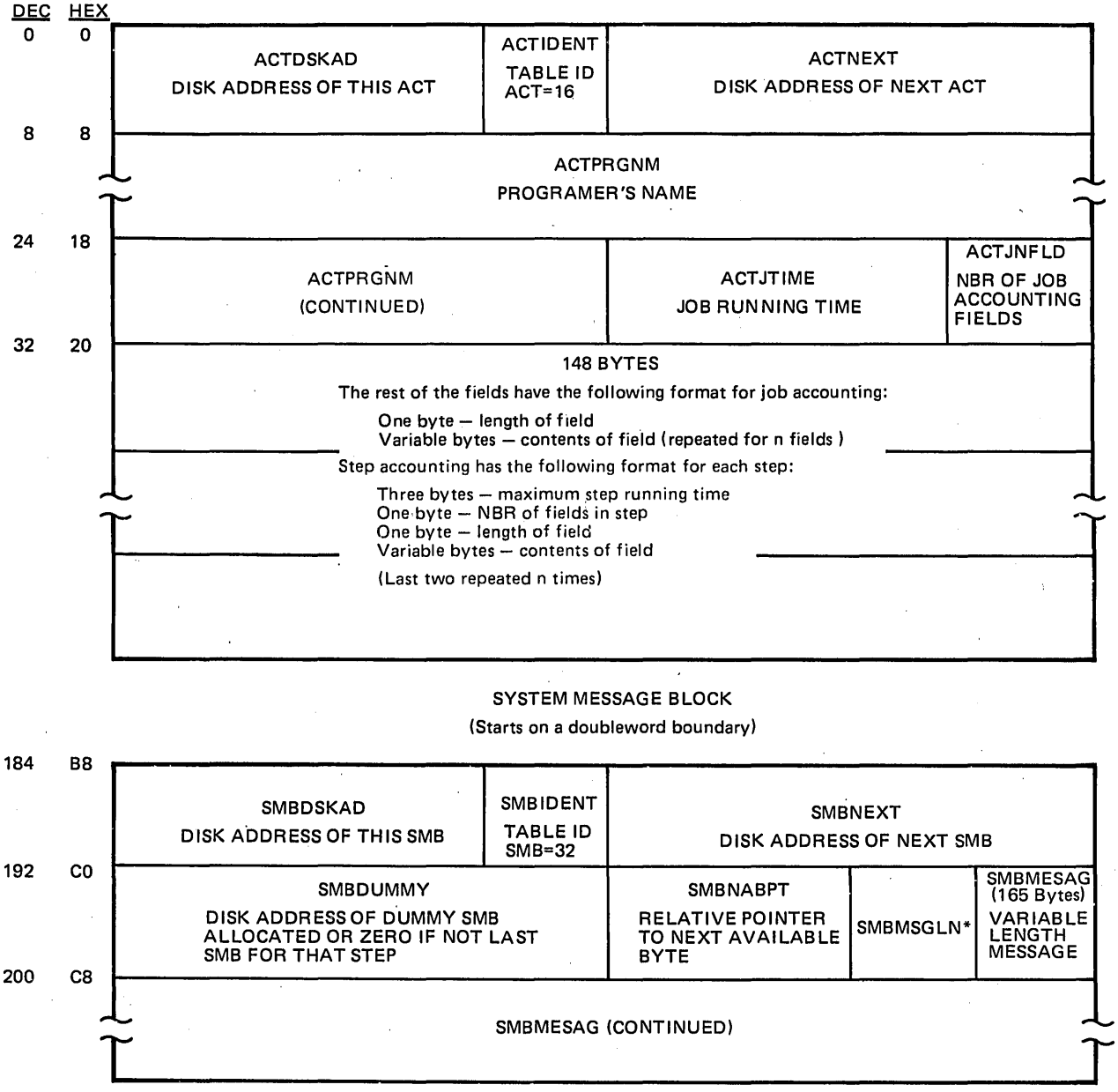
This section describes the data areas used only by the job management routines. All other data areas used in OS/VS1 are documented in the *OS/VS1 System Data Areas,* SY28-0605.

CONTENTS

Account Control Table (ACT)

The Account Control Table (ACT) is created by the interpreter. It contains accounting information relevant to the job. The ACT is deleted at job termination time.

ACT Storage Map



* Length of message, or zero if no more messages in the block, or each bit = 1 to indicate that a data set follows.

Fields in ACT — by displacement

DEC	HEX	FIELD
0	0	ACTDSKAD
3	3	ACTIDENT
4	4	ACTNEXT
8	8	ACTPRGNM
28	1C	ACTJTIME
31	1F	ACTJNFLD
184	B8	SMBDSKAD
187	BB	SMBIDENT
188	BC	SMBNEXT
192	C0	SMBDUMMY
196	C4	SMBNABPT
198	C6	SMBMSGLN
199	C7	SMBMESAG

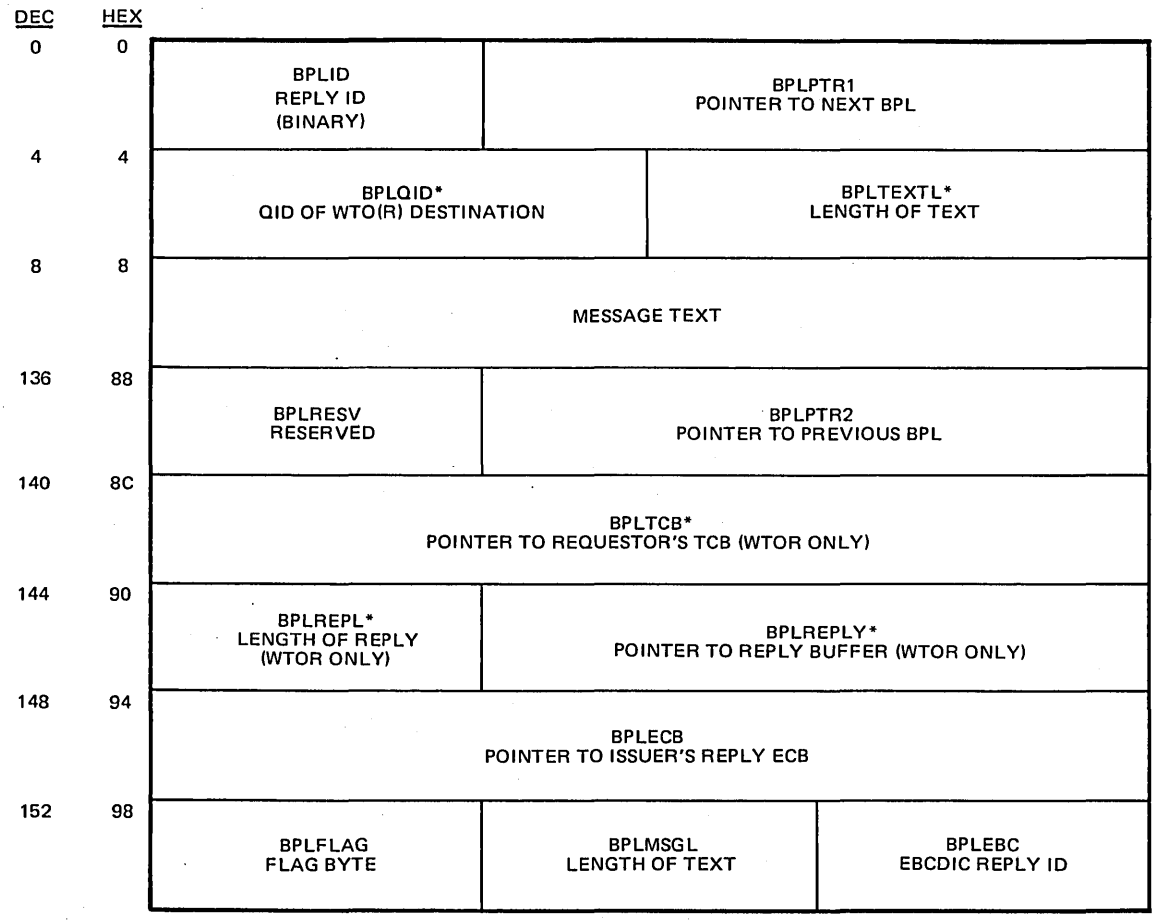
Alphabetical list of fields in ACT

FIELD	DEC	HEX
ACTDSKAD	0	0
ACTIDENT	3	3
ACTJNFLD	31	1F
ACTJTIME	28	1C
ACTNEXT	4	4
ACTPRGNM	8	8
SMBDSKAD	184	B8
SMBDUMMY	192	C0
SMBIDENT	187	B8
SMBMESAG	199	C7
SMBMSGLN	198	C6
SMBNABPT	196	C4
SMBNEXT	188	BC

Buffer Parameter List (BPL)

The WTO(R) Buffer Parameter List (BPL) provides RTAM with information necessary to handle WTO(R)s from remote terminals.

BPL Storage Map



* THIS INFORMATION IS PROVIDED BY THE COMMUNICATION TASK.

Fields in PLB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	BPLID	136	88	BPLRESV	148	94	BPLECB
1	1	BPLPTR1	137	89	BPLPTR2	152	98	BPLFLAG
4	4	BPLQID	140	8C	BPLTCB	153	99	BPLMSGL
6	6	BPLTEXTL	144	90	BPLREPL	154	9A	BPLEBC
8	8	MESSAGE TEXT	145	91	BPLREPLY			

Alphabetical list of fields in BPL

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
BPLEBC	154	9A	BPLPTR1	1	1	BPLREPLY	145	91
BPLECB	148	94	BPLPTR2	137	89	BPLRESV	136	88
BPLFLAG	152	98	BPLQID	4	4	BPLTCB	140	8C
BPLID	0	0	BPLREPL	144	90	BPLTEXTL	6	6
BPLMSGL	153	99						

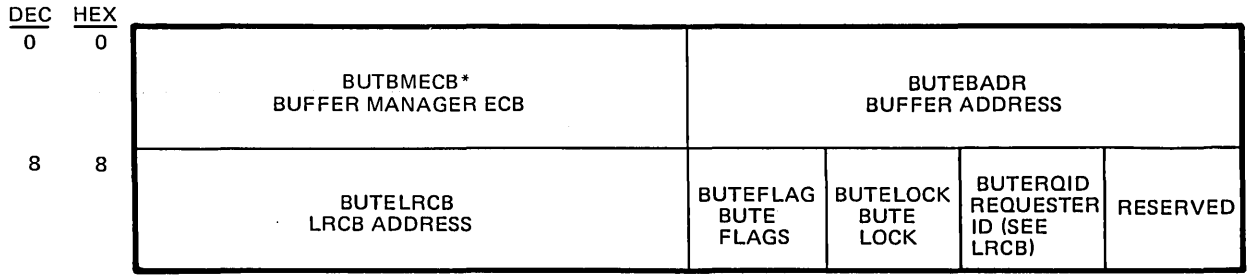
Flags and Masks

FLAG FIELD	CONTAINS	VALUE	MEANS
BPLFLAG	FLAGS	40	BPL TO BE FREED
		80	WTOR WAITING FOR A REPLY

Buffer Usage Table (BUTBL)

The Buffer Usage Table (BUTBL) contains the status of buffers. It shows if a buffer is assigned and has an address of LRCB to which the buffer was assigned. IEFBMINT obtains the space. The BUTBL, which is never freed, occupies 20 bytes of storage.

BUTBL Storage Map



Fields in BUTBL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	BUTBMECB	8	8	BUTELRCB	13	D	BUTELOCK
4	4	BUTEBADR	12	C	BUTEFLAG	14	E	BUTERQID

Alphabetical list of fields in BUTBL

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
BUTBMECB	0	0	*BUTEFLAG	12	C	BUTELRCB	8	8
BUTEBADR	4	4	*BUTELOCK	13	D	BUTERQID	14	E

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
BUTEFLAG	BUTE FLAGS	BUTEASGN	80	BIT0-BUFFER ASSIGNED FLAG
		BUTEMFD	40	BIT1-BUFFER MODIFIED FLAG
				BITS2-7-RESERVED
BUTELOCK	BUTE LOCK	BUTELK0	80	BIT0-BUFFER LOCK BIT

*ONLY THE FIRST ENTRY CONTAINS BUTBMECB (BUFFER MANAGER ECB).
EASH SUCCESSIVE ENTRY IS ONLY THREE WORDS IN LENGTH.

Checkpoint Master Cylinder Map (CMCM)

The Checkpoint Master Cylinder Map (CMCM) contains information reflecting the status of the spool data set at the time of the last checkpoint. The CMCM is created by IEFWAMIN, read at system restart time, and is never deleted. The CMCM requires a 15-byte header and a variable number of bytes for its map. (One bit is required in the map for each logical cylinder.)

CMCM Storage Map

DEC	HEX					
0	0	CHMCMID ID FIELD		CHMCMDSK TTRL OF CHECKPOINTED MCM		
8	8	CHMCMTHR SPOOL THRESHOLD VALUE	CHMCMCTR NUMBER OF LOGICAL CYLINDERS ALLOCATED TO SYSTEM AT TIME OF LAST CHECKPOINT	RESERVED	CHMCMFLG FLAGS	FIRST BYTE OF BIT MAP

Fields in CMCM — by displacement

DEC	HEX	FIELD
0	0	CHMCMID
4	4	CHMCMDSK
8	8	CHMCMTHR
10	A	CHMCMCTR
14	E	CHMCMFLG

Alphabetical list of fields in CMCM

FIELD	DEC	HEX
CHMCMCTR	10	A
CHMCMDSK	4	4
CHMCMFLG	14	E
CHMCMID	0	0
CHMCMTHR	8	8

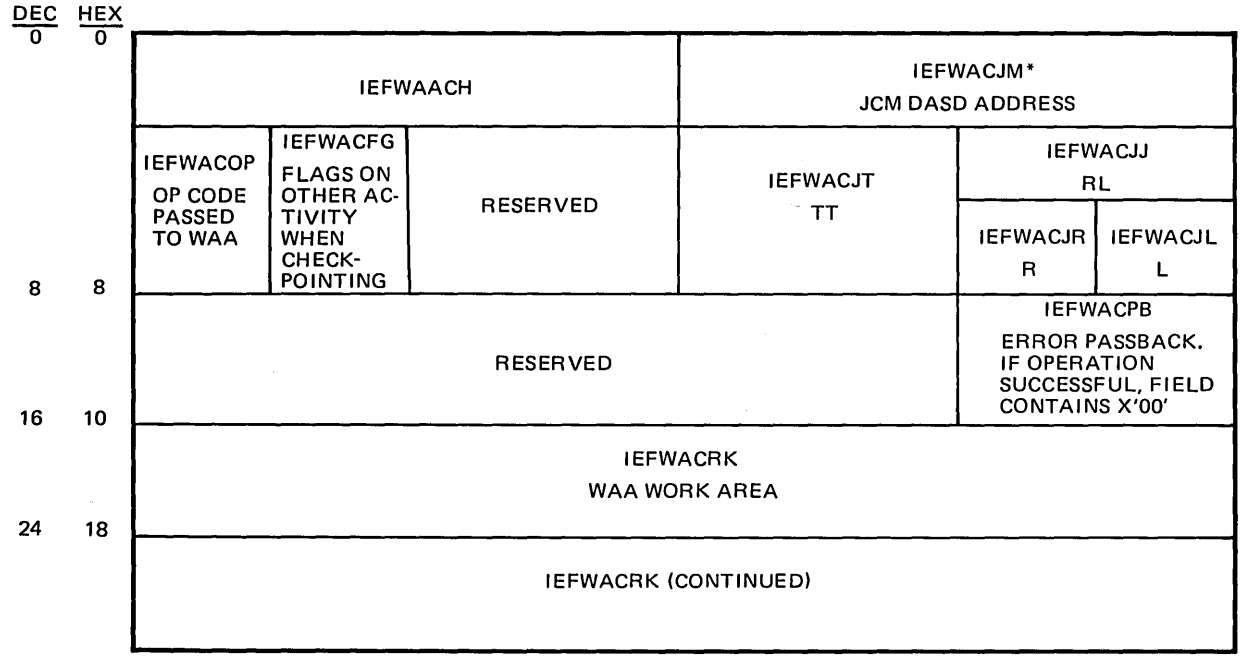
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
CHMCMFLG	FLAGS	CHMCM001	'80'	START INITIATOR/WRITER MESSAGE HAS BEEN ISSUED
		CHMCM003	'20'	JOB QUEUE IS BEING HELD

Checkpoint Work Area Allocation Parameter List (WAACH)

The requester builds the work area allocation parameter list for a checkpoint request (WAACH) and passes it to IEFWAALC to indicate that a checkpoint of a JCM or CMCM is required. WAACH requires 32 bytes of storage.

WAACH Storage Map



* The DASD address of a record is in the form of TTRL where:

TT is the track address relative to the beginning of SYS1.SYSPOOL data set.
R is the block number on tract TT.
L is the logical record number in block R.

Fields in WAACH– by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFWAACH	6	6	IEFWACJJ
0	0	IEFWACOP	6	6	IEFWACJR
1	1	IEFWACFG	7	7	IEFWACJL
4	4	IEFWACJM	14	E	IEFWACPB
4	4	IEFWACJT	16	10	IEFWACRK

Alphabetical list of fields in WAACH

FIELD	DEC	HEX	FIELD	DEC	HEX
IEFWAACH	0	0	IEFWACJR	6	6
IEFWACFG	1	1	IEFWACJT	4	4
IEFWACJJ	6	6	IEFWACOP	0	0
IEFWACJL	7	7	IEFWACPB	14	E
IEFWACJM	4	4	IEFWACRK	16	10

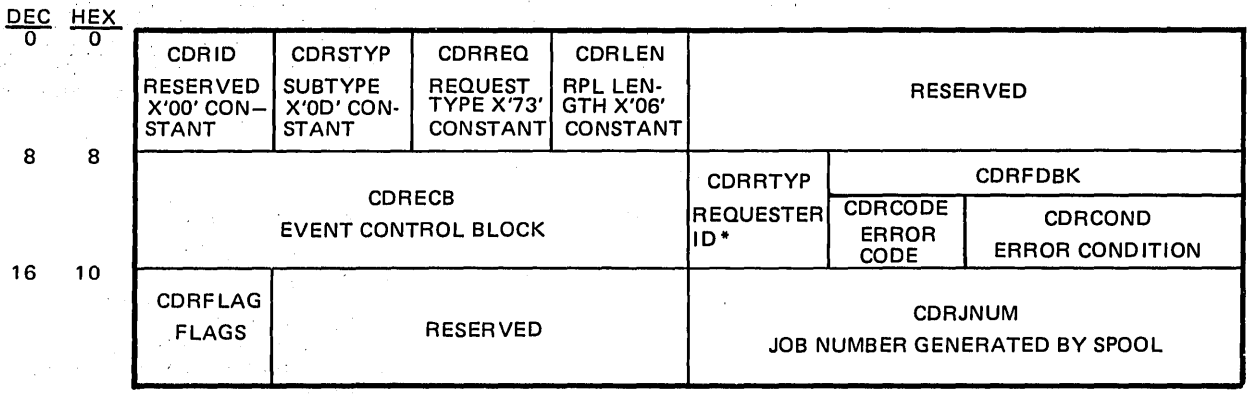
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
IEFWACOP	OP CODE PASSED TO WAA	IEFWAAC	'02'	CODE PASSED AT CLOD OR WHEN CHKPT MACRO ISSUED
IEFWACFG	FLAGS ON OTHER ACTIVITY WHEN CHECKPOINT–ING	IEFWACFR	'80'	FREE JCM SPACE
		IEFWACAC	'40'	SET BY WAA – ALLOCATION HAS TAKEN PLACE SINCE LAST DESTRUCTIVE CLOD – OF INTEREST TO SPOOL MANAGER AT CLOD TIME
		IEFWACRT	'20'	WHEN BIT IS ON, CONTROL RETURNS TO PROCESSING OF JCM EXTENSION
		IEFWACSQ	'10'	USE SQS
IEFWACPB	ERROR PASSBACK. IF OPERATION SUCCESSFUL, FIELD CONTAINS X'00'	IEFWACBD	'80'	FUNCTION NOT PERFORMED – INVALID TTRL
		IEFWACCR	'40'	FUNCTION NOT PERFORMED – WAA GETMAIN NOT SATISFIED
		IEFWACBW	'20'	FUNCTION NOT PERFORMED – JCM CANNOT BE WRITTEN BECAUSE OF BAD TRACK
		IEFWACBX	'04'	RESERVED
		IEFWACX1	'02'	FUNCTION NOT PERFORMED – INVALID OP CODE
		IEFWACX2	'01'	FUNCTION NOT PERFORMED – SPOOL VOLUME IDENTIFICATION NOT POSSIBLE

Close Data Set Request Parameter List (CDRPL)

The user passes the Close Data Set Request Parameter List (CDRPL) to JECS for the close data set function (IEFSMCLD). The user allocates, initializes, and frees the storage required by the CDRPL. CDRPL requires 24 bytes of storage.

CDRPL Storage Map



* The following are valid requester type IDs:

X'01'	Interpreter	X'09'	Initiator
X'02'	Allocation	X'0A'	Checkpoint/Restart
X'03'	WTP	X'0B'	System restart
X'04'	Termination	X'0C'	SWADS
X'05'	Transient	X'10'	JES Writer on behalf of user writer
X'06'	Restart reader in problem program partition	X'40'	Special data set
		X'90'	JES Writer
X'07'	DSO	X'00'	JES Reader
X'08'	Data Management		

Fields in CDRPL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	CDRID	13	D	CDRFDBK
1	1	CDRSTYP	13	D	CDRCODE
2	2	CDRREQ	14	E	CDRCOND
3	3	CDRLEN	16	10	CDRFLAG
8	8	CDRECB	20	14	CDRJNUM
12	C	CDRRTYP			

Alphabetical list of fields in CDRPL

FIELD	DEC	HEX	FIELD	DEC	HEX
CDRCODE	13	D	CDRJNUM	20	14
CDRCOND	14	E	CDRLEN	3	3
CDRECB	8	8	CDRREQ	2	2
CDRFDBK	13	D	CDRRTYP	12	C
CDRFLAG	16	10	CDRSTYP	1	1
CDRID	0	0			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
CDRFLAG	FLAGS	CDRCDST	'80'	BIT 0 = 0 - CHECKPOINT AND RETAIN CORE IMAGE BIT 0 = 1 - CHECKPOINT AND DESTROY CORE IMAGE
		CDRRES	'40'	BIT 0 = 0 - KEEP LOGICAL RECORD COUNT TO ZERO BIT 1 = 1 - RESET LOGICAL RECORD COUNT TO ZERO
		CDRFCOR	'01'	BITS 2 - 7 RESERVED

Command Input Buffer (CIB)

The Command Input Buffer (CIB) is created by the command processor and contains information about the command that has been issued. The CIB is deleted by the task receiving the command or at job termination time.

CIB Storage Map

DEC	HEX					
0	0	CIBNEXT ADDRESS OF NEXT CIB IN QUEUE (ZERO FOR LAST)		CIBVERB COMMAND VERB CODE	CIBLEN LENGTH IN DOUBLE-WORDS OF CIB, INCLUDING CIB DATA	RESERVED FOR CSCB COMPATIBILITY
8	8	RESERVED (CONTINUED)	CIBTJID TSO TERMINAL JOB IDENTIFIER	CIBCONID IDENTIFIER OF CONSOLE ISSUING COMMAND	RESERVED	CIBDATLN LENGTH IN BYTES OF DATA IN CIBDATA.
16	10	CIBDATA DATA FROM COMMAND OPERAND (LENGTH OF CIBDATA IS A MULTIPLE OF EIGHT BYTES, DEPENDING ON THE VALUE CONTAINED IN CIBLEN)				

Fields in CIB — by displacement

DEC	HEX	FIELD
0	0	CIBNEXT
4	4	CIBVERB
5	5	CIBLEN
10	A	CIBTJID
12	C	CIBCONID
14	E	CIBDATLN
16	10	CIBDATA

Alphabetical list of fields in CIB

FIELD	DEC	HEX
CIBCONID	12	C
CIBDATA	16	10
CIBDATLN	14	E
CIBLEN	5	5
CIBNEXT	0	0
CIBTJID	10	A
CIBVERB	4	4

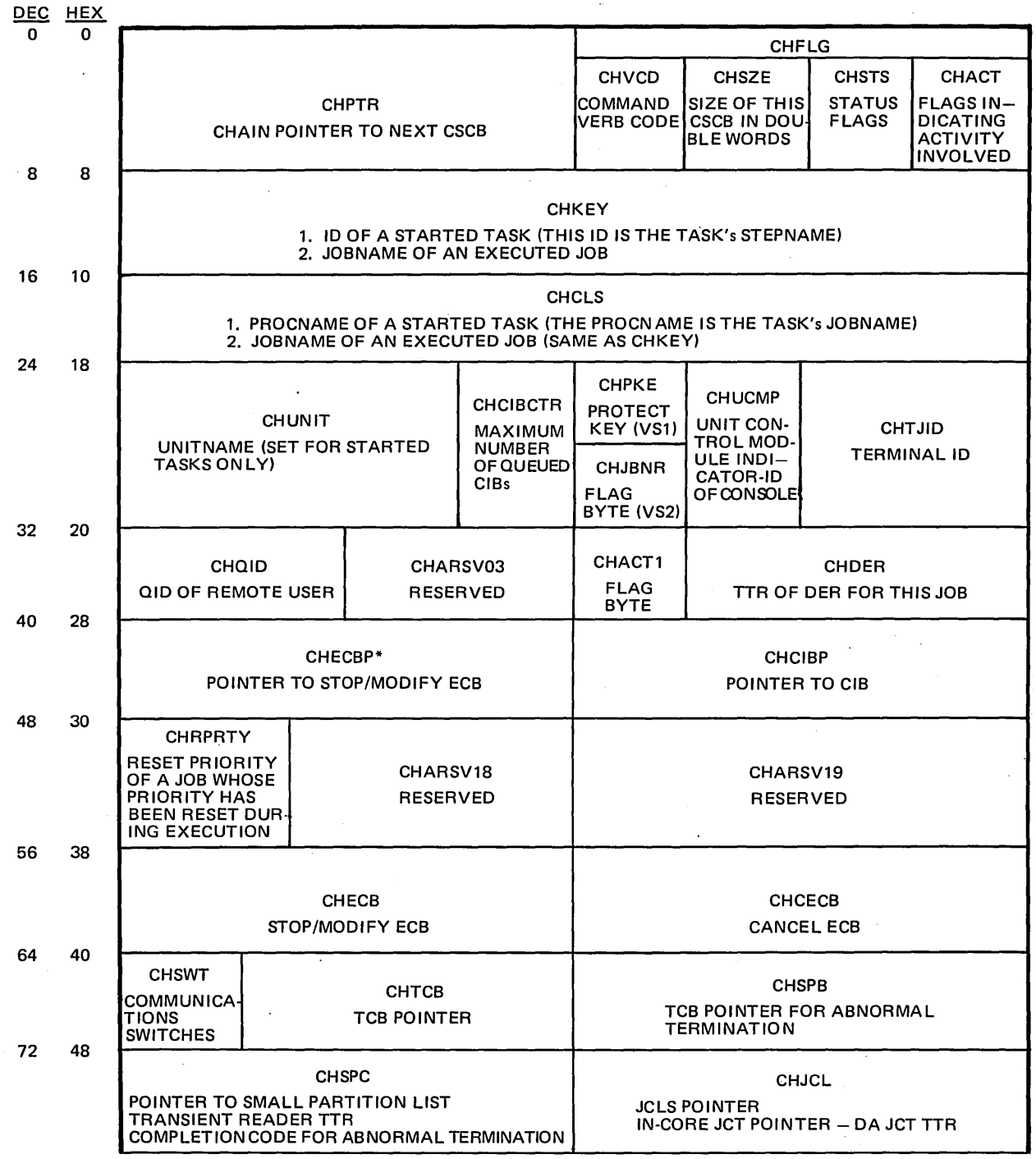
Flags amd Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
CIBVERB	COMMAND VERB CODE	CIBSTART	'04'	COMMAND CODE FOR START
		CIBMODFY	'44'	COMMAND CODE FOR MODIFY
		CIBSTOP	'44'	COMMAND CODE FOR STOP
		CIBMOUNT	'0C'	COMMAND CODE FOR MOUNT

Command Scheduling Control Block (CSCB)

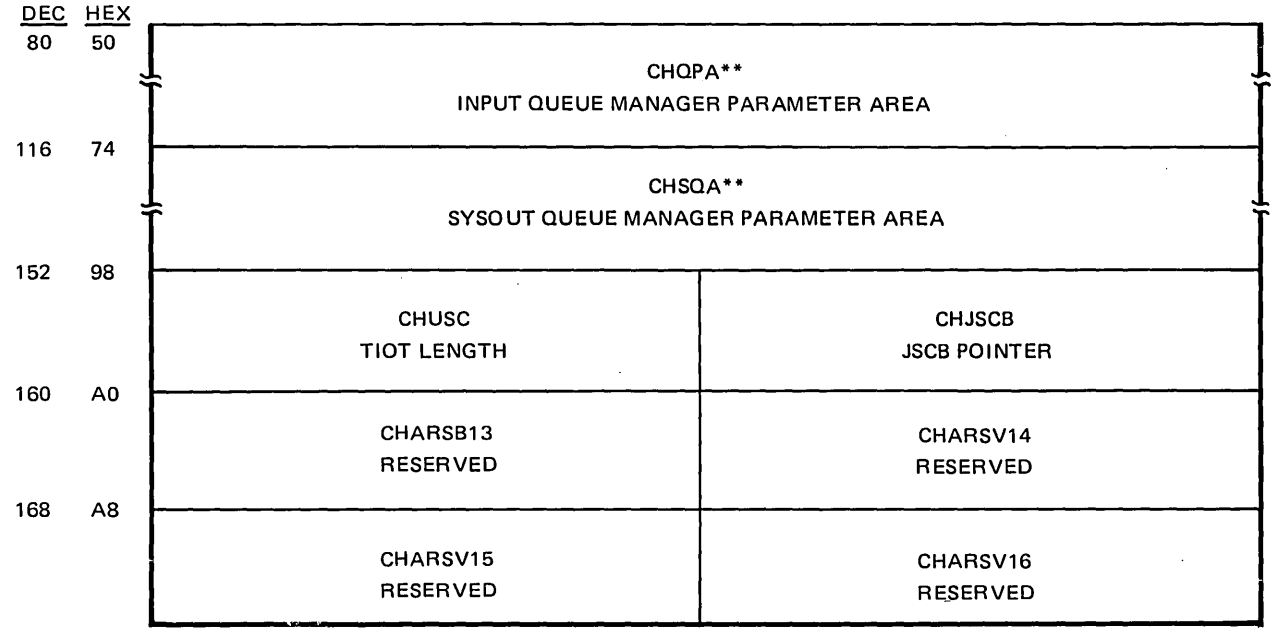
The Command Scheduling Control Block (CSCB) is used to identify system tasks, problem programs, or commands issued to the system as tasks. The CSCB requires 176 bytes of storage and is created by: the command scheduler for the life of a startable program; the queue management dequeue routine for the life of a job between dequeueing and job termination; and the SYSOUT writer for the DSB string in the queue entry to allow cancellation of writer subtasks.

CSCB Storage Map



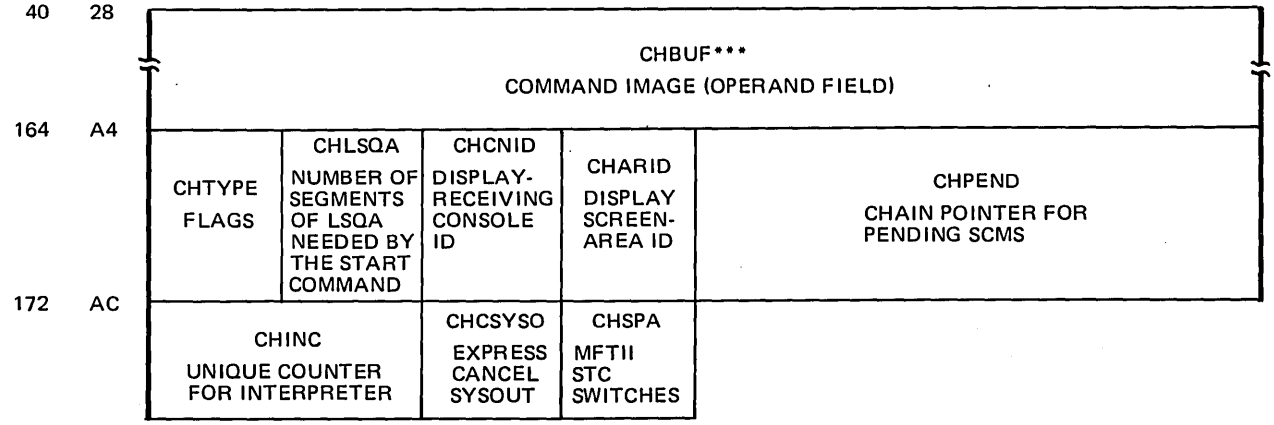
* Beginning of overlay segment.

CSCB Storage Map (Contd.)



** These two QMPAs exist in MVT CSCBs only before intiator select time.

ORG '28' OVERLAY SEGMENT



*** Beginning of mapping unique to input format CSCB before interpretation of command operands.

Fields in CSCB — by displacement

DEC	HEX	FIELD
0	0	CHPTR
4	4	CHFLG
4	4	CHVCD
5	5	CHSZE
6	6	CHSTS
7	7	CHACT
8	8	CHKEY
16	10	CHCLS
24	18	CHUNIT
27	1B	CHCIBCTR
28	1C	CHPKE
28	1C	CHJBNR
29	1D	CHUCMP
30	1E	CHTJID
32	20	CHQID
34	22	CHARSV03
36	24	CHACT1
37	25	CHDER
40	28	CHECBP
44	2C	CHCIBP
48	30	CHRPRTY
49	31	CHARSV18
52	34	CHARSV19
56	38	CHECB
60	3C	CHCECB
64	40	CHSWT
65	41	CHTCB
68	44	CHSPB
72	48	CHSPC
76	4C	CHJCL
80	50	CHQPA
116	74	CHSQA
152	98	CHUSC
156	9C	CHJSCB
160	A0	CHARSV13
164	A4	CHARSV14
168	A8	CHARSV15
172	AC	CHARSV16
ORG '28' OVERLAY SEGMENT		
40	28	CHBUF
164	A4	CHTYPE
165	A5	CHLSQA
166	A6	CHCNID
167	A7	CHARID
168	A8	CHPEND
172	AC	CHINC
174	AE	CHCSYSO
175	AF	CHSPA

Alphabetical list of fields in CSCB

FIELD	DEC	HEX
CHACT	7	7
CHACT1	36	24
CHARSV03	34	22
CHARSV13	160	A0
CHARSV14	164	A4
CHARSV15	168	A8
CHARSV16	172	AC
CHARSV18	49	31
CHARSV19	52	34
CHCECB	60	3C
CHCIBCTR	27	1B
CHCIBP	44	2C
CHCLS	16	10
CHDER	37	25
CHECB	56	38
CHECBP	20	28
CHFLG	4	4
CHJBNR	28	1C
CHJCL	76	4C
CHJSCB	156	9C
CHKEY	8	8
CHPKE	28	1C
CHPTR	0	0
CHQID	32	20
CHQPA	80	50
CHRPRTY	48	30
CHSCZ	5	5
CHSPB	68	44
CHSPC	72	48
CHSQA	116	74
CHSTS	6	6
CHSWT	64	40
CHTCB	65	41
CHTJID	30	1E
CHUCMP	29	1D
CHUNIT	24	18
CHUSC	152	98
CHVCD	4	4
ORG '28' OVERLAY SEGMENT		
CHARID	167	A7
CHBUF	40	28
CHCNID	166	A6
CHCSYSO	174	AE
CHINC	172	AC
CHLSQA	165	A5
CHPEND	168	A8
CHSPA	175	AF
CHTYPE	164	A4

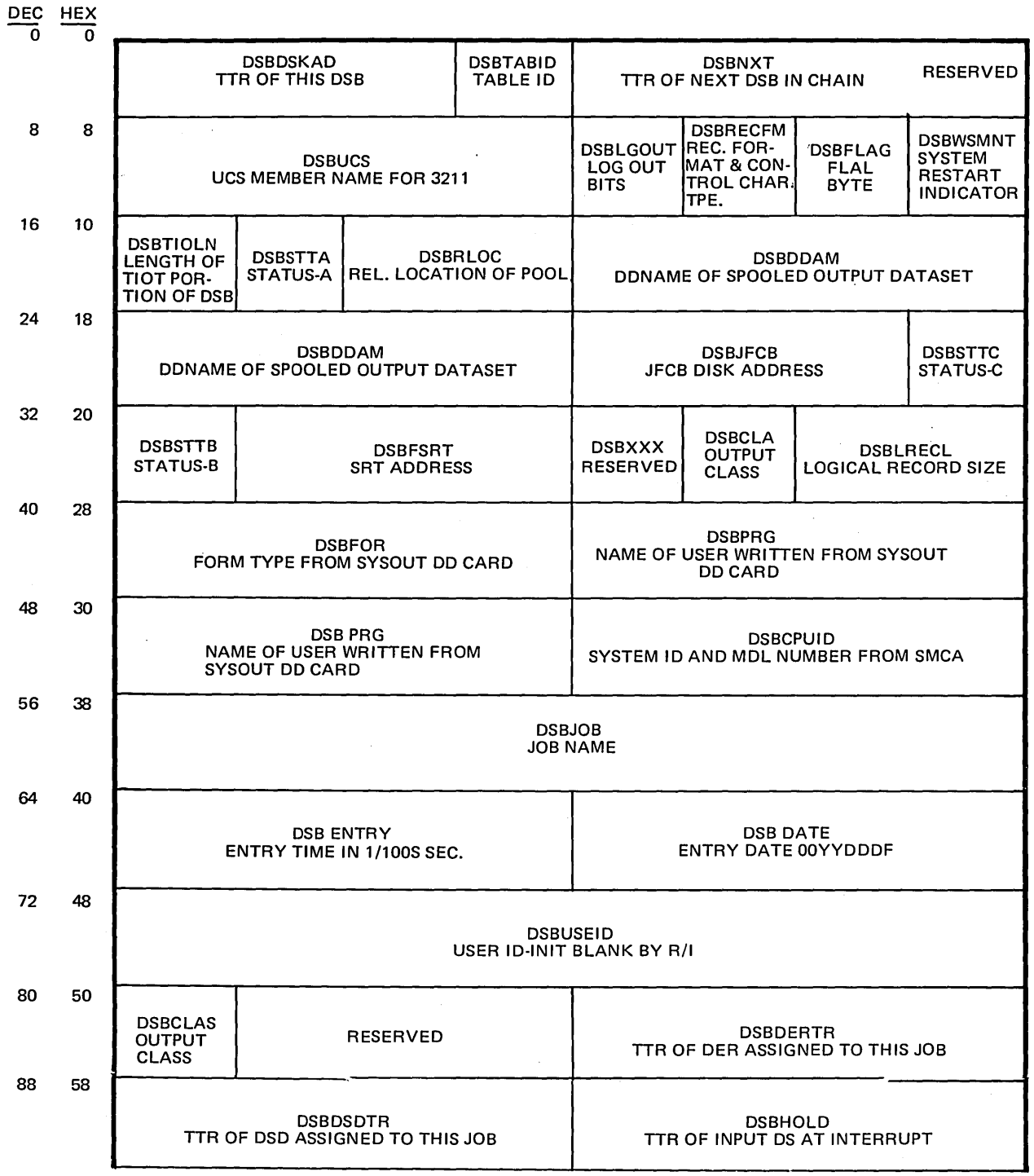
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
CHSTS	STATUS FLAGS	CHAP	'80'	ASSIGNMENT PENDING
		CHSYS	'40'	SYSTEM TASK CSCB
		CHSOUT	'20'	CANCEL ALL SYSOUT
		CHQSPC	'10'	INSUFFICENT QSPACE CAUSING ABEND 422
		CHAD	'08'	ADD THIS CSCB TO CHAIN
		CHDL	'04'	DELETE THIS CSCB FROM CHAIN
		CHFC	'02'	FREE THIS CSCB's CORE
		CHABTERM	'01'	EXECUTE BRANCH ENTRY TO ABTERM
CHACT	FLAGS INDICATING ACTIVITY INVOLVED	CHSWAP	'80'	SWAPPABLE JOB
		CHTERM	'40'	TERMINAL JOB
		CHDISC	'20'	CANCEL IMPLIES DISCONNECT
		CHDSI	'10'	ON MEANS NO DATA SET INTEGRITY
		CHCL	'08'	CANCELABLE JOB STEP
		CHCLD	'04'	CANCEL COMMUNICATION SWITCH
		CHAIFX	'02'	CANCELABLE – MFTII ONLY
		CHIFY	'01'	SYSTEM ASSIGNED PRECEDURE – MFTII
CHJBNR	FLAG BYTE (VS2)	CHJBNRF	'FC'	INITIATOR IDENTIFIER (VS2)
CHACTI	FLAG BYTE	CHRDWTR	'80'	COMMAND WAS START RDR OR WTR
CHCIBP	PTR TO CIB	CHEND	'80'	HIGH ORDER BIT OF LAST PARAMETER POINTER IS ON
CHSWT	COMMUNICA-TIONS SWITCHES	CHJCT	'40'	READER RETURN WITH IN-CORE JCT
		CHPSD	'20'	WRITER PAUSE DATASET
		CHPSF	'10'	WRITER PAUSE FORMS
		CHAC	'08'	ID SPECIFIED ON S COMMAND
CHTYPE	FLAGS	CHDSTAT	'80'	STATUS DISPLAY (SVC104) CMD
		CHHIAR	'02'	ON MEANS H1 SPECIFIED ON COMMAND (ICB337)
		CHDEF	'01'	ON MEANS DEFAULT TO H0 (ICB337)
CHCSYSO	EXPRESS CANCEL SYSOUT	CHALL	'80'	ALL SPECIFIED
		CHINN	'40'	IN SPECIFIED
		CHOUT	'20'	OUT SPECIFIED
		CHHOLD	'10'	HOLD QUEUE SPECIFIED
		CHQUE	'08'	SPECIFIC QUEUE
		CHDUMP	'04'	DUMP SPECIFIED
		CHJB	'02'	END SCAN SWITCH
		CHUSERID	'01'	INDICATES 'USER=' SPECIFIED ON CANCEL

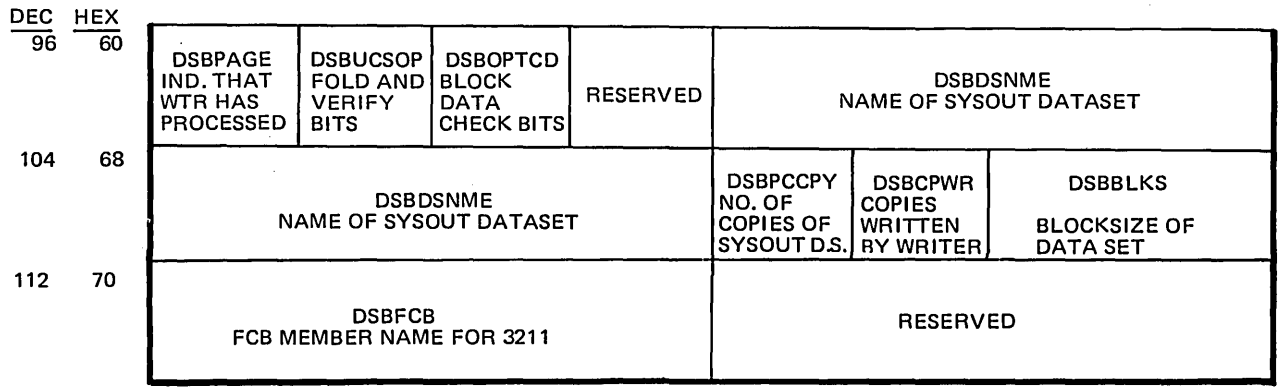
Data Set Block (DSB)

The Data Set Block (DSB) describes the characteristics of the SYSOUT data set as it exists on SPOOL. The interpreter allocates 176 bytes on SYS1.JOBQUEUE for each SYSOUT DD card encountered. Step termination initializes the DSB and the writer deletes it after each SYSOUT class has been processed.

DSB Storage Map



DSB Storage Map (Contd.)



Fields in DSB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	DSBDSKAD	31	1F	DSBSTTC	80	50	DSBCLAS
3	3	DSBTABID	32	20	DSBSTTB	84	54	DSBDERTR
4	4	DSBNXT	34	22	DSBFSRT	88	58	DSBDSDTR
8	8	DSBUCS	36	24	DSBXXX	92	5C	DSBHOLD
12	C	DSBLGOUT	37	25	DSBCLA	96	60	DSBPAGE
13	D	DSBRECFM	38	26	DSBLRECL	97	61	DSBUCSOP
14	E	DSBFLAG	40	28	DSBFOR	98	62	DSBOPTCD
15	F	DSBWSMNT	44	2C	DSBPRG	100	64	DSBSNME
16	10	DSBTIOLN	52	34	DSBCPUID	108	6C	DSBPCCPY
17	11	DSBSTTA	56	38	DSBJOB	109	6D	DSBCPWR
18	12	DSBRLOC	64	40	DSBENTRY	110	6E	DSBBLKS
20	14	DSBDDAM	68	44	DSBDATF	112	70	DSBFCB
28	1C	DSBJFCB	72	48	DSBUSEID			

Alphabetical list of fields in DSB

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
DSBBLKS	110	6E	DSBFOR	40	28	DSBRLOC	18	12
DSBCLA	37	25	* DSBFSRT	34	22	DSBSNME	100	64
DSBCLAS	80	50	DSBHOLD	92	5C	DSBSTTA	17	11
DSBCPUID	52	34	DSBJFCB	28	1C	DSBSTTB	32	20
DSBCPWR	109	6D	DSBJOB	56	38	DSBSTTC	31	1F
DSBDATE	68	44	DSBLGOUT	12	C	DSBTABID	3	3
DSBDDAM	20	14	DSBLRECL	38	26	DSBTIOLN	16	10
DSBDERTR	84	54	DSBNXT	4	4	DSBUCS	8	8
DSBDSDTR	88	58	DSBOPTCD	98	62	DSBUCSOP	97	61
DSBDSKAD	0	0	DSBPAGE	96	60	DSBUSEID	72	48
DSBENTRY	64	40	DSBPCCPY	108	6C	DSBWSMNT	15	F
DSBFCB	112	70	* DSBPRG	44	2C	DSBXXX	36	24
* DSBFLAG	14	E	DSBRECFM	13	D			

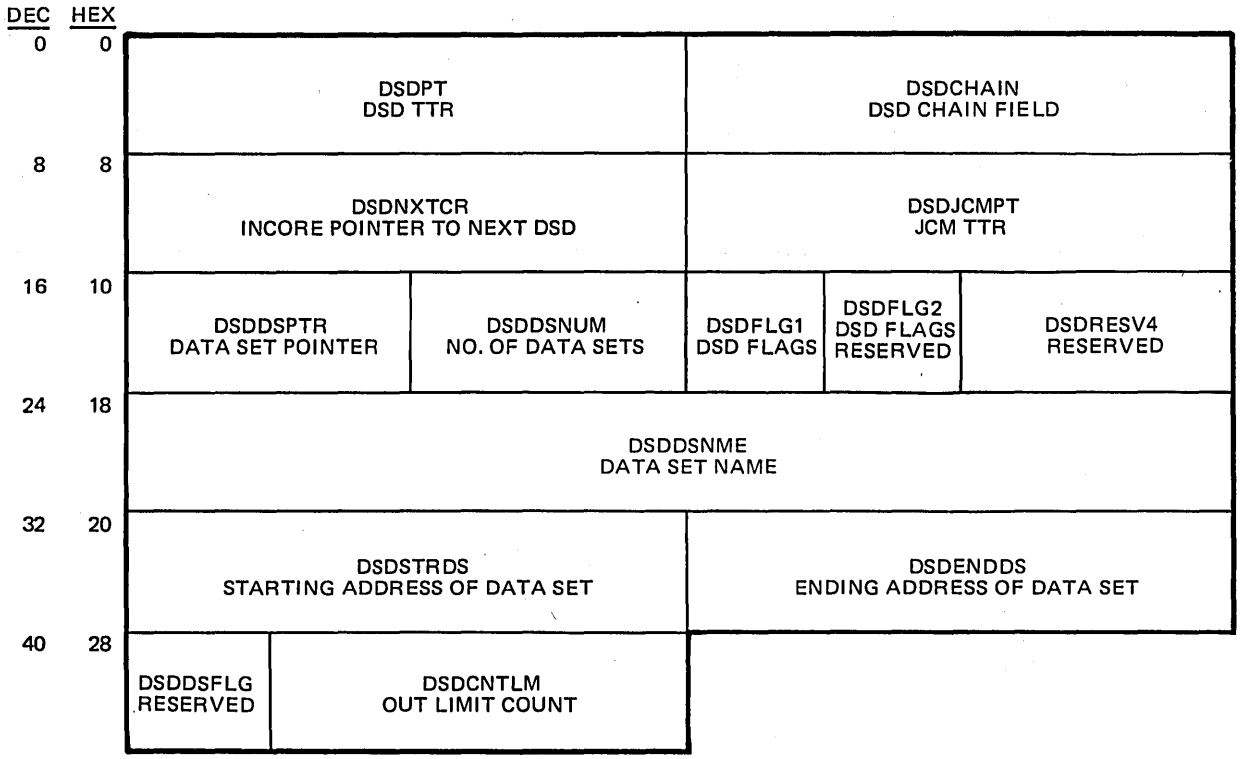
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
DSBFLAG	STATUS BYTE	DSBSYSO	80	SYSOUT DATA SET BIT0-OFF INDICATES SYSTEM MESSAGE DATA SETS
		DSBEMPTY	40	EMPTY SYSOUT DATA SET
		DSBCANCL	20	CANCEL-IGNORE ALL OUTPUT FOR THIS CLASS
DSBFSRT	SRT ADDRESS	DSBBASE	DSBTIOLN-DSBDSKAD	LENGTH OF DSB BASE
DSBPRG	NAME OF USER WRITTEN PACK	DSBSMF	*	ESTABLISHES AN AREA FOR BUILDING SMF RECORDS

Data Set Dictionary (DSD)

The Data Set Dictionary (DSD) maintains status information on all SYS1.SYSPOOL data sets. IEFSMODS obtains the storage space when the current DSD is filled. IEFSMCLD deletes the storage space when a destructive close occurs. It occupies 264 bytes of storage. There are eleven additional entries of the area from Hex 18 to Hex 2C which are repeated until 264 bytes are reached.

DSD Storage Map



Fields in DSD — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	DSDPT	18	12	DSDDSNUM	32	20	DSDSTRDS
4	4	DSDCHAIN	20	14	DSDFLG1	36	24	DSDENDDS
8	8	DSDNXTCR	21	15	DSDFLG2	40	28	DSDDSFLG
12	C	DSDJCMPT	22	16	DSDRESV4	41	29	DSDCNTLM
16	10	DSDDSPTR	24	18	DSDDSNME			

Alphabetical list of fields in DSD

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
DSDCHAIN	4	4	DSDDSNUM	18	12	DSDJCMPT	12	C
DSDCNTLM	41	29	DSDDSPTR	16	10	DSDNXTCR	8	8
DSDDPT	0	0	DSDENDDS	36	24	DSDRESV4	22	16
DSDDSFLG	40	28	* DSDFLG1	20	14	DSDSTRDS	32	20
DSDDSNME	24	18	DSDFLG2	21	15			

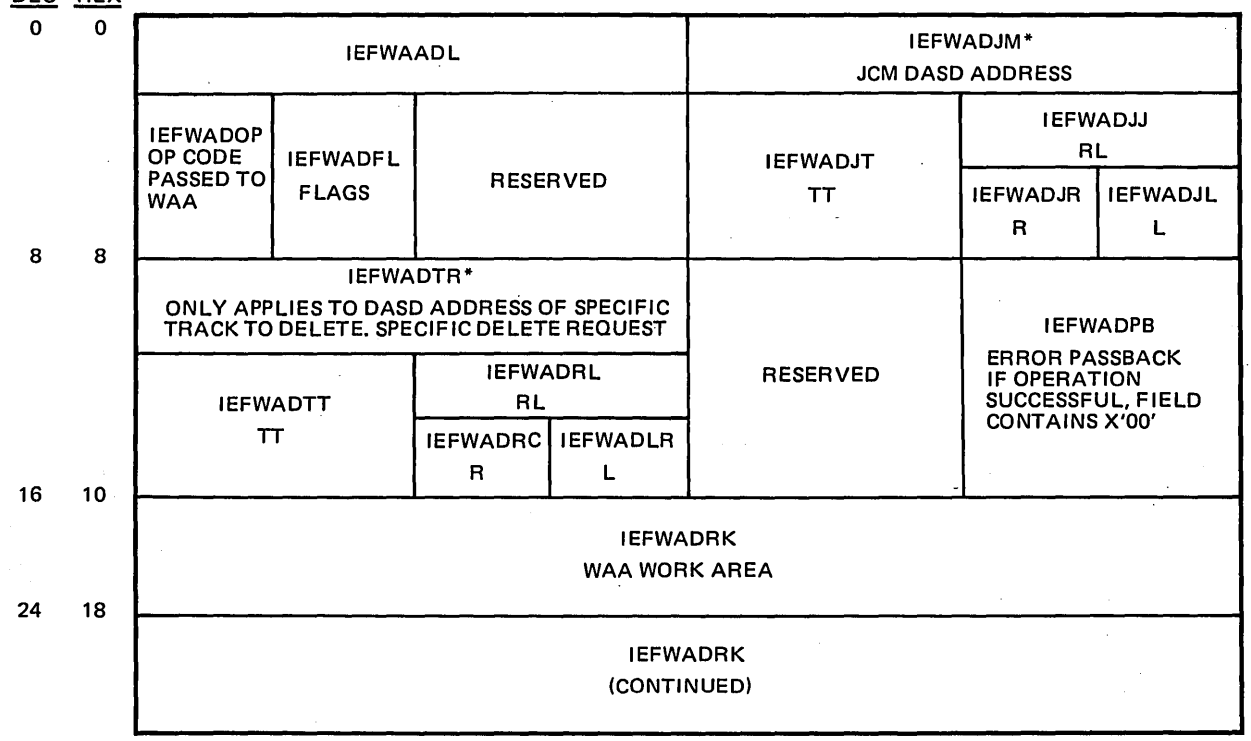
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
DSDFLG1	DSDFLAGS	DSDUPDAT	80	DSD UPDATED INDICATOR BITS 1-7 RESERVED

Delete Work Area Allocation Parameter List (WAADL)

The requester builds the work area allocation parameter list for a delete or specific delete request and passes it to IEFWAALC to indicate that the SYS1.SYSPOOL logical cylinder(s) used by a specific job can be returned to the system. WAADL requires 32 bytes of storage.

WAADL Storage Map



* The DASD address of a record is in the form of TTRL where:

TT is the track address relative to the beginning of SYS1.SYSPOOL data set.
R is the block number on track TT.
L is the logical record number in block R.

Fields in WAADL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFWAADL	8	8	IEFWADTR
0	0	IEFWADOP	8	8	IEFWADTT
1	1	IEFWADFL	10	A	IEFWADRL
4	4	IEFWADJM	10	A	IEFWADRC
4	4	IEFWADJT	11	B	IEFWADLR
6	6	IEFWADJJ	14	E	IEFWADPB
6	6	IEFWADJR	16	10	IEFWADRK
7	7	IEFWADJL			

Alphabetical list of fields in WAADL

FIELD	DEC	HEX	FIELD	DEC	HEX
IEFWAADL	0	0	IEFWADOP	0	0
IEFWADFL	1	1	IEFWADPB	14	E
IEFWADJJ	6	6	IEFWADRC	10	A
IEFWADJL	7	7	IEFWADRK	16	10
IEFWADJM	4	4	IEFWADRL	10	A
IEFWADJR	6	6	IEFWADTR	8	8
IEFWADJT	4	4	IEFWADTT	8	8
IEFWADLR	11	B			

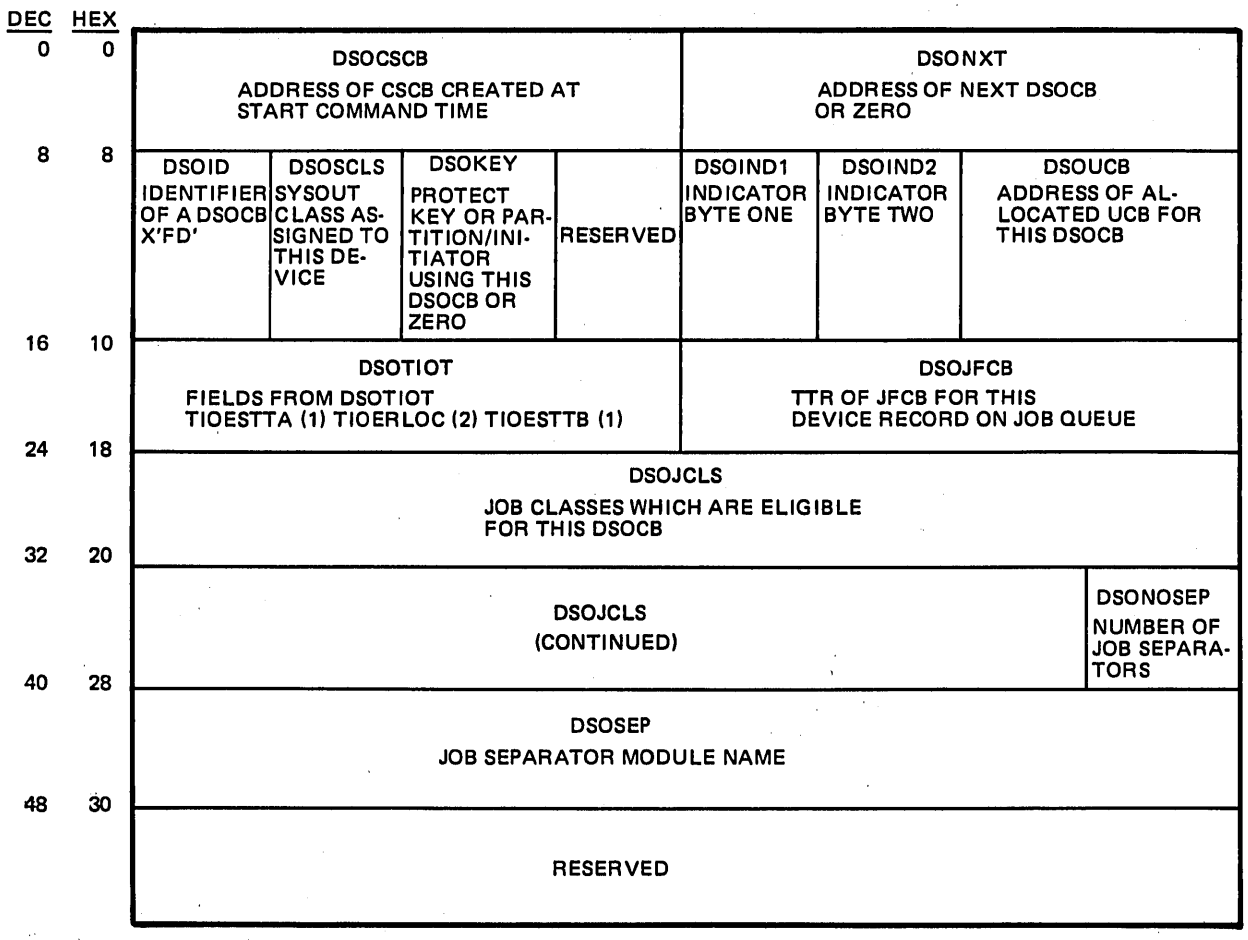
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
IEFWADOP	OP CODE PASSED TO WAA	IEFWAAD	'04'	CODE TO DELETE — RETURN JOB CYLINDERS
IEFWADFL	FLAGS	IEFWADDQ	'80'	WAA WORK FLAG
		IEFWAADS	'01'	REQUEST TO PERMANENTLY DELETE A SPECIFIC LOG CYLINDER FROM SPOOL
IEFWADPB	ERROR PASSBACK	IEFWADBD	'80'	FUNCTION NOT PERFORMED — INVALID TTRL
		IEFWADCR	'40'	FUNCTION NOT PERFORMED — WAA GETMAIN NOT SATISFIED
		IEFWADBW	'20'	ON SPECIFIC DELETE ONLY — FUNCTION NOT PERFORMED — JCM CANNOT BE WRITTEN BECAUSE OF BAD TRACK
		IEFWADBR	'10'	FUNCTION NOT PERFORMED — JCM CANNOT BE READ BECAUSE OF BAD TRACK
		IEFWADNA	'08'	RESERVED
		IEFWADBX	'04'	RESERVED
		IEFWADX1	'02'	FUNCTION NOT PERFORMED — INVALID OP CODE
		IEFWADX2	'01'	FUNCTION NOT PERFORMED — SPOOL VOLUME IDENTIFICATION NOT POSSIBLE

Direct SYSOUT Control Block (DSOCB)

The Direct SYSOUT Control Block (DSOCB) is built and initialized whenever the direct SYSOUT writer is started in a partition. A DSO initialization routine builds it when DSO is started and frees it when DSO is stopped. Users of the DSOCB include the initiator, termination, and direct SYSOUT routines. The DSOCB occupies 56 bytes of storage.

DSOCB Storage Map



Fields in DSOCB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	DSOCSCB	14	E	DSOUCB
4	4	DSONXT	16	10	DSOTIOT
8	8	DSOID	20	14	DSOJFCB
9	9	DSOSCLS	24	18	DSOJCLS
10	A	DSOKEY	39	27	DSONOSEP
12	C	DSOIND1	40	28	DSOSEP
13	D	DSOIND2			

Alphabetical list of fields in DSOCB

FIELD	DEC	HEX	FIELD	DEC	HEX
DSOCSCB	0	0	DSONOSEP	39	27
DSOID	8	8	DSONXT	4	4
DSOIND1	12	C	DSOSCLS	9	9
DSOIND2	13	D	DSOSEP	40	28
DSOJCLS	24	18	DSOTIOT	16	10
DSOJFCB	20	14	DSOUCB	14	E
DSOKEY	10	A			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
ECBBITS		DSOWAIT	'80'	BIT 0 ECB WAIT BIT
		DSOPOST	'40'	BIT 1 ECB POST BIT
DSOIND1	INDICATOR BYTE 1	DSOTAPE	'80'	BIT 0 TAPE OUTPUT DEVICES
		DSOPRINT	'40'	BIT 1 PRINTER OUTPUT
		DSOPUNCH	'20'	BIT 2 PUNCH OUTPUT DEVICES
		DSOCR	'10'	BIT 3 THIS DSOCB NOT AVAILABLE FOR THIS STEP (CONDITIONED BY C/R)
		DSOSTEP	'08'	BIT 4 THIS DSO HAS BEEN SELECTED FOR PRESENT STEP
		DSOJSEP	'04'	BIT 5 JOB SEPARATORS HAVE BEEN WRITTEN
		DSOMSGCL	'02'	'DSOSCL5' ALSO EQUALS THE JOB MESSAGE CLASS
		DSONOOP	'01'	THIS DSOCB NO LONGER AVAILABLE FOR SELECTION
DSOIND2	INDICATOR BYTE 2	DSOISTOP	'80'	BIT 0 INTERNAL STOP ISSUED
		DSOSM	'40'	BIT 1 STOP/MODIFY PENDING

Disk Entry Record (DER)

The Disk Entry Record (DER) contains information describing the user's job that is required by the scheduler during processing. The reader builds the 176-byte DER when it detects a JOB card in the input stream. The writer deletes the DER after all SYSOUT classes for the job are processed.

DER Storage Map

DEC	HEX				
0	0	DERDSKAD TTR OF THIS DER	DERID INDICATOR FLAG BYTE	DERNXT TTR OF JMR	RESERVED
8	8	DERIDSD TTR OF INPUT DSD		DERODSD TTR OF OUTPUT DSD	
16	10	DERIJCM TTR OF INPUT JCM		DEROJCM TTR OF OUTPUT JCM	
24	18	DERRCT TTR OF FIRST RECORD CONTROL TBL		DERJCT TTR OF JCT ON JOBQ OR SWADS	
32	20	DERINJCT TTR OF INITIATOR TIOT		DERSTDCB ADDRESS OF SWADS DCB	
40	28	DERSTAWR NO. OF ASSIGN/WRITES TO SWADS	DERDSBCT NO. OF DSB ASSIGNED TO JOB	DERLOG LOG STA- TUS BYTE	DERCQMP2 COMPRESSED QMPA FOR LOG PROCESSING
48	30	DERCQMP2 COMPRESSED QMPA FOR LOG PROCESSING		DERSTREC TOTAL NUMBER OF SWADS RECORDS	
56	38	DERSTRTK NUMBER OF SWADS RECORDS/TRACK	DERINREV NUMBER OF SWADS OVERFLOW RECORDS	DERJCLDF JCL DEFAULTS	
96	60	DERJCLDF JCL DEFAULTS		DERFLAGS DER FLAG BIT 7 UNUSED	DERCQMPA COMPRESSED QMPA
104	68	DERCQMPA COMPRESSED QMPA		DERJACT TTR OF JACT RECORD	
112	70	DERSCDTR TTR OF SCD		DERJOBID JOB IDENTIFICATION	
				DERJOBNM JOB NAME	
120	78	DERJOBID JOB IDENTIFICATION		DERJOBID JOB IDENTIFICATION	
		DERJOBNM (CONTINUED)		DERJOBNO JOB NUMBER	
128	80	DERTTRRT TTR OF THE ROUTE TABLE		DERQID QUEUE ID	DERFLAG1 DER FLAGS RESERVED (41 BYTES)

Fields in DER — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	DERDSKAD	36	24	DERSTDCB	101	65	DERCQMPA
3	3	DERID	40	28	DERSTAWR	108	6C	DERJACT
4	4	DERNXT	42	2A	DERDSBCT	112	70	DERSCDTR
8	8	DERIDSD	44	2C	DERLOG	116	74	DERJOBID
12	C	DERODSD	45	2D	DERCQMP2	116	74	DERJOBNM
16	10	DERIJCM	52	34	DERSTREC	124	7C	DERJOBND
20	14	DEROJCM	56	38	DERSTRTK	128	80	DERTTRRT
24	18	DERRCT	58	3A	DERINREV	132	84	DERQID
28	1C	DERJCT	60	3C	DERJCLDF	134	86	DERFLAG1
32	20	DERINJCT	100	64	DERFLAGS			

Alphabetical list of fields in DER

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
DERCQMPA	101	65	DERINREV	58	3A	DEROJCM	20	14
DERCQMP2	45	2D	DERJACT	108	6C	DERQID	132	84
DERDSBCT	42	2A	DERJCLDF	60	3C	DERRCT	24	18
DERDSKAD	0	0	DERJCT	28	1C	DERSCDTR	112	70
* DERFLAGS	100	64	DERJOBID	116	74	DERSTAWR	40	28
* DERFLAG1	134	86	DERJOBNM	116	74	DERSTDCB	36	24
DERID	3	3	DERJOBNO	124	7C	DERSTREC	52	34
DERIDSD	8	8	* DERLOG	44	2C	DERSTRTK	56	38
DERIJCM	16	10	DERNXT	4	4	DERTTRRT	128	80
DERINJCT	32	20	DERODSD	12	C			

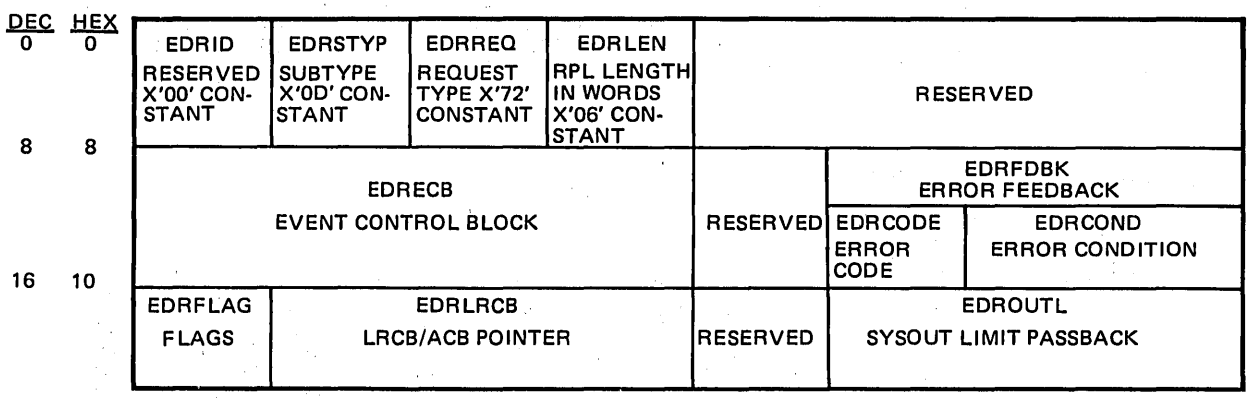
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
DERLOG	LOG STATUS BYTE	DERLOGID	80	DER FOR LOG STATUS BYTE
		DERLOGXD	40	LOG X OPENED
		DERLOGXE	20	LOG X ENQUEUED
		DERLOGYO	10	LOG Y OPENED
		DERLOGYE	08	LOG Y ENQUEUED
DERFLAGS	DER FLAG-BIT 7	DERJOBRS	80	JOB TO BE RESTARTED
	UNUSED	DERWMST	40	WARM START CONDITION
		DERJOBF	20	JOB FAIL BIT
		DERINTRP	10	INTERPRETATION COMPLETE
		DERTERM	08	TERMINATION COMPLETE
		DERGSTRT	04	GENERALIZED START IDENTIFICATION
		DERPROCS	02	PROCEDURE SPOOLED INDICATOR
DERFLAG1	DER FLAGS-BITS 2-7	DERDSO	80	JOB SELECTED BY DSO WRITER
	UNUSED	DERSYSTK	40	JOB IS A SYSTEM TASK

End Data Set Request Parameter List (EDRPL)

The user passes the End Data Set Request Parameter List (EDRPL) to JECS to request the end data-set function or the forced PUT function (IEFSMEND). The user allocates and frees the storage required by the EDRPL. EDRPL required 24 bytes of storage.

EDRPL Storage Map



Fields in EDRPL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	EDRID	13	D	EDRCODE
1	1	EDRSTYP	14	E	EDRCOND
2	2	EDRREQ	16	10	EDRFLAG
3	3	EDRLEN	17	11	EDRLRCB
8	8	EDRECB	21	15	EDROUTL
13	D	EDRFDBK			

Alphabetical list of fields in EDRPL

FIELD	DEC	HEX	FIELD	DEC	HEX
EDRCODE	13	D	EDRLEN	3	3
EDRCOND	14	E	EDRLRCB	17	11
EDRECB	8	8	EDROUTL	21	15
EDRFDBK	13	D	EDRREQ	2	2
EDRFLAG	16	10	EDRSTYP	1	1
EDRID	0	0			

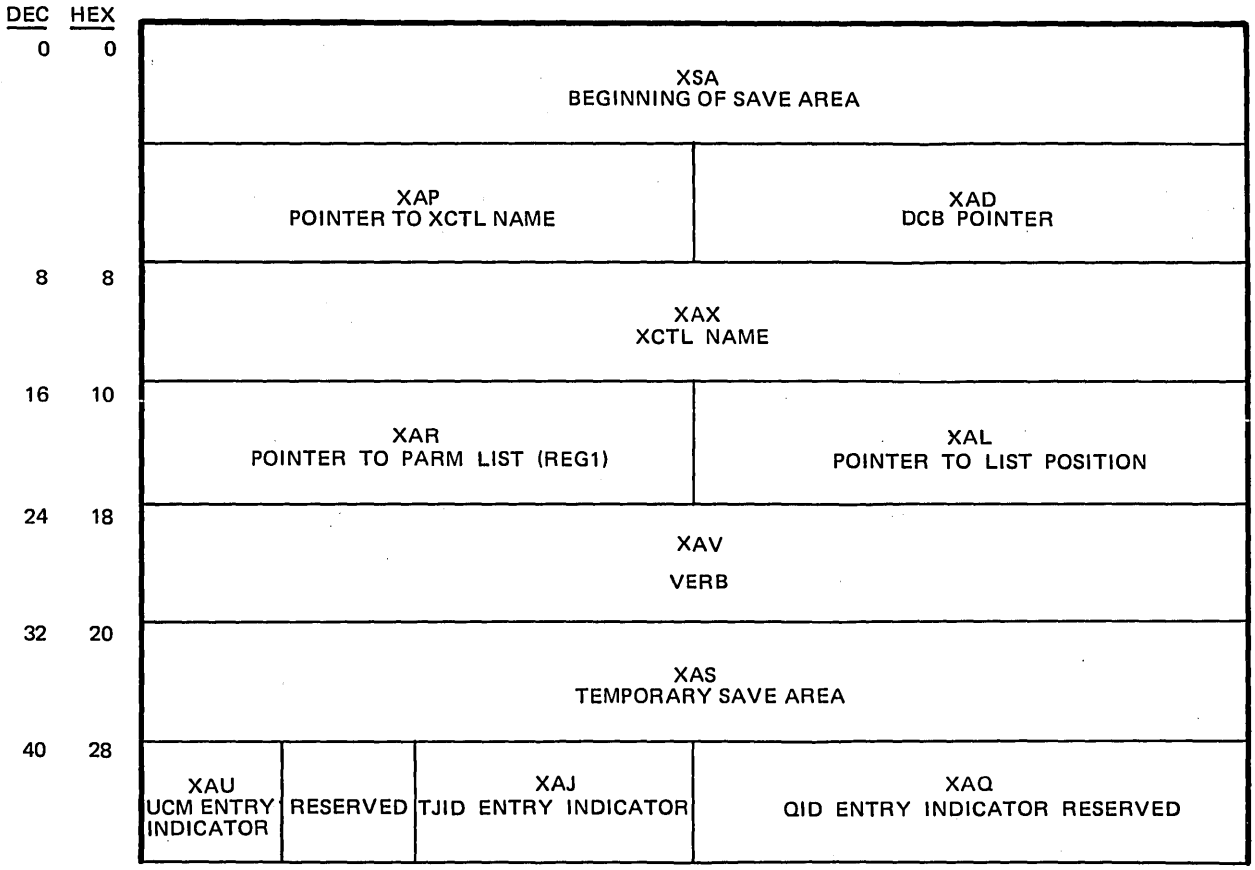
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
EDRFLAG	FLAGS	EDRFPUT	'80'	BIT 0 = 0 NORMAL END BIT 0 = 1 FORCED PUT BITS 1-7 - RESERVED

Extended Save Area (XSA)

Transient routines (command processors) use the Extended Save Area (XSA) as a work area. It has 48 bytes and contains pointers to text entered into the console, as well as a parameter list to pass information from one transient routine to another. The second level interrupt handler builds it as an extension to the SVRB and deletes it when the SVRB is freed. It is built for type 3 and type 4 SVC modules. For MLWTO, the XSA has forty bytes.

XSA Storage Map



XSA Storage Map (Contd.)

MESSAGE AREA TABLE

DEC	HEX		
0	0	XAH RECORD HEADER	XAI MESSAGE ID
8	8	XAI MESSAGE ID	XAF FILL (VARIABLE TEXT)
16	10	XAF FILL (VARIABLE TEXT)	XAT PREFORMATTED TEXT, DESCRIPTOR CODE, ROUTING CODE
24	18	XAT PREFORMATTED TEXT	XAT PREFORMATTED TEXT
32	20	XAT PREFORMATTED TEXT	XAT PREFORMATTED TEXT
40	28	XAT PREFORMATTED TEXT	RESERVED

Fields in XSA — by displacement

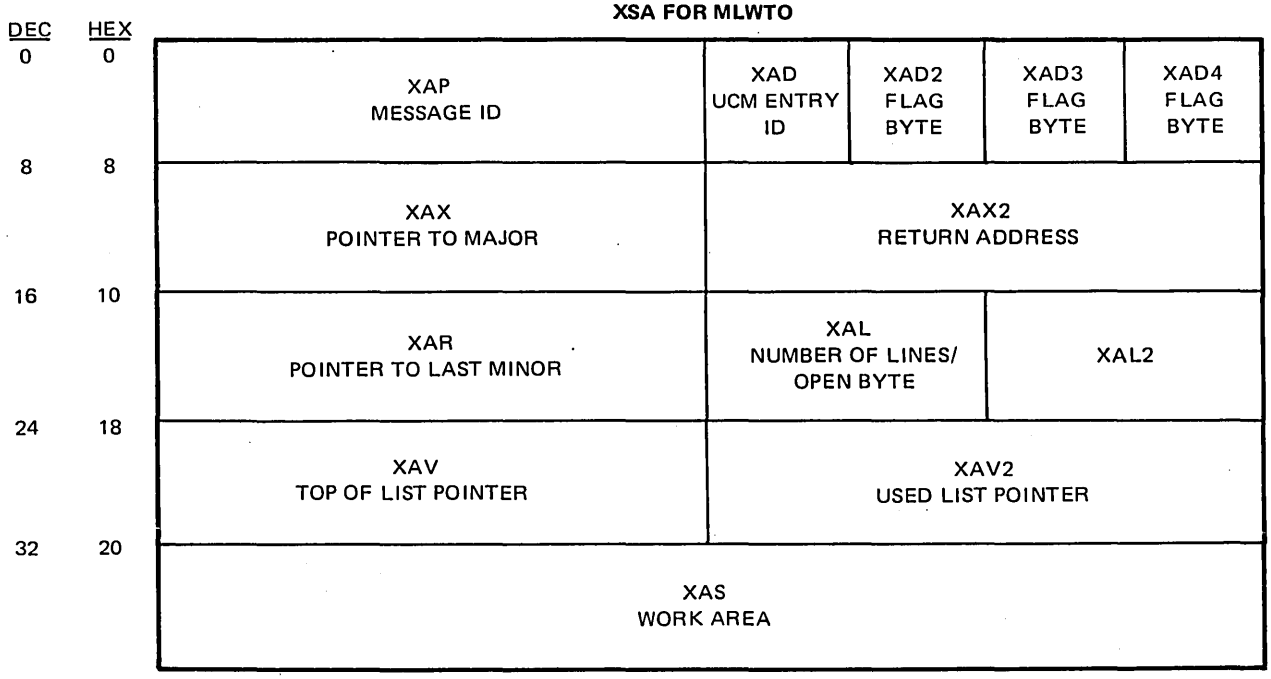
DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	XSA	24	18	XAV			MESSAGE AREA TABLE
0	0	XAP	32	20	XAS	0	0	XAH
4	4	XAD	40	28	XAU	4	4	XAI
8	8	XAX	42	30	XAJ	8	8	XAF
16	10	XAR	44	3C	XAQ	20	1C	XAT
20	14	XAL						

Alphabetical list of fields in XSA

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
XAD	4	4	XAS	32	20			MESSAGE AREA TABLE
XAJ	42	30	XAU	40	28	XAF	8	8
XAL	20	14	XAV	24	18	XAH	0	0
XAP	0	0	XAX	8	8	XAI	4	4
XAQ	44	3C	XSA	0	0	XAT	20	1C
XAR	16	10						

XSA Storage Map (Contd.)

XSA FOR MLWTO



FIELDS IN XSA FOR MLWTO

DEC	HEX	FIELD	DEC	HEX	FIELD
4	4	XAD	16	10	XAR
5	5	*XAD2	32	20	XAS
6	6	*XAD3	24	18	XAV
7	7	*XAD4	28	1C	XAV2
20	14	XAL	8	8	XAX
22	16	XAL2	12	C	XAX2
0	0	XAP			

*FLAGS	CONTAINS	VALUE	MEANS
XAD2	FLAG BYTE	X'80'	FIRST ENTRY
		X'40'	CODE = 4
		X'20'	CHAIN RE-USED
		X'10'	ENQUEUE DONE
		X'08'	PROBLEM STATE
		X'04'	SECOND HALF AVAILABLE
		X'02'	CODE = 8
		X'01'	FIRST 'LINECHK' ENTRY
XAD3	FLAG BYTE	X'80'	OPEN
		X'40'	QUEUE MAJOR
		X'20'	FORCE END
		X'10'	OUR CONTROL LINE USED
		X'08'	QUEUE MINOR
		X'04'	CODE = 12
		X'02'	CODE = 16
		X'01'	CODE = 20
XAD4	FLAG BYTE	X'80'	CONTROL LINE FOUND
		X'40'	ONE LABEL FOUND
		X'20'	SECOND LABEL FOUND
		X'10'	LAST LINE WAS CONTROL
		X'08'	LAST LINE WAS LABEL
		X'04'	OPEN
		X'02'	OPEN
		X'01'	OPEN

Initiator Entrance List (IEL)

The Initiator Entrance List (IEL), which is six fullwords long, contains pointers to the tables used in communicating information between the caller and the initiator.

The IEEVRCTL and IEFIIC routines build the IEL for a started initiator and IEFSD160 deletes them. IEL is built for any started task.

IEL Storage Map

DEC	HEX							
0	0	IELPARM ADDRESS OF 'PARM' FIELD			IELCOM ADDRESS OF ECB/CIB COMMUNICATION LIST			
8	8	IELQPLST ADDRESS OF INITIATOR OPTION LIST			IELQMPA ADDRESS OF QUEUE MANAGER PARAMETER AREA			
16	10	IELJCT ADDRESS OF JOB CONTROL TABLE			IELEXIT ADDRESS OF INITIATOR EXIT LIST			
24	18	IELOPLEN LENGTH OF OPTION LIST	IELOPSW1 OPTION SWITCH ONE	IELOPSW2 OPTION SWITCH TWO	IELOPSW3 OPTION SWITCH THREE	IELOPSW4 OPTION SWITCH FOUR	RESERVED	IELPKEY PROTECT KEY
32	20	IELECB ADDRESS OF ECB LIST						
40	28	IELXTLEN LENGTH OF EXITLIST	IELRTNCD RETURN CODE	RESERVED	RTNEXLR LINKAGE DEFINITION	RTNEXID EXIT DEFINITION	RTNEXNM EXIT NAME	
48	30	RTNEXNM (continued) EXIT NAME						

Fields in IEL — by displacement

DEC	HEX	FIELD
0	0	IELPARM
4	4	IELCOM
8	8	IELOPLST
12	C	IELQMPA
16	10	IELJCT
20	14	IELEXIT
24	18	IELOPLEN
26	1A	IELOPSW1
27	1B	IELOPSW2
28	1C	IELOPSW3
29	1D	IELOPSW4
31	1F	IELPKEY
32	20	IELECB
40	28	IELXTLEN
42	2A	IELRTNCD
45	2D	RTNEXID
46	2E	RTNEXNM

Alphabetical list of fields in IEL

FIELD	DEC	HEX
IELCOM	4	4
IELCB	32	20
IELEXIT	20	14
IELJCT	16	10
IELOPLEN	24	18
IELOPLST	8	8
IELOPSW1	26	1A
IELOPSW2	27	1B
IELOPSW3	28	1C
IELOPSW4	29	1D
IELPARM	0	0
IELPKEY	31	1F
IELQMPA	12	C
IELRTNCD	42	2A
IELXTLEN	40	28
RTNEXID	45	2D
RTNEXLR	44	2C
RTNEXNM	46	2E

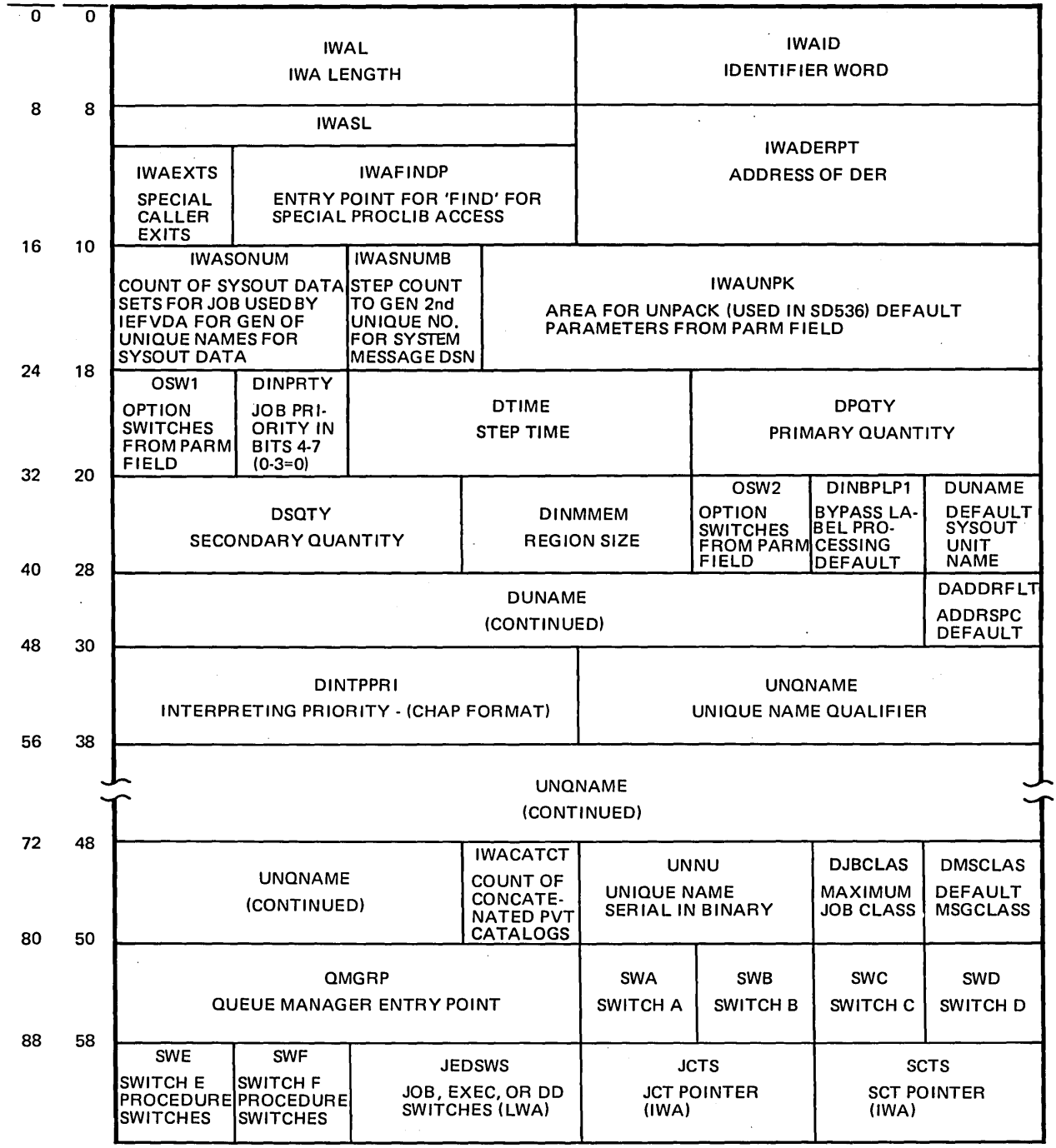
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
IELEXIT	ADDRESS OF INITIATOR EXIT LIST	IELEND	80	SYMBOL TO SET HIGH-ORDER BIT OF IELLAST
IELOPTS		IELOPTLN	16	SYMBOLIC LENGTH OF OPTION LIST
IELOPSW1	OPTION SWITCH ONE	IELPKEYF	80	INIT. TO ATTACH TASK IN PROTECT KEY IELPKEY
		IELDWFF	40	BYPASS USE OF DEDICATED WORK FILES
		IELSTMDF	20	DO NOT PROCESS STOP OR MODIFY COMMANDS
		IELMINPF	10	GET REGION EVEN IF LESS THAN MINPART
		IELCANF	08	SUPPRESS CANCEL EXCEPT DURING ALLOCATION
		IELONEJF	04	PROCESS ONLY ONE JOB
		IELICMDF	02	BYPASS START/MODIFY CIB AND 'PARM' INFO.
		IELWTPF	01	SET UP WRITE-TO-PROGRAMMER CHAIN
IELOPSW2	OPTION SWITCH TWO	IELTIMEF	80	DO NOT TIME STEPS
		IELCRF	40	DO NOT PROCESS CHECKPOINT/RESTART
		IELDSOF	20	SUPPRESS DIRECT SYSOUT
		IELINTH0	10	INITIATED PROGRAM TO RUN IN HIERARCHY ZERO
		IELINTH1	08	INITIATED PROGRAM TO RUN IN HIERARCHY ONE
IELOPSW3	OPTION SWITCH THREE	IELS161	80	SUSPEND IN IEFSD161
		IELSNOWK	40	SUSPEND ON NO WORK
		IELSBTJB	20	SUSPEND BETWEEN JOBS
		IELECBF	10	DO NOT CONSTRUCT ECB LIST
		IELPIBJB	08	GET JOB CLASSES FROM PIB
		IELPRIOR	04	BYPASS STEP DISPATCHING AUTHORITY
		IELGCB	02	BYPASS GSB PROCESSING
		IELCRREG	01	CHECK REGION BOUNDARIES (C/R)
IELOPSW4	OPTION SWITCH FOUR	IELECBPB	80	PLACE ECB LIST POINTER INTO PIB
		IELREGON	40	BYPASS REGION PROCESSING
		IELATDET	20	BYPASS ATTACH/DETACH
		IELWRITE	10	WRITE THE LOT WITH THE TIOT
		IELNREAD	08	DO NOT READ JCT AND SCT IN STEP DELETE
		IELSBOL	04	INITIATOR TO GET THE FOLLOWING CONTROL BLOCKS IN SUBPOOL 255 1. WTP CONTROL BLOCK (WTPCB) 2. JOB STEP CONTROL BLOCK (JSCB)
		IELNPKEY	02	PROGRAM WILL RUN IN KEY ZERO
		IELMFTIO	01	IEFMFTIO WILL BE USED FOR TERMINATING THIS PROGRAM.
&EXITS (&N) EXAD	EXIT ADDRESS	IELEXNOP	00	EXIT IS NO-OPED
		IELEXADD	40	BRANCH TO ADDRESS
		IELEXLNK	80	LINK TO NAME
		IELEXXTL	C0	XCTL TO NAME
		IELRTNEX	80	RETURN EXIT AFTER TERMINATION

Interpreter Work Area (IWA)

The Interpreter Work Area (IWA) contains switches, save areas, and data to be passed from one interpreter module to another. The IWA is created at interpreter initialization time and deleted at the end of interpretation. The IWA requires 2048 bytes of storage.

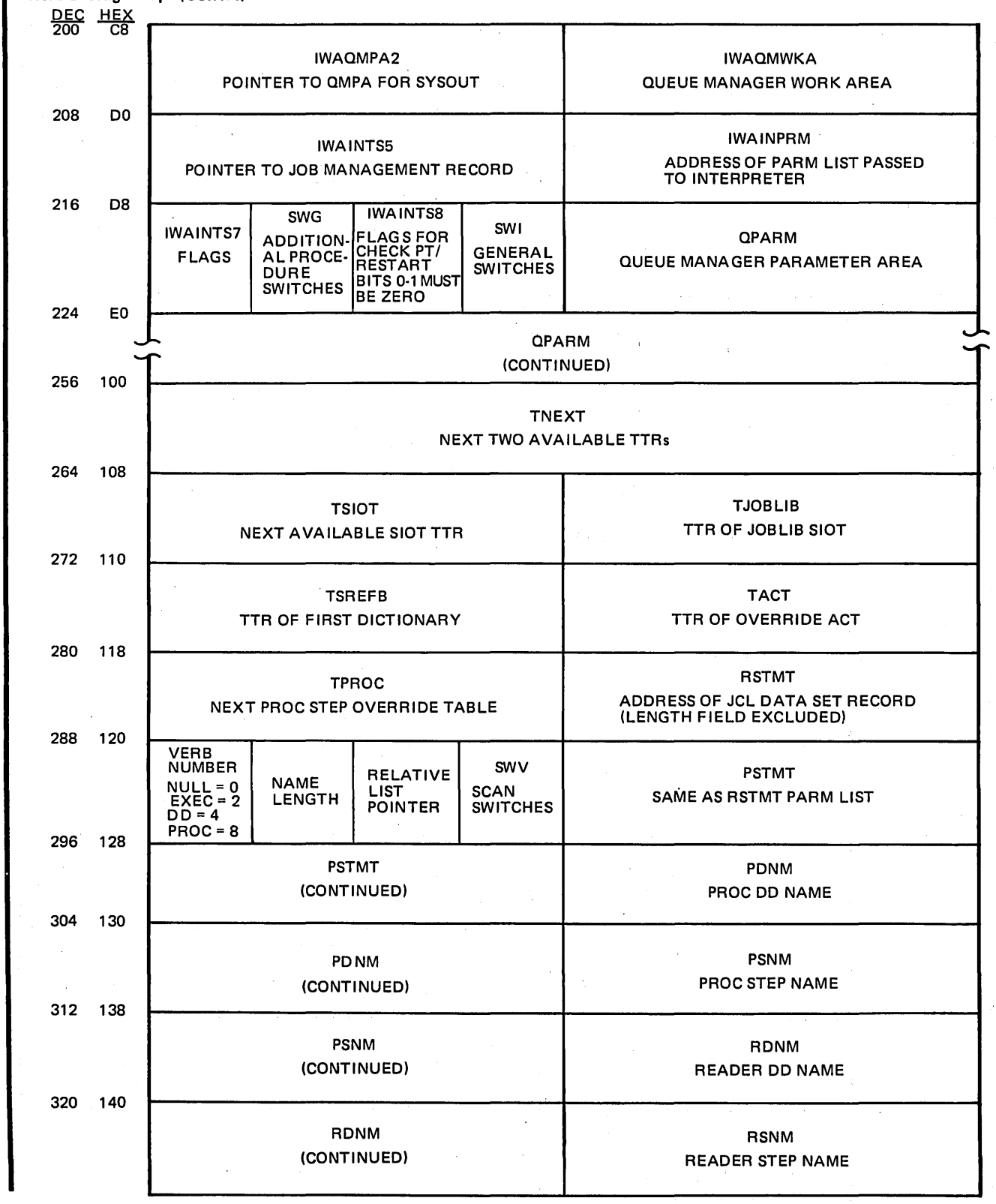
IWA Storage Map



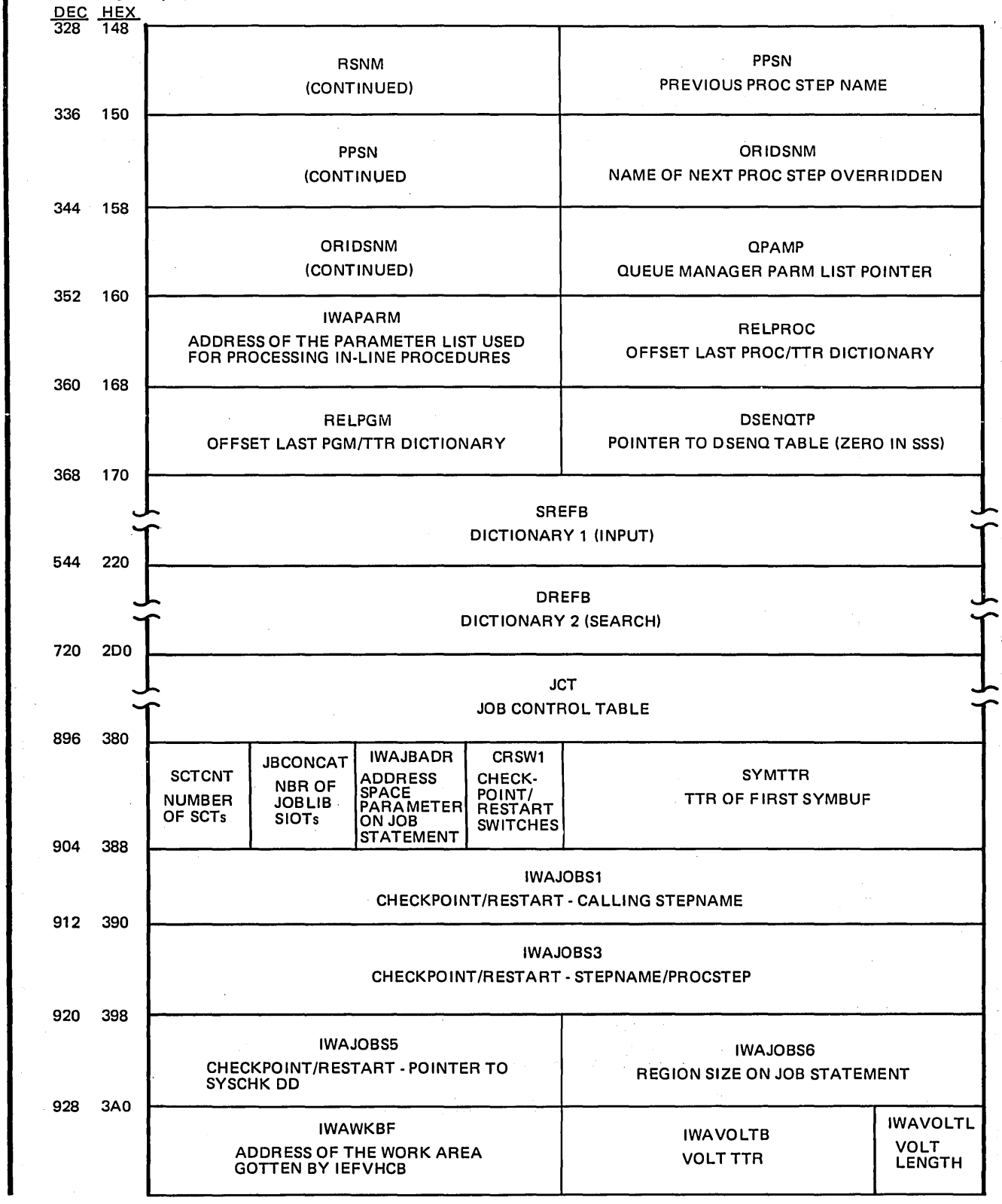
IWA Storage Map (Contd.)

DEC	HEX				
96	60	JACTS JACT POINTER (LWA)	SIOTS SIOT POINTER (LWA)	JFCBS JFCB POINTER (LWA)	JFCBXS JFCBX POINTER (LWA)
104	68	VOLTS VOLT POINTER (IWA)	DSNAMES DSNAME POINTER (IWA)	SREFBS DICTIONARY 1 POINTER (IWA)	DREFBS DICTIONARY 2 POINTER (IWA)
112	70	POVRRDS POVRRD POINTER (IWA)	ACTS ACT POINTER (IWA)	IWAPTTRL SAVE AREA FOR PROC STATEMENT (TTRL)	
120	78	IWAJTTRL SAVE AREA FOR JCL STATEMENT TTRL		IWAPRCWA POINTER TO PROC WORK AREA	
128	80	IWALREC POINTER TO LOGICAL RECORD BEING PROCESSED		IWATIOTL LENGTH OF INTERPRETER TIOT	IWAPICBL PROCLIB INPUT CONTROL BLOCK LENGTH
136	88	IWAENDAD ENDING TTR FOR PROCEDURE		IWAQMPA3 POINTER TO QMPA3 USED TO WRITE JMR	
144	90	IWABRGSV FIRST OF CONSECUTIVE REGISTER SAVE AREAS.		IWAERGSV LAST OF CONSECUTIVE REGISTER SAVE AREAS	
152	98	IWAUJVP SMF USER EXIT POINTER		IWAGRBG FIELD REQUIRED FOR JECS	
160	A0	IWAJLRCB JCL LRCB POINTER		IWAPLRCB PROC LRCB POINTER	
168	A8	IWASLRCB SYSTEM MESSAGE LRCB POINTER		IWAOLRCB SYSOUT LRCB POINTER	
176	B0	IWAILRCB INSTREAM PROCEDURE LRCB POINTER		IWA INTIO SAVE AREA FOR POINTER TO INITIATOR'S TIOT	
184	B8	IWACCODE AREA FOR CONDITION CODE		IWAEOBUF END OF RECORDS IN DATA AREA	
192	C0	IWAINTSO MASTER REGISTER SAVE AREA POINTER		IWAPDCBP PROCLIB DCB POINTER	

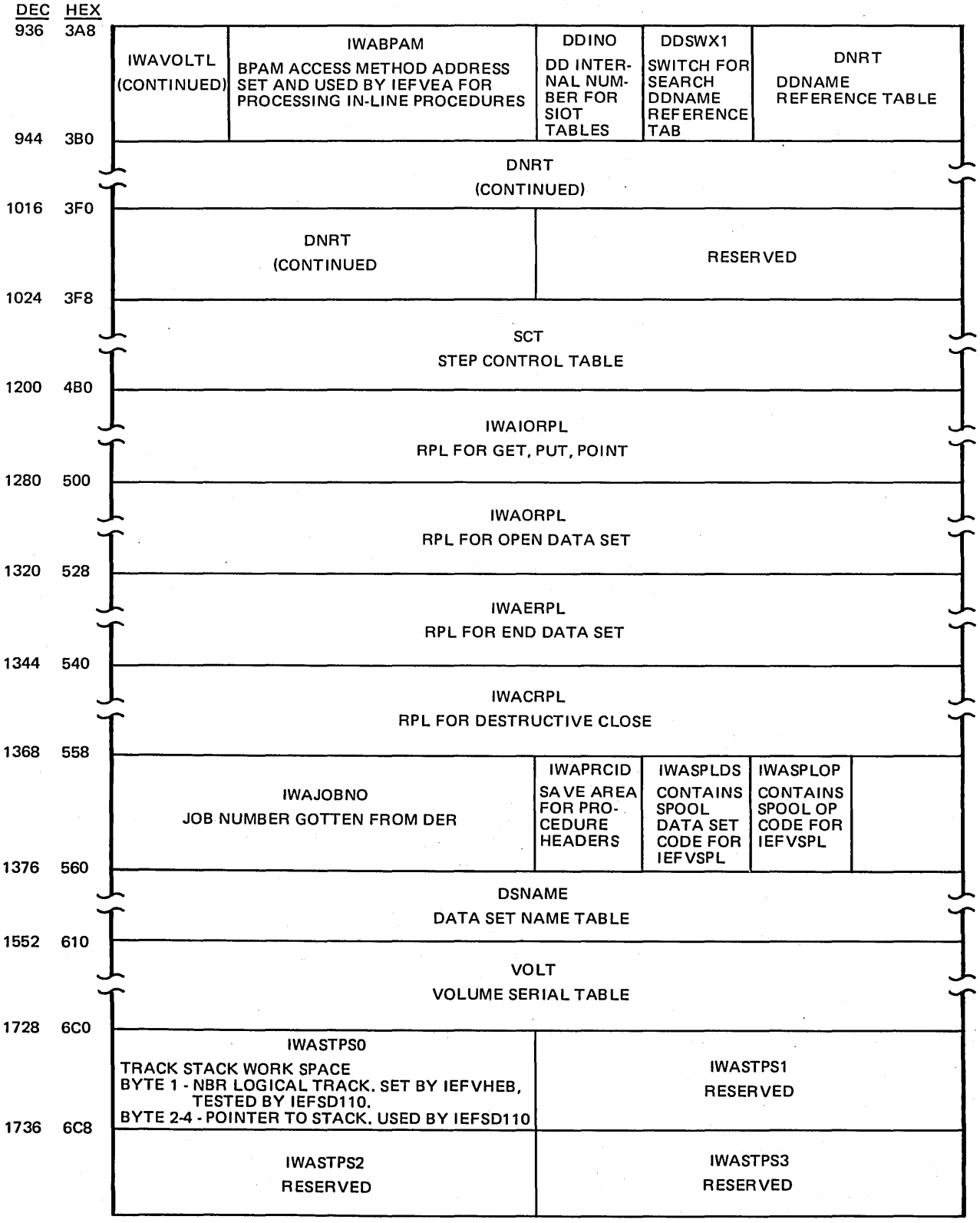
IWA Storage Map (Contd.)



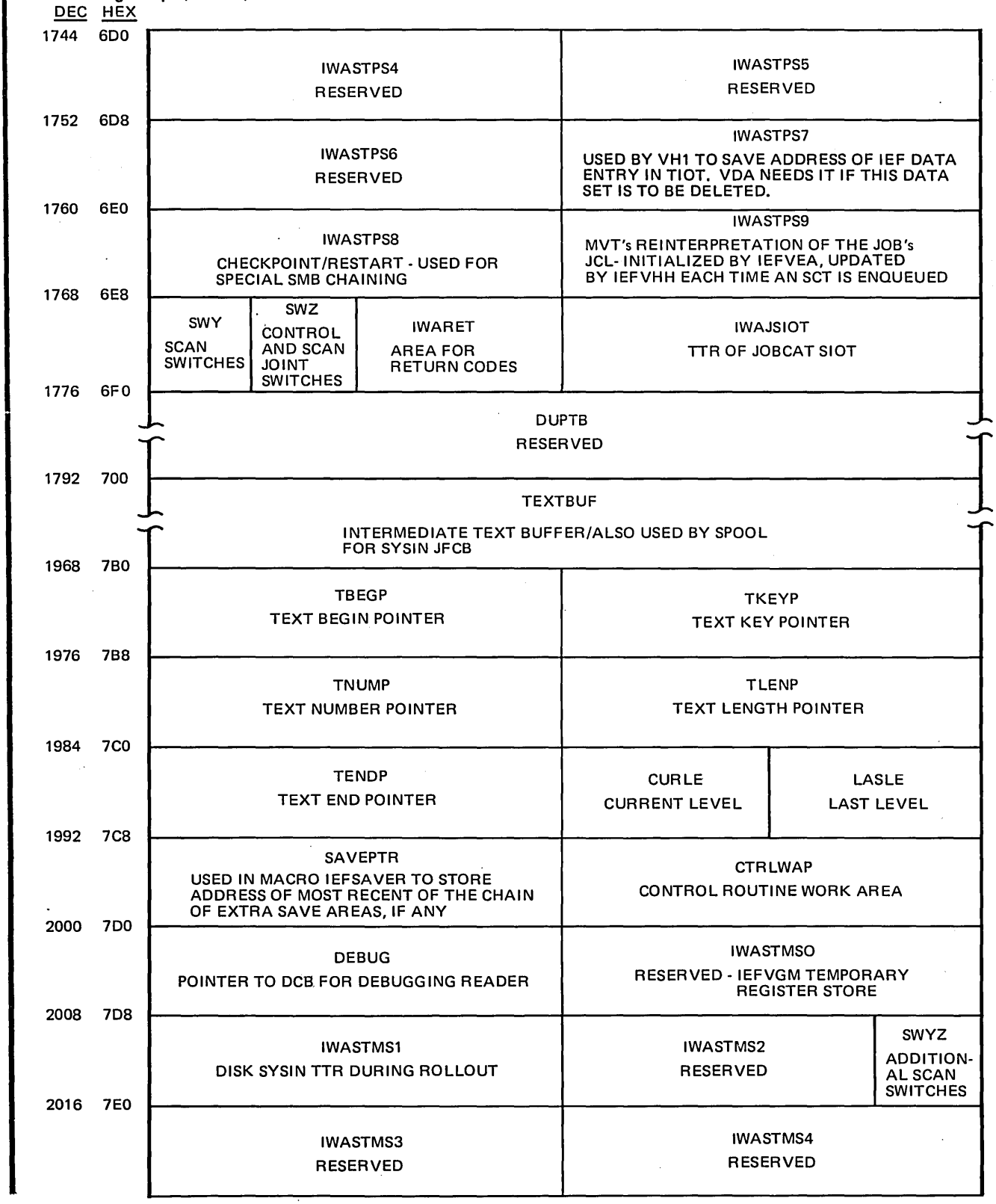
IWA Storage Map (Contd.)



IWA Storage Map (Contd.)



IWA Storage Map (Contd.)



IWA Storage Map (Contd.)

DEC	HEX							
2024	7E8	IWASTMS5 RESERVED				IWANELJC NEL POINTER FIELD. STORED BY IEFVH1. USED BY IEFVFA FOR INPUT TO POST SCAN ROUTINE		
2032	7F0	IWASTMS7 RESERVED	IWANELEN LENGTH OF NEL	IWAPCV PRIORITY CHANGE VALUE	IWAJDALL DEFAULT ALLOCA-TION LEVEL IN MSGLEVEL	IWAJDJCL DEFAULT JCL LEVEL IN MSGLEVEL	IWAMSLEN LENGTH OF MESSAGE FOR MSGLEVEL	IWAMCSCA MCS COMMAND AUTHORITY
2040	7F8	IWACONID MCS POINTER TO CONSOLE ID				RESERVED		

Fields in IWA — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IWAL	256	100	TNEXT
4	4	IWAID	264	108	TSIOT
8	8	IWASL	268	10C	TJOBLIB
8	8	IWAEXTS	272	110	TSREFB
9	9	IWAFINDP	276	114	TACT
12	C	IWADERPT	280	118	TPROC
16	10	IWASONUM	284	11C	RSTMT
18	12	IWASNUMB	291	123	SWV
19	13	IWAUNPK	292	124	PSTMT
24	18	OSW1	300	12C	PDNM
25	19	DINPRTY	308	134	PSNM
26	1A	DTIME	316	13C	RDNM
29	1D	DPQTY	324	144	RSNM
32	20	DSQTY	332	14C	PPSN
35	23	DINMMEM	340	154	ORIDSNM
37	25	OSW2	348	15C	QPAMP
38	26	DINBPLP1	352	160	IWAPARM
39	27	DUNAME	356	164	RELPROC
47	2F	DADDRFLT	360	168	RELPGM
48	30	DINTPPRI	364	16C	DSENQTP
52	34	UNQNAME	368	170	SREFB
75	4B	IWACATCT	544	220	DREFB
76	4C	UNNU	720	2D0	JCT
78	4E	DJBCLAS	896	380	SCTCNT
79	4F	DMSCLAS	897	381	JBCONCAT
80	50	QMGRP	898	382	IWAJBADR
84	54	SWA	899	383	CRSW1
85	55	SWB	900	384	SYMTTR
86	56	SWC	904	388	IWAJOBS1
87	57	SWD	912	390	IWAJOBS3
88	58	SWE	920	398	IWAJOBS5
89	59	SWF	924	39C	IWAJOBS6
90	5A	JEDSWS	928	3A0	IWAWKBF
92	5C	JCTS	932	3A4	IWAVOLTB
94	5E	SCTS	935	3A7	IWAVOLTL
96	60	JACTS	937	3A9	IWABPAM
98	62	SIOTS	940	3AC	DDINO
100	64	JFCBS	941	3AD	DDSWX1
102	66	JFCBXS	942	3AE	DNRT
104	68	VOLTS	1024	3F8	SCT
106	6A	DSNAMES	1200	4B0	IWAIORPL
108	6C	SREFBS	1280	500	IWAORPL
110	6E	DREFBS	1320	528	IWAERPL
112	70	POVRRDS	1344	540	IWACRPL
114	72	ACTS	1368	558	IWAJOBNO
116	74	IWAPTTRL	1372	55C	IWAPRCID
120	78	IWAJTTRL	1373	55D	IWASPLDS
124	7C	IWAPRCWA	1374	55E	IWASPLOP
128	80	IWALREC	1376	560	DSNAME
132	84	IWATIOTL	1552	610	VOLT
134	86	IWAPICBL	1728	6C0	IWASTPS0
136	88	IWAENDAD	1728	6C4	IWASTPS1
140	8C	IWAQMPA3	1736	6C8	IWASTPS2
144	90	IWABRGSV	1740	6CC	IWASTPS3
148	94	IWAERGSV	1744	6D0	IWASTPS4
152	98	IWAUJVP	1748	6D4	IWASTPS5
156	9C	IWAGRBG	1752	6D8	IWASTPS6
160	A0	IWAJLRCB	1756	6DC	IWASTPS7
164	A4	IWAPLRCB	1760	6E0	IWASTPS8
168	A8	IWASLRCB	1764	6E4	IWASTPS9
172	AC	IWAOLRCB	1768	6E8	SWY
176	B0	IWAILRCB	1769	6E9	SWZ
180	B4	IWAINTIO	1770	6EA	IWARET
184	B8	IWACCODE	1772	6EC	IWAJSIOT
188	BC	IWAEOBUF	1776	6F0	DUPTB
192	C0	IWAINTSO	1792	700	TEXTBUF
196	C4	IWAPDCBP	1968	7B0	TBEGP
200	C8	IWAQMPA2	1972	7B4	TKEYP
204	CC	IWAQMWKA	1976	7B8	TNUMP
208	D0	IWAINTS5	1980	7BC	TLENP
212	D4	IWAINPRM	1984	7C0	TENDP
216	D8	IWAINTS7	1988	7C4	CURLE
217	D9	SWG	1990	7C6	LASLE
218	DA	IWAINTS8	1992	7C8	SAVEPTR
219	DB	SWI	1996	7CC	CTRLWAP
220	DC	QPARM	2000	7D0	DEBUG

Fields in IWA — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
2004	7D4	IWASTMS0	2032	7F0	IWASTMS7
2008	7D8	IWASTMS1	2033	7F1	IWANELEN
2012	7DC	IWASTMS2	2034	7F2	IWAPCV
2015	7DF	SWY2	2035	7F3	IWAJDALL
2016	7E0	IWASTMS3	2036	7F4	IWAJDJCL
2020	7E4	IWASTMS4	2037	7F5	IWAMSLEN
2024	7E8	IWASTMS5	2038	7F6	IWAMCSCA
2028	7EC	IWANELJC	2040	7F8	IWACONID

Alphabetical list of fields in IWA

FIELD	DEC	HEX	FIELD	DEC	HEX
ACTS	114	72	IWAQMPA3	140	8C
CRSW1	899	383	IWAQMWKA	204	CC
CTRLWAP	1996	7CC	IWARET	1770	6EA
CURLE	1988	7C4	IWASL	8	8
DADDRFLT	47	2F	IWASLRCB	168	A8
DDINO	940	3AC	IWASNUMB	18	12
DDSWX1	941	3AD	IWASONUM	16	10
DEBUG	2000	7D0	IWASPLDS	1373	55D
DINBPLP1	38	26	IWASPLOP	1374	55E
DINMMEM	35	23	IWASTMSO	2004	7D4
DINPRTY	25	19	IWASTMS1	2008	7D8
DINTPPRI	48	30	IWASTMS2	2012	7DC
DJBCLAS	78	4E	IWASTMS3	2016	7E0
DMSCLAS	79	4F	IWASTMS4	2020	7E4
DNRT	942	3AE	IWASTMS5	2024	7E8
DPQTY	29	1D	IWASTMS7	2032	7F0
DREFB	544	220	IWASTPSO	1728	6C0
DREFBS	110	6E	IWASTPS1	1728	6C4
DSENQTP	364	16C	IWASTPS2	1736	6C8
DSNAME	1376	560	IWASTPS3	1740	6CC
DSNAMES	106	6A	IWASTPS4	1744	6D0
DSQTY	32	20	IWASTPS5	1748	6D4
DTIME	26	1A	IWASTPS6	1752	6D8
DUPTB	1776	6F0	IWASTPS7	1756	6DC
DUNAME	39	27	IWASTPS8	1760	6E0
IAWJOBS5	920	398	IWASTPS9	1764	6E4
IWABPAM	937	3A9	IWATIOTL	132	84
IWABRGSV	144	90	IWAUJVP	152	98
IWACATCT	75	4B	IWAUNPK	19	13
IWACCODE	184	B8	IWAVOLTB	932	3A4
IWACONID	2040	7F8	IWAVOLTL	935	3A7
IWACRPL	1344	540	IWAWKBF	928	3A0
IWADERPT	12	C	JACTS	96	60
IWAENDAD	136	88	JBCONCAT	897	381
IWAEOBUF	188	BC	JCT	720	2D0
IWAERGSV	148	94	JCTS	92	5C
IWAERPL	1320	528	JEDSWS	90	5A
IWAEXTS	8	8	JFCBS	100	64
IWAFINDP	9	9	JFCBXS	102	66
IWAGRBG	156	9C	LASLE	1990	7C6
IWAID	4	4	ORIDSNM	340	154
IWAILRCB	176	B0	OSW1	24	18
IWAINPRM	212	D4	OSW2	37	25
IWAINTIO	180	B4	PDNM	300	12C
IWAINTSO	192	C0	POVRRDS	112	70
IWAINTS5	208	D0	PPSN	332	14C
IWAINTS7	216	D8	PSNM	308	134
IWAINTS8	218	DA	PSTMT	292	124
IWAIORPL	1200	4B0	QMGRP	80	50
IWAJBADR	898	382	QPAMP	348	15C
IWAJDALL	2035	7F3	QPARM	220	DC
IWAJDJCL	2036	7F4	RDNM	316	13C
IWAJLRCB	160	A0	RELPGM	360	168
IWAJOBNO	1368	558	RELPROC	356	164
IWAJOBS1	904	388	RSNM	324	144
IWAJOBS3	912	390	RSTMT	284	11C
IWAJOBS6	924	39C	SAVEPTR	1992	7C8
IWAJSIOT	1772	6EC	SCT	1024	3F8
IWAJTTRL	120	78	SCTCNT	896	380
IWAL	0	0	SCTS	94	5E
IWALREC	128	80	SIOTS	98	62

Alphabetical list of fields in IWA

FIELD	DEC	HEX	FIELD	DEC	HEX
IWAMCSCA	2038	7F6	SREFB	368	170
IWAMSLEN	2037	7F5	SREFBS	108	6C
IWANELEN	2033	7F1	SWA	84	54
IWANELJC	2028	7EC	SWB	85	55
IWAOLRCB	172	AC	SWC	86	56
IWAORPL	1280	500	SWD	87	57
IWAPARM	352	160	SWE	88	58
IWAPCV	2034	7F2	SWF	89	59
IWAPDCBP	196	C4	SWG	217	D9
IWAPICBL	134	86	SWI	219	DB
IWAPLRCB	164	A4	SWV	291	123
IWAPRCID	1372	55C	SWY	1768	6E8
IWAPRCWA	124	7C	SWY2	2015	7DF
IWAPTTRL	116	74	SWZ	1769	6D9
IWAQMPA2	200	C8	SYMTTR	900	384
TACT	276	114	TNUMP	1976	7B8
TBEGP	1968	7B0	TPROC	280	118
TENDP	1984	7C0	TSIOT	264	108
TEXTBUF	1792	700	TSREFB	272	110
TJOBLIB	268	10C	UNNC	76	4C
TKEYP	1972	7B4	UNQNAME	52	34
TLENP	1980	7BC	VOLT	1552	610
TNEXT	256	100	VOLTS	104	68

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
OSW2	OPTION SWITCHES FROM PARM FIELD	IWAALTPL	'80'	BIT-0 READER HAS SPOOLED FROM AN ALTERNATE PROCLIB
		IWASTAE	'40'	STAE HAS ALREADY BEEN ISSUED
		IWADMSCT	'10'	DUMMY SCT REQUIRED - JCL ERROR
		CMAUTH	'03'	BITS 6, 7 COMMAND AUTHORIZATION
SWA	SWITCH A	JTOP	'80'	BIT-0 JOB TO PROCESS
		JHS	'40'	BIT-1 JOB HAS A STEP
		JCTTQ	'20'	BIT-2 JCT TO PUT IN QUEUE
		SCTTQ	'10'	BIT-3 SCT TO PUT IN QUEUE
		DFSH	'08'	BIT-4 DATA FLUSH
		JFSH	'04'	BIT-5 JOB FLUSH
		EOFR	'02'	BIT-6 EOF RECEIVED
		SAFSH	'01'	BIT-7 FLUSH TO A /*
SWB	SWITCH B	CXP	'80'	BIT-0 CONTINUATION EXP BY SCAN
		CXPN	'40'	BIT-1 CONT EXP & NOT RECEIVED
		CXPC	'20'	BIT-2 CONT EXP & CANCELLED
		CANDD	'10'	BIT-3 CANNED DD *
		DDAST	'08'	BIT-4 DD * OR DD DATA
		DDATA	'04'	BIT-5 DD DATA
		FRCV	'02'	BIT-6 1ST STMT RECEIVED
		SFJN	'01'	BIT-7 SEARCH FOR JOBNAME
SWC	SWTICH C	JCTRTN	'80'	BIT-0 CSCB RETURN
		IDERR	'40'	BIT-1 I/O ERROR ON INPUT
		NRCV	'20'	BIT-2 RDR NULL RECEIVED
		PEXP	'10'	BIT-3 PROCEDURE EXEC STMT EXPEC
		VOLTQ	'08'	BIT-4 VOLTABLE TO PUT INI QUEUE
		DSNTQ	'04'	BIT-5 DSN TABLE TO PUT IN QUEUE
		IWAFSM0	'02'	BIT-6 ON=FIRST ATTEMPT TO OPEN SYSTEM MSG DATA SET
		QMERR	'01'	BIT-7 Q MGR I/O ERROR
SWD	SWTICH D	JOBADDRF	'80'	BIT-0 ADDRSPC ON JOB STMT
		JCBREGNS	'40'	BIT-1 REGION ON JOB STMT
		FEXRCV	'20'	BIT-2 1ST EXEC RCVD THIS JOB
		FDDRCV	'10'	BIT-3 FIRST DD RCVD THIS STEP
		DBFST	'08'	BIT-4 FIRST ENTRY TO DSENQ
		DBLST	'04'	BIT-5 LAST ENTRY TO DSENQ
		DCTFST	'02'	BIT-6 FIRST DICT ENTRY RCVD
		SYMPRC	'01'	BIT-7 FIRST ACCESS OF PROC AAAA

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SWE	SWITCH E PROCEDURE SWITCHES	PROC	'80'	BIT-0 PROCLIB BEING USED
		GPI	'40'	BIT-1 GET PROCLIB INPUT
		PREF	'20'	BIT-2 PROCLIB EOF
		PRCV	'10'	BIT-3 PRIME PROC BUFFER
		CONCAT	'08'	BIT-4 CONCATENATION IN MERGE
		POVRD	'04'	BIT-5 OVERRIDE PROC DD STMT
		POVRX	'02'	BIT-6 OVERRIDE PROC EXEC STMT
		SEQUENCE	'01'	BIT-7 USED FOR CHECKING PROPER SEQUENCE OF ADDITIONS TO PROC STEPS
SWF	SWTICH F PAROCEDURE SWITCHES	ORPARMOR	'80'	BIT-0 PARM OVERRIDE
		ORPARMBL	'40'	BIT-1 PARM BLOCK
		ORCONDOR	'20'	BIT-2 COND OVERRIDE
		ORTIMEOR	'10'	BIT-3 TIME OVERRIDE
		ORTIMEO	'08'	BIT-4 TIME ZERO
		ORACTOR	'04'	BIT-5 ACCT OVERRIDE
		ORREGOR	'02'	BIT-6 REGION OVERRIDE
		ORADDROR	'01'	BIT-7 ADDRSPC OVERRIDE
SWH	FLAGS	IWASMDS	'80'	BIT-0 ON=SYSTM MSG DATA SET OPEN
		RDRDCBO	'40'	BIT-1 RDR OPENED
		PROCDCBO	'20'	BIT-2 PROCLIB OPENED
		IWASCLDS	'40'	BIT-1 ON=JCL DATA SET OPEN
		IWAPRCDS	'20'	BIT-2 ON=PROC DATA SET OPEN
		CPSYSFLG	'10'	BIT-3 C/R EXEC STMT
		CPFLGXX	'08'	BIT-4 C/R RESERVED
		PROCSW	'04'	BIT-5 STMT INVOKES PROC
		CPSTPFL	'02'	BIT-6 C/R STEP FLUSH
		PCPSYSIN	'01'	BIT-7 PCP SYSIN DD *
		IWASSODS	'01'	BIT-7 ON=SYSOUT DATA SET IS OPEN
SWG	ADDITIONAL PROCEDURE SWITCHES	ORRDOR	'80'	RD OVERRIDE
		ORSDPOR	'40'	STEP DISPATCHING PRIORITY OVERRIDE-SET AND TESTED IN IEFVEA
		DLMSW	'20'	BIT-2=1 IF DLM KEYWORD IS ON STMT. SET BY IEFVDA, TESTED BY VHG AND VHA.
		NOSYSN	'10'	NO IEFDATA CARD RECEIVED IN RDR PROC
IWAINTS8	FLAGS FOR CHKPT/RESTART	JOBRDNR	'20'	BIT-2 RD= NR
		JOBRDNC	'10'	BIT-3 RD=NC OR RD=RNC
		JOBRDR	'08'	BIT-4 RD=R OR RD=RNC
SWI	GENERAL SWITCHES	BLCPRC	'80'	BIT-0 BLOCK PROCLIB FLAG
		IWABAS	'40'	BIT-1 BYPASS ASSIGN START
		IWADDNM	'20'	BIT-2 DDNAME= KEY THIS CARD
		IWAKGSW	'10'	BIT-3 BLOCKED PROC
		BLKMLTER	'08'	BIT-4 PROCLIB BLKSIZE NON-MULTIPLE OF 80
		DSNLIT	'04'	BIT-5 DSN='LITERAL'
		SPOOL	'02'	BIT-6 DATA SET HAS BEEN SPOOLED
		SPOOLDD	'01'	BIT-7 DD * OR DATA INDICATOR. SET BY IEFVFA, TESTED BY IEFVDA
RSTMT+4	VERB NUMBER	SSE	'80'	DD STATEMENT SEQUENCE ERROR
		CRE	'40'	MISPLACED SYSCHK DD STATEMENT

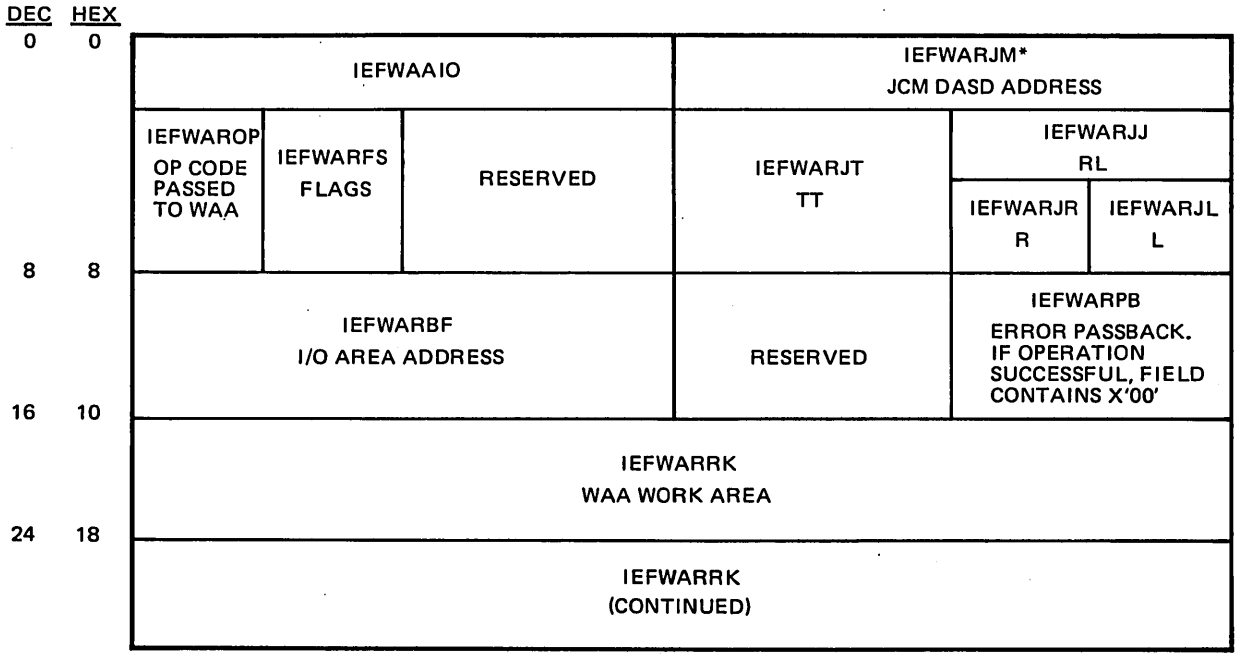
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SWV	SCAN SWITCHES	OVKEYSWZ	'123'	IF ON, THIS KEY OVERRIDEN
		OVKEYSW	'80'	
		PROCERRZ	'123'	IF ON, ERROR THIS STATEMENT
		PROCERR	'40'	
		VERBCSWZ	'123'	
		VERBCSW	'20'	
		FBFLUSHZ	'123'	SYMBOLIC PARM FLUSH INDICATOR
		FBFLUSH	'10'	
		AMPSWZ	'123'	IF ON, TEXT DEFINES A SYMBOLIC PARAMETER
		AMPSW	'08'	
		FBLITRLZ	'123'	IF ON, TEXT IS LITERAL (ENCLOSED IN QUOTES)
		FBLITRL	'04'	
		FPRNSWZ	'123'	IF ON, INDICATES LEFT PARENTHESIS
		FPRNSW	'02'	
		LPBYSWZ	'123'	IF ON, BYPASS LEFT PARENTHESIS CHECK
		LPBYSW	'01'	
		CLEARVZ	'123'	
		CLEARV	'FE'	OVKEYSW + PROCERR + VERBCSW + FBFLUSH + AMPSW + FBLITRL + FPRNSW
PSTMT	ADDRESS OF JCL DATA SET RECORD	VERB	'04'	OFFSET FOR VERB BYTE
		NAMEL	'05'	OFFSET FOR NAME LENGTH
		LISTPTR	'06'	OFFSET FOR RELATIVE LIST POINTER
IWAPARM	ADDRESS OF PARAMETER LIST USED FOR PROCESSING IN-LINE PROCEDURES	IWAINSTP	'80'	FIRST BYTE OF POINTER USED TO INDICATE INSTREAM PROCEDURE BEING EXPANDED
CRSW1	CHECKPOINT RESTART SWITCHES	CPFLG	'80'	GET/FREE SYSCHK DD STATEMENT CORE
		CPDUM	'20'	DUMMY SCT REQUIRED STEP FLUSH
		CRIMRS	'01'	PCP-IMMEDIATE RESTART
DDSWX1	SWITCH FOR SEARCH DDNAME REF TABLE	SDDNSW	'80'	
		IWAJCAT	'20'	JOBCAT SWITCH
		IWASCAT	'10'	STEPCAT SWITCH
		IWASCAT2	'08'	STEPCAT FOUND BIT
		SYSUTSW	'40'	BIT FOR SYSOUT SWITCH
SWZ	CONTROL AND SCAN JOINT SWITCHES	CMT	'80'	COMMENT SWITCH
		DDOV	'40'	DD OVERRIDE SIWTCH
		ENDS	'20'	END SCAN SWITCH
		COLST	'10'	COLUMN 72 SWITCH
		JOBSW	'08'	JOB SWITCH
		EXECSW	'04'	EXEC SWITCH
		DDSW	'02'	DD SWITCH
		SNPSW	'01'	STATEMENT SYSOUT SWITCH
SWY2	ADDITIONAL SCAN SWITCHES	DUMMYSW	'80'	BIT 0=1 IF DUMMY POSITIONAL KEYWORD OF STMT. SET BY IEFVFA, TESTED BY IEFVDA
		DYNAMSW	'40'	BIT 1=1 IF DYNAM POSITIONAL KEYWORD ON STMT. SET BY IEFVFA, TESTED BY IEFVDA
		KEYNXTSW	'20'	BIT 2=1 IF FIRST EQUAL ENCOUNTERED, NO MORE POSITIONAL PARAMETER EXPECTED.

I/O Work Area Allocation Parameter List (WAAIO)

The requester builds the work area allocation parameter list for an I/O request (WAAIO) and passes it to IEFWAALC to indicate that a read or write of a JCM is required. WAAIO requires 32 bytes of storage.

WAAIO Storage Map



* The DASD address of a record is in the form of TTRL where:

TT	is the track address relative to the beginning of SYS1.SYSPOOL data set.
R	is the block number on track TT.
L	is the logical record number in block R.

Fields in WAAIO — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFWAAIO	6	6	IEFWARJR
0	0	IEFWAROP	7	7	IEFWARJL
1	1	IEFWARFS	8	8	IEFWARBF
4	4	IEFWARJM	14	E	IEFWARPB
4	4	IEFWARJT	16	10	IEFWARRK
6	6	IEFWARJJ			

Alphabetical list of fields in WAAIO

FIELD	DEC	HEX	FIELD	DEC	HEX
IEFWAAIO	0	0	IEFWARJR	6	6
IEFWARBF	14	E	IEFWARJT	4	4
IEFWARFS	1	1	IEFWAROP	0	0
IEFWARJJ	4	4	IEFWARPB	14	E
IEFWARJL	7	7	IEFWARRK	16	10
IEFWARJM	4	4			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
IEFWAROP	OP CODE PASSED TO WAA	IEFWAARW	'08'	CODE TO READ OR WRITE JCM
IEFWARFS	FLAGS	IEFWAAR	'80'	READ JCM
		IEFWAAW	'40'	WRITE JCM
IEFWARPB	ERROR PASSBACK	IEFWARBD	'80'	FUNCTION NOT PERFORMED – INVALID TTRL
		IEFWARCR	'40'	FUNCTION NOT PERFORMED – WAA GETMAIN NOT SATISFIED
		IEFWARBW	'20'	FUNCTION NOT PERFORMED – JCM CANNOT BE WRITTEN BECAUSE OF BAD TRACK
		IEFWARBR	'10'	FUNCTION NOT PERFORMED – JCM CANNOT BE READ BECAUSE OF BAD TRACK
		IEFWARBX	'04'	RESERVED
		IEFWARX1	'02'	FUNCTION NOT PERFORMED – INVALID OP CODE
		IEFWARX2	'01'	FUNCTION NOT PERFORMED – SPOOL VOLUME IDENTIFICATION NOT POSSIBLE

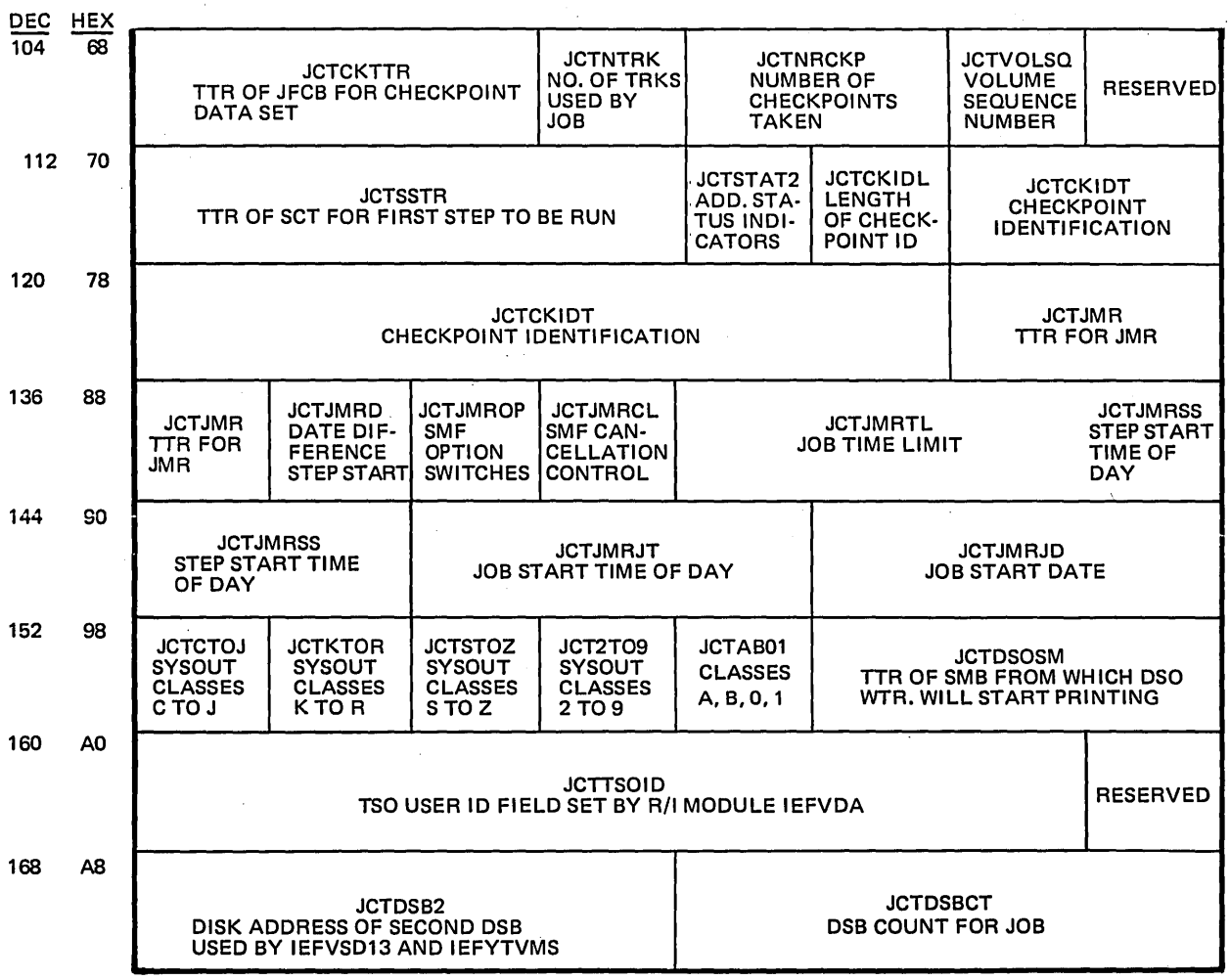
Job Control Table (JCT)

The Job Control Table (JCT) is 176 bytes long and contains information regarding the whole job. The interpreter module (IEFVJA) builds the JCT as part of the processing of the JOB statement. The JCT is deleted at JOB termination time.

JCT Storage Map

DEC	HEX						
0	0	JCTDSKAD DISK ADDRESS OF THIS JCT	JCTIDENT JCT ID=64	JCLJSRNO INTERNAL JOB SERIAL NUMBER	JCTJSTAT JOBLIB SWITCH BITS 0-3	JCTJMGPO MESSAGE CLASS	JCTJPRTY 4 BITS FOR JOB PRIORITY
8	8	JCTJNAME JOBNAME					
16	10	JCTJPTN T/P TERMINAL NAME					
24	18	JCTPDQDA DISK ADDRESS OF PDQ		JCTBCTDA DISK ADDRESS OF BIAS CT TABLE			
32	20	JCTSDKAD DISK ADDRESS OF FIRST SCT		JCTSMBAD DISK ADDRESS OF FIRST SMB OR OF THE FIRST DSB FOR VS1			
40	28	JCTACTAD DISK ADDRESS OF FIRST ACT		JCTDSSBA TTR OF FIRST SYSOUT CLASS DIRECTORY (SCD), CREATED AT INTERPRETER TIME, USED BY INITIATOR			
48	30	JCTDSBAD DISK ADDRESS OF LAST DATA SET B		JCTSMBID KEY (OF TRACK ID) FOR SMBS	JCTJDPCD DEPENDENCY CODE		
56	38	JCTJDPOP DEPENDENCY OPERATOR	SPACE FOR SEVEN MORE DEP CODES (28 BYTES)				
80	50				JCTRSW1 CHKPT RESTART SWITCHES	JCTRSW2 CHKPT RESTART SWITCHES	
88	58	JCTDETDA TTR OF DISENQ TABLE		JCTEQREG REGION PARAMETER	JCTQIDNT Q FOR JOB (MVT)	JCTSNUMB NUMBER OF STEPS RUN (MVT)	
96	60	JCTSTIOT TTR OF COMPRESSED TIOT (MVT ONLY)		JCTDEVT DEVICE TYPE OF CHECKPOINT DATA SET			

JCT Storage Map (Contd.)



Fields in JCT — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	INJMJCT	54	36	JXRJSPXS	137	89	JCTJMRD
0	0	JCTDSKAD	56	38	JCTJDPOP	138	8A	JCTJMROP
3	3	JCTIDENT	85	55	JCTRSW1	139	8B	JCTJMRCL
4	4	JCTJSRNO	87	57	JCTRSW2	140	8C	JCTJMRTL
5	5	JCTJBLBS	93	5D	JCTQIDNT	143	8F	JCTJMRSS
6	6	JCTJMGPO	95	5F	JCTSNUMB	146	92	JCTJMRJT
7	7	JCTJMGLV	96	60	JCTSTIOT	149	95	JCTJMRJD
8	8	JCTJNAME	100	64	JCTDEVT	152	98	JCTCTOJ
16	10	JCTJTPTN	104	68	JCTCKTTR	153	99	JCTKTOR
24	18	JCTPDQDA	107	6B	JCTNTRK	154	9A	JCTSTOZ
28	1C	JCTBCTDA	108	6C	JCTNRCKP	155	9B	JCT2TO9
32	20	JCTSDKAD	110	6F	JCTVOLSQ	156	9C	JCTAB01
36	24	JCTSMBAD	112	70	JCTSSTR	157	9D	JCTDSOSM
40	28	JCTACTAD	116	74	JCTSTAT2	160	A0	JCTTSOID
44	2C	JCTDSSBA	117	75	JCTCKIDL	168	A8	JCTDSB2
48	30	JCTDSBAD	120	78	JCTCKIDT	172	AC	JCTDSBCT
52	34	JCTSNBID	134	86	JCTJMR			

Alphabetical list of fields in JCT

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
INJMJCT	0	0	* JCTJMGLV	7	7	JCTQIDNT	93	5D
JCTAB01	156	9C	JCTJMGPO	6	6	* JCTRSW1	85	55
JCTACTAD	40	28	JCTJMR	134	86	* JCTRSW2	87	57
JCTBCTDA	28	1C	JCTJMRCL	139	8B	JCTSDKAD	32	20
JCTCKIDL	117	75	JCTJMRD	137	89	JCTSMBAD	36	24
JCTCKIDT	120	78	JCTJMRJD	149	95	JCTSNBID	52	34
JCTCKTTR	104	68	JCTJMRJT	146	92	JCTSNUMB	95	5F
JCTCTOJ	152	98	JCTJMROP	138	8A	JCTSSTR	112	70
JCTDB2	168	A8	JCTJMRSS	143	8F	* JCTSTAT2	116	74
JCTDEVT	100	64	JCTJMRTL	140	8C	JCTSTIOT	96	60
JCTDSBAD	48	30	JCTJNAME	8	8	JCTSTOZ	154	9A
JCTDSBCT	172	AC	JCTJSRNO	4	4	JCTTSOID	160	A0
JCTDSKAD	0	0	JCTJTPTN	16	10	JCTVOLSQ	110	6E
JCTDSOSM	157	9D	JCTKTOR	153	99	JCT2TO9	155	9B
JCTDSSBA	44	2C	JCTNRCKP	108	6C	JXRJSPXS	54	36
* JCTIDENT	3	3	JCTNTRK	107	6B			
JCTJBLBS	5	5	JCTPDQDA	24	18			
JCTJDPOP	56	38						

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JCTIDENT	JCT ID=64	JCTID	0	
JCTJSTAT	BIT1-JOB FLUSH BIT	JCTJSTPC	32	BIT2-JOB STEP CANCELLED BIT3-STEP FLUSH BIT
		JCTABEND	8	BIT4-JCT ABEND BIT
		INCMSTS	4	BIT5=1-JOB FAILED BIT6=0-GO JOB
		INDMCTLL	2	BIT6=1 CATALOG JOB
		INCMCAT	2	BIT6-CATALOG BIT
		INCMNSET	1	BIT7-NON SETUP BIT7=0-SETUP JOB BIT7=1-NONSETUP JOB
JCTJMGLV	BITS FOR MESSAGE	INCMMGL1	16	JCL MESSAGE LEVEL=1 BIT
		INCMMGL2	32	JCL MESSAGE LEVEL=2 BIT
		INCMALL	128	ALLOC. MESSAGE LEVEL= 1 BIT
JCTRSW1	CHECKPOINT/RESTART SWITCHES	JCTWARMS	128	BIT0-WARM START
		JCTSTERM	64	STEP TERMINATION HAS BEGUN
		JCTCKFT	16	BIT3=CHECKPOINT THIS STEP
		JCTCKPTR	8	BIT4=CHECKPOINT RESTART
		JCTSTEPR	4	BIT5=STEP RESTART TO BE DONE BIT6&7-MUST BE ZERO
JCTRSW2	CHECKPOINT/RESTART	JCTSYSCK	128	BIT0-SYSCHK DD STATEMENT PRESENT
		JCTNORST	32	BIT2-NO RESTART TO BE DONE
		JCTNOCKP	16	BIT3-NO CHECKPOINTS TO BE TAKEN
		JCTRESTT	8	BIT4-DO RESTART IF NECESSARY
		JCTDSOCR	4	BIT5-DO ACTIVE AT CHECKPOINT
		JCTDSOJB	2	BIT6-DSOCB'S SELECTED FOR JOB
		JCTSDSRA	1	BIT7-DSDR PROCESSING HAS NOT SUCCEEDED
JCTSTAT2	ADDITIONAL STATUS	JCTSPSYS	128	BIT0=1-SPOOLED SYSIN FOR JOB
		JCTENDIT	32	JOB TERM. INDICATOR-SET BY IEFSD410
		JCTSWSM	16	BIT3=1-WARM START MESSAGE

Job Cylinder Map (JCM)

The Job Cylinder Map (JCM) header contains information reflecting the bit map and the number of logical cylinders present. The JCM is built by IEFWAA01 at initial open time and is deleted at either job termaination or job purge time by module IEFWAA02.

JCM Dictionary Entry

DEC	HEX
0	0

IEFJCMDK JCM DISK ADDRESS LENGTH OF ORIGINAL JCM IF SPOOL VOLUMES ADDED	IEFJCMCR JCM CORE ADDRESS

Job Cylinder Map (JCM) Header

DEC	HEX
0	0
8	8

JCMID JCM IDENTIFIER		JCMDISK JCM DISK ADDRESS - TTRL	
JCMLGTH JCM LENGTH	JCMRSCTR # OF LOGICAL CYLINDERS ALLOCATED TO THIS JCM	JCMCHCTR # OF LOGICAL CYLINDERS ALLOCATED TO JCM SINCE LAST CKPT.	JCMFLGS FLAGS

Fields in JCM --- By Displacement

DEC	HEX	LABEL	DEC	HEX	LABEL
0	0	IEFJCMDK	8	8	JCMLGTH
4	4	IEFJCMCR	10	A	JCMRSCTR
			12	C	JCMCHCTR
0	0	JCMID	14	E	JCMFLGS
4	4	JCMDISK			

Alphabetical List of Fields in JCM

LABEL	DEC	HEX	LABEL	DEC	HEX
IEFJCMCR	4	4	JCMFLGS	14	E
IEFJCMDK	0	0	JCMID	0	0
JCMCHCTR	12	C	JCMLGTH	8	8
JCMDISK	4	4	JCMRSCTR	10	A

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JCMFLGS	FLAGS	JCMCHNGD	80	JCM HAS CHANGED SINCE LAST CHECKPOINT
		JCMBDTRK	40	JCM CANNOT BE CHECKPOINTED - BAD TRACK
		JCMEXT	20	JCM POINTS TO AN EXTENDED JCM
		JCMWSCHK	10	SPOOL VOLUMES HAVE BEEN ADDED AT SYSTEM RESTART

Job Entry Subsystem Communication Table (JESCT)

The JESCT, which is 236 bytes long, is the communication table for Job Entry Subsystem. It is assembled at sysgen time and is loaded with the JES load module. It is never deleted. JESCT is modifiable at IPL time.

JESCT Storage Map

DEC	HEX					
0	0	JESREQ JECS ROUTER ENTRY POINT			JESWAM WORKAREA MANAGER ENTRY POINT	
8	8	JESWAA WORKAREA ALLOCATION ENTRY POINT			JESQMGR QUEUE MANAGER ENTRY POINT	
16	10	JESRESQM ENTRY POINT INTERFACES BETWEEN THE QMNGR IO MACRO AND THE RESIDENT Q/M			JESSMUT SPOOL MANAGER USER TABLE ADDRESS POINTER	
24	18	JESBFPOL BUFFER MANAGER BUFFER POOL ADDRESS POINTER			JESBFUT BUFFER MANAGER BUFFER USAGE TABLE POINTER	
32	20	JESIOSC INPUT/OUTPUT STREAM CONTROL PARAMETER AREA POINTER			JESWAMDA WORKAREA MANAGER DATA AREA ADDRESS POINTER	
40	28	JESALCVT SYSTEM LOG DER POINTER			JESRWECB JOB ENTRY PERIPHERAL SERVICES MONITOR ECB	
48	30	JESUNTST FLAG BYTE	JESUNTAD POINTER UNIT ADDRESS OF DEVICES TO BE SUPPORTED BY WAM FOR SYS1.SYSPOOL		JESJOBID UNIQUE JOB IDENTIFICATION	
56	38	JESSMFOP SMF OPT AT SYSGEN TIME	JESLOGOP LOG SYSGEN OPTION	JESJOBCM WAA JOB CYLINDER MAP LENGTH INDICATOR	JESBFSIZ BUFFER SIZE TO BE USED BY BUFFER MANAGER	JESBFNBR NO. OF BUFFERS TO BE INITIALIZED
64	40	JESWACAP BREAKING LIMIT (SPECIFIED AT SYSGEN AS A PERCENTAGE)		JESSWADS RECORDS USED BY Q-MGR TO HANDLE OVERFLOW OF SWADS	JESMWTP MESSAGES THE WTP ROUTINE ISSUES DURING A JOB STEP	JESNORDR NO. OF READERS
72	48	JESNOWTR NO. OF WRITERS		JESRESLK START RTAM RECEIVED FLAG RESERVED	JESWTBLK SUM OF ALL OUTPUT BLOCKSIZES	JESUSRTN AMT. OF MAIN STORAGE FOR USER-SUPPLIED ROUTINES
80	50	JESINCTL CONTROL UNIT INTERVAL TO BE USED FOR UNIT RECORD DEVICES		JESOUCTL CONTROL UNIT INTERVAL TO BE USED FOR OUTPUT DEVICES	JESRWSUP STORAGE FOR READERS AND WRITERS	
88	58	JESWTLRS NO. OF WTL RECORDS ISSUED BEFORE THE DATA SET IS AUTOMATICALLY CLOSED			JESOUTLI OUTPUT LIMIT FOR P/P SYSOUT	

JESCT Storage Map (Contd.)

DEC	HEX				
96	60	JESSMNBR NO. OF ENTRIES IN SPOOL MANAGER USER TABLE	JESUNITS UNIT ADDRESS FOR SYS1.SYSPOOL	JESUNITS UNIT ADDRESS FOR SYS1.SYSPOOL	
176	B0	JESUNITS	JESUNTDL DELIMITER FOR SYS1.SYSPOOL DEVICES	JESRESST RTAM FLAG BYTE	JESRESAD ADDRESS OF RESET
184	B8	JESLDPT ADDRESS OF JES RB		JESRDBLK SUM OF ALL INPUT BLOCKSIZES	
102	C0	JESQCTL ADDRESS OF JQCT		JESVMA JEPS ENTRY	
200	C8	JESVMB ISC ENTRY		JESOSC01 OSC ENTRY	
208	D0	JESOSC02 OSC ENTRY		JESMSGJP ADDRESS OF JEPS MESSAGE WRITER	
216	D8	JESAMAP ADDRESSES OF AMAP HOOKS IN JECS			
224	E0	JESAMAP (CONTINUED)		JESSMUT ECB LOCK	
232	E8	JESSMUT LOCK CONTROL WORD			

Fields in JESCT — By displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	JESREQ	48	30	JESUNTST	72	48	JESNOWTR	182	B6	JESRESAD
4	4	JESWAM	49	31	JESUNTAD	74	4A	JESRESLK	184	B8	JESLDPT
8	8	JESWAA	52	34	JESJOBID	76	4C	JESWTBLK	188	BC	JESRDBLK
12	C	JESQMGR	56	38	JESSMFOP	78	4E	JESUSRTN	192	C0	JESQCTL
16	10	JESRESQM	57	39	JESLOGOP	80	50	JESINCTL	196	C4	JESVMA
20	14	JESSMUT	58	3A	JESJOBCM	82	52	JESOUCTL	200	C8	JESVMB
24	18	JESBFPOL	60	3C	JESBFSIZ	84	54	JESRWSUP	204	CC	JESOSC01
28	1C	JESBFUT	62	3E	JESBFNBR	88	58	JESWTLRS	208	D0	JESOSC02
32	20	JESIOSC	64	40	JESWACAP	92	5C	JESOUTL1	212	D4	JESMSGJP
36	24	JESWAMDA	66	42	JESSWADS	96	60	JESSMNBR	216	D8	JESAMAP
40	28	JESALCVT	68	44	JESMWTP	98	62	JESUNITS	228	E4	JESSMUT
44	2C	JESRWECB	70	46	JESNORDR	178	B2	JESUNTDL	232	E8	JESSMUT

Alphabetical list of fields in JESCT

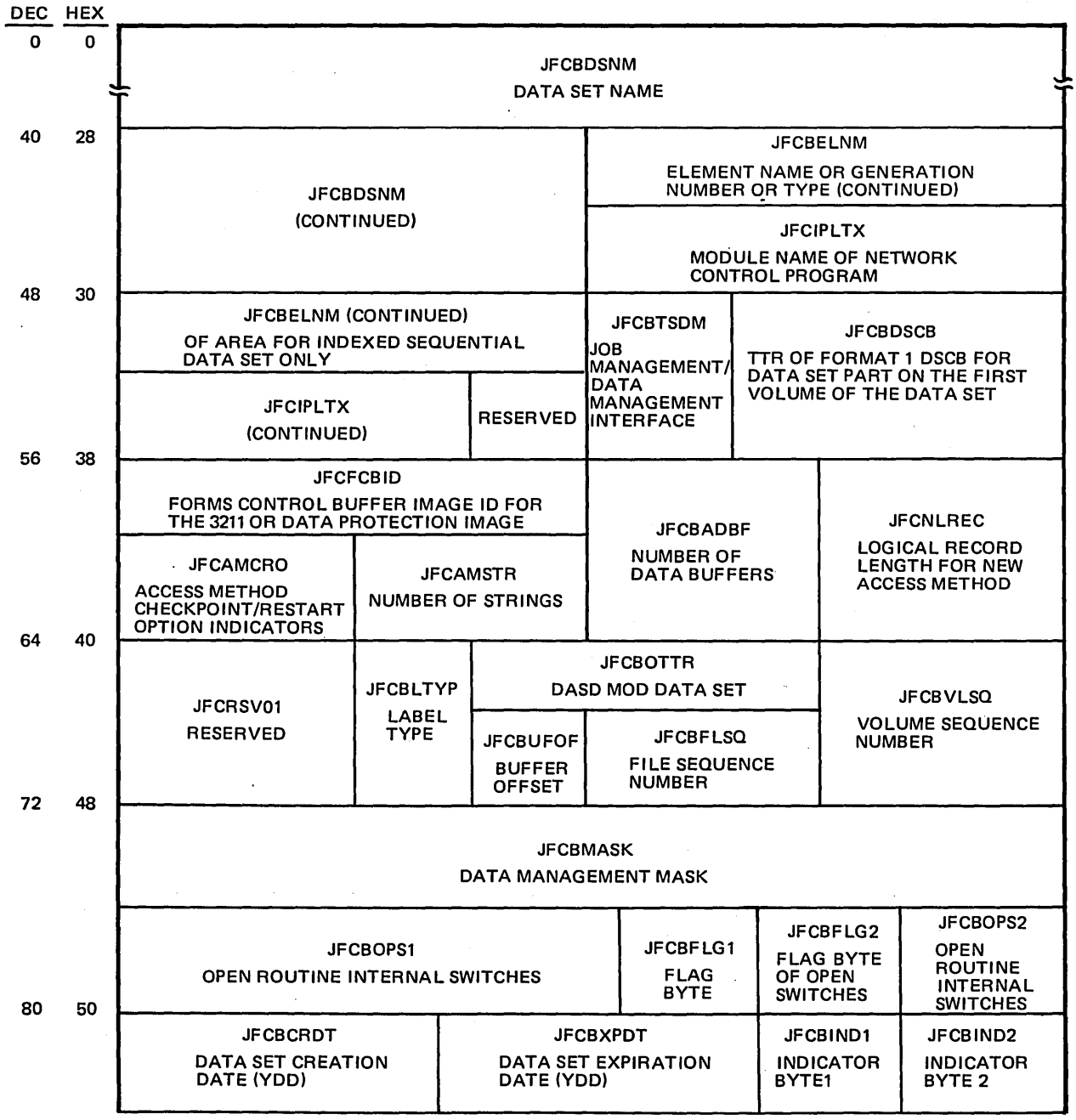
FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
JESAMAP	216	D8	JESMSGJP	212	D4	JESREQ	0	0	JESUNTDL	178	B2
JESBFNBR	62	3E	JESNORDR	70	46	JESRESQM	16	10	* JESUNTST	48	30
JESBFPOL	24	18	JESOSC01	204	CC	JESRWFCB	44	2C	JESUSRTN	4E	78
JESBFSIZ	60	3C	JESOSC02	208	D0	JESRWSUP	84	54	JESVMA	196	C4
JESBFUT	28	1C	JESQCTL	192	C0	* JESSMFOP	56	38	JESVMB	200	C8
JESINCTL	80	50	JESRDBLK	188	BC	JESSMNBR	96	60	JESWAA	8	8
JESIOSC	32	20	JESRESAD	182	B6	JESSMUT	20	14	JESWACAP	64	40
JESJOBCM	58	3A	JESNOWTR	72	48	JESSMUT	228	E4	JESWAM	4	4
JESJOBID	52	34	JESOUCTL	82	52	JESMUT	232	E8	JESWAMDA	36	24
JESLDPT	184	B8	JESOUTL1	92	5C	JESSWADS	66	42	JESWTBLK	76	4C
* JESLOGOP	57	39	JESQMGR	12	C	JESUNITS	98	62	JESWTLRS	88	58
JESMWTP	68	44	JESRDPLK	188	BC	JESUNTAD	49	31			

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JESRESST	RTAM FLAG BYTE			
JESUNTST	STATUS BYTE	JESUNTF	80	SPOOL FORMAT INDICATOR
		JESUNTC	40	SPOOL=CHANGE INDICATOR
		JESUNTR	01	ERROR DETECTED IN SPOOL VOL. PROCESSING
JESSMFOP	SMF OPT SPECIFIED SYSGEN TIME	JESMFBAS	02	SMF BASIC WAS SPECIFIED
		JESMFULL	01	SMF FULL WAS SPECIFIED
JESLOGOP	LOG OPT SPECIFIED AT SYSGEN TIME	JESLOGSW	01	LOG OPTION SWITCH BIT

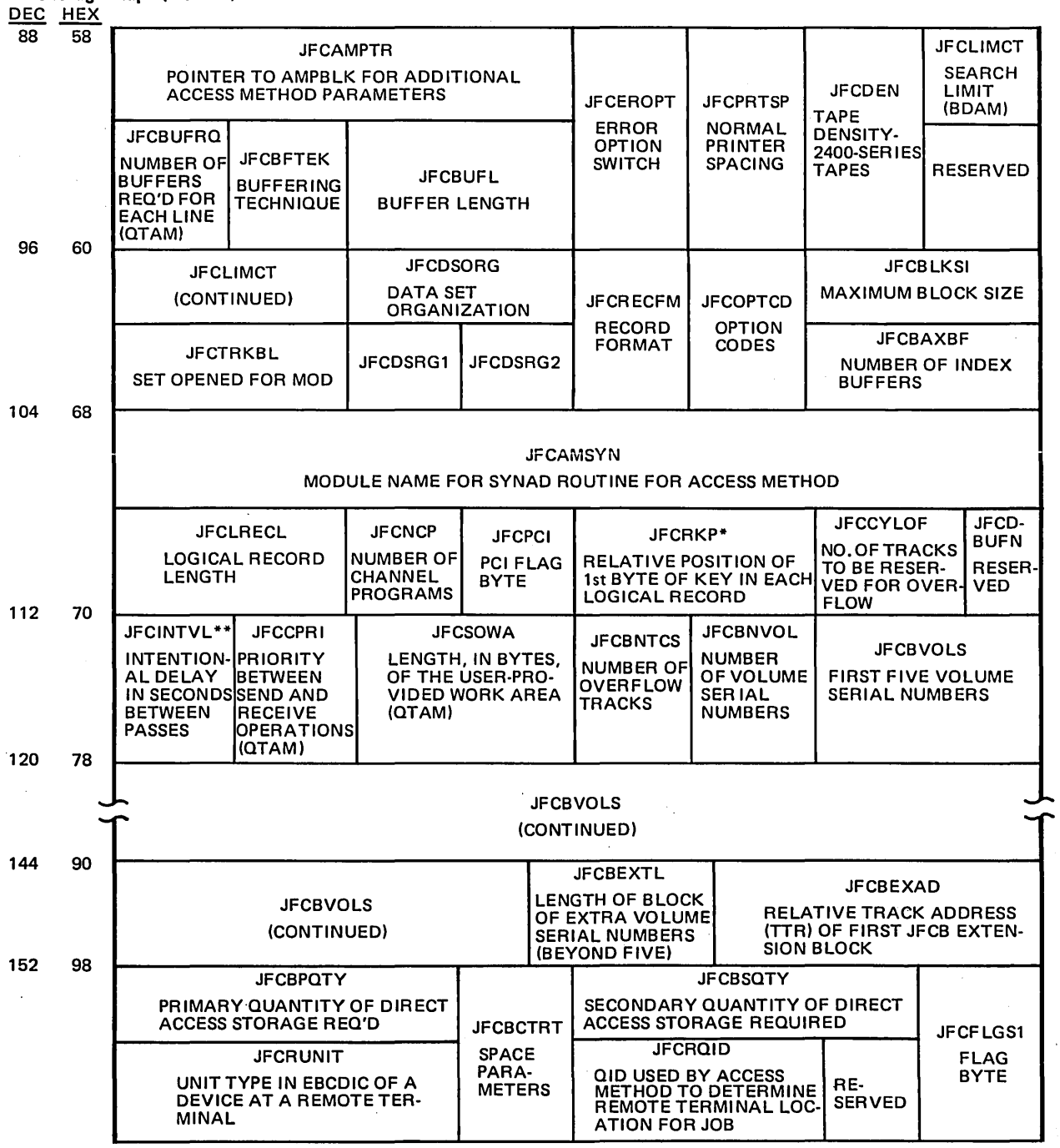
Job File Control Block (JFCB)

The Job File Control Block (JFCB) is built by the interpreter and contains some of the information specified on a DD statement. The JFCB, which requires 176 bytes of storage, is used by data management at open time and is deleted at job termination time.

JFCB Storage Map



JFCB Storage Map (Contd.)



* Normal 108 segment begins at this location (X'6C'). The 108 printer segment is shown at the end of this storage map.

** End of normal 108 segment.

JFCB Storage Map (Contd.)

DEC	HEX				
160	A0	JFCBDQTY QUANTITY OF DIRECT ACCESS STORAGE REQUIRED FOR A DIRECTORY OR AN EMBEDDED INDEX AREA	JFCBSPNM MAIN STORAGE ADDRESS OF THE JFCB WITH WHICH CYLINDERS ARE SPLIT		JFCBABST RELATIVE ADDRESS OF FIRST TRACK TO BE ALLOCATED
168	A8	JFCBSBNM MAIN STORAGE ADDRESS OF THE JFCB FROM WHICH SPACE IS TO BE SUBALLOCATED	JFCBDRLH AVERAGE DATA BLOCK LENGTH	JFCBVLCT VOLUME COUNT	JFCBSPTN NO. OF TRACKS PER CYLINDER TO BE USED FOR SPLIT CYLINDER

108 PRINTER SEGMENT
ORG '6C'

DEC	HEX				
108	6C	JFCRKP RELATIVE POSITION OF FIRST BYTE OF KEY WITHIN EACH LOGICAL RECORD	JFCCYLOF NUMBER OF TRACKS TO BE RESERVED FOR OVERFLOW	RESERVED	JFCINTVL INTENTIONAL DELAY IN SECONDS, BETWEEN PASSES

Fields in JFCB — by displacement

DEC	HEX	FIELD
0	0	JFCBDSNM
44	2C	JFCBELNM
44	2C	JFCIPLTX
52	34	JFCBTSDM
53	35	JFCBDSCB
56	38	JFCFCBID
56	38	JFCAMCRO
58	3A	JFCAMSTR
60	3C	JFCBADBF
62	3E	JFCNLREC
64	40	JFCRSV01
66	42	JFCBLTYP
67	43	JFCBOTTR
67	43	JFCBUFOF
68	44	JFCBFLSQ
70	46	JFCBVLSQ
72	48	JFCBMASK
72	48	JFCBOPS1
77	4D	JFCBFLG1
78	4E	JFCBFLG2
79	4F	JFCBOPS2
80	50	JFCBCRDT
83	53	JFCBXPDT
86	56	JFCBIND1
87	57	JFCBIND2
88	58	JFCAMPTR
88	58	JFCBUFRQ
89	59	JFCBFTEK
90	5A	JFCBUFL
92	5C	JFCEROPT
93	5D	JFCPRTSP
94	5E	JFCDEN
95	5F	JFCLIMCT
96	60	JFCTRKBL
98	62	JFCDSORG
98	62	JFCDSRG1
99	63	JFCDSRG2
100	64	JFCRECFM
101	65	JFCOPTCD
102	66	JFCBLKSI
102	66	JFCBAXBF
104	68	JFCAMSYN
104	68	JFCLRECL
106	6A	JFCNCP
107	6B	JFCPCI
108	6C	JFCRKP
110	6E	JFCCYLOF
111	6F	JFCDBUFN
112	70	JFCINTVL
113	71	JFCCPRI
114	72	JFCSOWA
116	74	JFCBNTCS
117	75	JFCBNVOL
118	76	JFCBVOLS
148	94	JFCBEXTL
149	95	JFCBEXAD
152	98	JFCBPQTY
152	98	JFCRUNIT
155	9B	JFCBCTRT
156	9C	JFCBSQTY
156	9C	JFCRQID
159	9F	JFCFLGS1
160	A0	JFCBDQTY
163	A3	JFCBSPNM
166	A6	JFCBABST
168	A8	JFCBSBNM
171	AB	JFCBDRLH
174	AE	JFCBVLCT
175	AF	JFCBSPTN
		PRINTER SEGMENT
108	6C	JFCRKP
110	6E	JFCCYLOF
112	70	JFCINTVL

Alphabetical list of fields in JFCB

FIELD	DEC	HEX
JFCAMCRO	56	38
JFCAMPTR	88	58
JFCAMSTR	58	3A
JFCAMSYN	104	68
JFCBABST	166	A6
JFCBADBF	60	3C
JFCBAXBF	102	66
JFCBCRDT	80	50
JFCBCTRT	155	9B
JFCBDQTY	160	A0
JFCBDRLH	171	AB
JFCBDSCB	53	35
JFCBDSNM	0	0
JFCBELNM	44	2C
JFCBEXAD	149	95
JFCBEXTL	148	94
JFCBFLG1	77	4D
JFCBFLG2	78	4E
JFCBFLSQ	68	44
JFCBFTEK	89	59
JFCBIND1	86	56
JFCBIND2	87	57
JFCBLKSI	102	66
JFCBLTYP	66	42
JFCBMASK	72	48
JFCBNTCS	116	74
JFCBNVOL	117	75
JFCBOPS1	72	48
JFCBOPS2	79	4F
JFCBOTTR	67	43
JFCBPQTY	152	98
JFCBSBNM	168	A8
JFCBSPNM	163	A3
JFCBSPTN	175	AF
JFCBSQTY	156	9C
JFCBTSDM	52	34
JFCBUFL	90	5A
JFCBUFOF	67	43
JFCBUFRQ	88	58
JFCBVLCT	174	AE
JFCBVLSQ	70	46
JFCBVOLS	118	76
JFCBXPDT	83	53
JFCCPRI	113	71
JFCCYLOF	110	6E
JFCDBUFN	111	6F
JFCDEN	94	5E
JFCDSORG	98	62
JFCDSRG1	98	62
JFCDSRG2	99	63
JFCEROPT	92	5C
JFCFCBID	56	38
JFCFLGS	159	9F
JFCINTVL	112	70
JFCIPLTX	44	2C
JFCLIMCT	95	5F
JFCLRECL	104	68
JFCNCP	106	6A
JFCNLREC	62	3E
JFCOPTCD	101	65
JFCPCI	107	6B
JFCPRTSP	93	5D
JFCRECFM	100	64
JFCRKP	108	6C
JFCRQID	156	9C
JFCRSV01	64	40
JFCRUNIT	152	98
JFCSOWA	114	72
JFCTRKBL	96	60

PRINTER SEGMENT

JFCCYLOF	110	6E
JFCINTVL	112	70
JFCRKP	108	6C

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JFCBTSDM	JOB MANAGEMENT/ DATA MANAGEMENT INTERFACE	JFCCAT	'80'	CATALOGED DATA SET
		JFCVSL	'40'	VOLUME SERIAL LIST ALTERED
		JFCSDS	'20'	SYSIN OR SYSOUT DATA SET
		JFCTTR	'10'	USE JFCBOTTR INSTEAD OF DSILSTAR FIELD TO REPOSITION DATA SET IF AUTOMATIC STEP RESTART OCCURS
		JFCNWRIT	'08'	DO NOT WRITE BACK THE JFCB DURING OPEN PROCESSING
		JFCNDSCB	'04'	DO NOT MERGE DSCB OR LABEL FIELDS INTO THIS JFCB
		JFCNDCB	'02'	DO NOT MERGE DCB FIELDS INTO THIS JFCB
		JFCPAT	'01'	THE PATTERNING DSCB IS COMPLETE
JFCBLTYP	LABEL TYPE	JFCBAL	'40'	USASI (AL OR AUL)
		JFCBLTM	'20'	NON-LABELED TAPE CREATED BY DOS MAY HAVE LEADING TAPE MARK. OPEN/CLOSE/EOV AND RESTART ARE TO SPACE OVER TAPE MARK IF IT EXISTS
		JFCBLP	'10'	BYPASS LABEL PROCESSING
		JFCSUL	'0A'	USER LABEL
		JFCNSL	'04'	NONSTANDARD LABEL
		JFCSL	'02'	STANDARD LABEL
		JFCNL	'01'	NO LABEL
JFCBUFOF	BUFFER OFFSET	JFCBFOFL	'80'	USASI/USASCII - BLOCK PREFIX IS 4 BYTES AND CONTAINS BLOCK LENGTH IN UNPACKED DECIMAL
JFCFUNCT	FUNCTION INDICATOR FOR THE 3525	JFCFNCBI	'80'	INTERPRET (PUNCH AND PRINT TWO LINES)
		JFCFNCBR	'40'	READ
		JFCFNCBP	'20'	PUNCH
		JFCFNCBW	'10'	PRINT
		JFCFNCBD	'08'	DATA PROTECTION
		JFCFNCBX	'04'	THIS DATA SET IS TO BE PRINTED
		JFCFNCBT	'02'	TWO-LINE PRINT SUPPORT REQUEST
JFCBOPS1	OPEN ROUTINE INTERNAL SWITCHES	JFCBPTRR	'01'	MODEL DEPENDENT SUPPORT
JFCBFLG1	FLAG BYTE	JFCSTAND	'80'	VOLUME LABEL PROCESSING STANDARD
		JFCSLCRE	'40'	CREATION OF A STANDARD LABEL IS NECESSARY
		JFCSLDES	'20'	DESTRUCTION OF A STANDARD LABEL IS NECESSARY
		JFCDUAL	'10'	DUAL DENSITY CHECK DETECTED
JFCBFLG2	FLAG BYTE OF OPEN SWITCHES	JFCINOP	'80'	TREAT THE INOUT OPTION OF OPEN AS INPUT
		JFCOUTOP	'40'	TREAT THE OUTIN OPTION OF OPEN AS OUTPUT
		JFCDEFER	'20'	DATA SET RELATED TO THE JFCB IS BEING PROCESSED SEQUENTIALLY, AT THE CHECKPOINT, ON A DIFFERENT VOLUME THAN WHEN PROCESSING BEGAN - WHEN RESTART OCCURS, CAUSE DEFERRED VOLUME MOUNTING
		JFCMODNW	'10'	DISPOSITION CHANGED FROM MOD TO NEW - DISPOSITION (IN JFCBIND2) WILL BE RESTORED TO MOD AFTER OPEN
		JFCSDRPS	'08'	SEARCH DIRECT FOR RPS DEVICES
		JFCTRACE	'04'	GTF TRACE IS TO OCCUR DURING OPEN/CLOSE/EOV AND DYNAMIC ALLOCATION PROCESSING OF DCB
		JFCBBUFF	'02'	INDICATOR TO OPEN THAT A NON-ZERO VALUE IN JFCBOTTR IS NOT TO PREVENT THE NORMAL STORING BY OPEN OF A TTR IN JFCBOTTR

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
		JFCRCTLG	'01'	SCHEDULER STEP TERMINATION ROUTINE IS TO RECATALOG THIS DATA SET AND PLACE IN THE CATALOG ENTRY THE DSCB TTR CONTAINED IN JFCBDSCB
JFCBIND1	INDICATOR BYTE 1	JFCRLSE	'C0'	RELEASE EXTERNAL STORAGE
		JFCLOC	'30'	DATA SET HAS BEEN LOCATED
		JFCADDED	'0C'	NEW VOLUME HAS BEEN ADDED TO DATA SET
		JFCGDG	'02'	DATA SET IS A MEMBER OF A GENERATION DATA GROUP
		JFCPDS	'01'	DATA SET IS A MEMBER OF A PARTITIONED DATA SET
JFCBIND2	INDICATOR BYTE 2	JFCNEW	'C0'	NEW DATA SET
		JFCMOD	'80'	MOD DATA SET
		JFCOLD	'40'	OLD DATA SET
		JFCBRWPW	'30'	BITS 2 & 3 ALLOWS READ WITHOUT PASSWORD WITH DATA SET SECURITY
		JFCSECUR	'10'	DATA SET SECURITY
		JFCSHARE	'08'	SHARED DATA SET
		JFCENT	'04'	DATA SET IS GENERATION OF A GDG - PROCESSING OF ENTIRE GDG WAS SPECIFIED - IF AUTOMATIC RESTART AT A CHECKPOINT OCCURS, JFCB WILL BE DELETED
		JFCREQ	'02'	STORAGE VOLUME REQUESTED
		JFCTEMP	'01'	TEMPORARY DATA SET
JFCBFTEK	BUFFERING TECHNIQUE - FOR GAM THIS FIELD IS USED FOR THE NUMBER OF IOBS CONSTRUCTED BY THE OPEN ROUTINE. MAX NUMBER IS 99	JFCSIM	'40'	SIMPLE BUFFERING
		JFCEXC	'10'	EXCHANGE BUFFERING
		JFCDYN	'08'	DYNAMIC BUFFERING
		JFCDWORD	'02'	DOUBLE WORD BOUNDARY
		JFCFWORD	'01'	FULL WORD BOUNDARY - NOT DOUBLE WORD
JFCEROPT	ERROR OPTION SWITCH	JFCACC	'80'	ACCEPT
		JFCSKP	'40'	SKIP
		JFCABN	'20'	ABNORMAL END OF TASK
		JFCTOPT	'10'	ON-LINE TERMINAL TASK (BTAM)
JFCTRTCH	TAPE RECORDING TECHNIQUE FOR 7-TRACK TAPE	JFCEVEN	'23'	EVEN PARITY
		JFCTRAN	'3B'	BCB/EBCDIC TRANSLATION
		JFCCONV	'13'	DATA CONVERSION
		JFCTREV	'2B'	EVEN PARITY AND TRANSLATION
JFCCODE	CONVERSION CODE	JFCNOCON	'80'	NO CONVERSION
		JFCBCD	'40'	IBM BCD
		JFCFRI	'20'	FRIDEN
		JFCBUR	'10'	BURROUGHS
		JFCNCR	'08'	NATIONAL CASH REGISTER
		JFCASCII	'04'	ASCII (8-TRACK)
		JFCTTY	'02'	TELETYPE
JFCSTACK	STACKER SELECTION	JFCBIN	'80'	COLUMN BINARY
		JFCEBCD	'40'	EBCDIC MODE
		JFCMODEO	'20'	OPTICAL MARK READ MODE
		JFCMODER	'10'	READ COLUMN ELIMINATE MODE
		JFCTWO	'02'	STACKER TWO
		JFCONE	'01'	STACKER ONE
JFCPRTSP	NORMAL PRINTER SPACING	JFCSPTHR	'19'	SPACE THREE LINES
		JFCSPTWO	'11'	SPACE TWO LINES
		JFCSPONE	'09'	SPACE ONE LINE
		JFCSPNO	'01'	NO SPACING

Flags and Masks

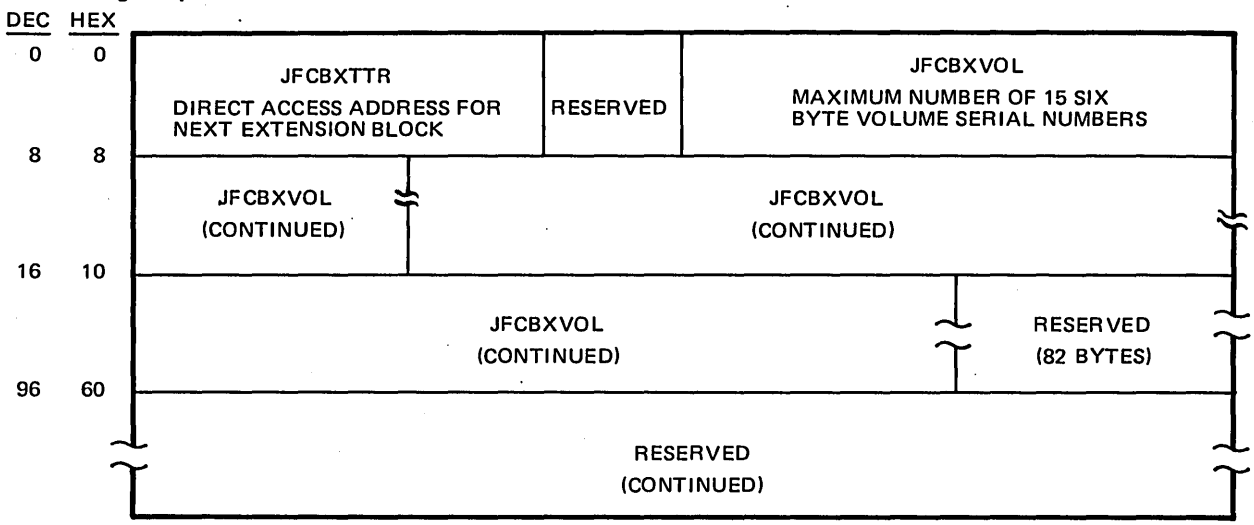
FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JFCDEN	TAPE DENSITY 2400 SERIES MAGNETIC TAPES	JFC200	'03'	7-TRACK 200 BPI
		JFC556	'43'	7-TRACK 556 BPI
		JFC800	'83'	7- and 9-TRACK 800 BPI
		JFC1600	'C3'	9-TRACK 1600 BPI
JFCDSORG	DATA SET ORGANIZATION BEING USED	JFCORGIS	'80'	INDEXED SEQUENTIAL
		JFCORGPS	'40'	PHYSICAL SEQUENTIAL
		JFCORGDA	'20'	DIRECT
		JFCORGPO	'02'	PARTITIONED
		JFCORGU	'01'	UNMOVABLE - THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
JFCDSRG2	BYTE 2 OF JFCDSORG	JFCORGGS	'80'	GRAPHICS
		JFCORGAM	'08'	NEW ACCESS METHOD
JFCRECFM	RECORD FORMAT	JFCND	'C0'	UNDEFINED
		JFCFIX	'80'	FIXED
		JFCVAR	'40'	VARIABLE
		JFCRCFM	'E0'	RECORD FORMAT (USASI/ USASCII)
		JFCVARD	'20'	VARIABLE (FORMAT D FOR USASI/USASCII)
		JFCRFO	'20'	TRACK OVERFLOW
		JFCRFB	'10'	BLOCKED - (MAY NOT OCCUR WITH UNDEFINED)
		JFCRFS	'08'	STANDARD - NO TRUNCATED BLOCKS OR UNFILLED BLOCKS ARE EMBEDDED IN THE DATA SET
		JFCASA	'04'	ASA CONTROL CHARACTER
		JFCMAC	'02'	MACHINE CODE CONTROL CHAR
		JFCNOCC	'00'	NO CONTROL CHARACTER
JFCOPTCD	OPTION CODES			
	QSAM - BSAM - BPAM			
		JFCWVCSP	'80'	WRITE VALIDITY CHECK
		JFCALLOW	'40'	ALLOW A DATA CHECK CAUSED BY AN INVALID CHARACTER (1403 PRINTER WITH UCS)
		JFCPCIBT	'20'	CHAINED SCHEDULING USING THE PROGRAMMED CONTROLLED INTERRUPTION
		JFCBCKPT	'10'	BYPASS EMBEDDED DOS CHECKPOINT RECORDS ON TAPE
		JFCREDUC	'04'	USE REDUCED ERROR RECOVERY PROCEDURE - (MAGNETIC TAPE)
		JFCSRCHD	'04'	SEARCH DIRECT INSTEAD OF SEARCH PREVIOUS - (ROTATIONAL POSITION SENSING DEVICES)
	BISAM - QISAM			
		JFCWVCIS	'80'	WRITE VALIDITY CHECK
		JFCMAST	'20'	MASTER INDEXES
		JFCIND	'10'	INDEPENDENT OVERFLOW AREA
		JFCCYL	'08'	CYLINDER OVERFLOW AREA
		JFCDEL	'02'	DELETE OPTION
		JFCREORG	'01'	REORGANIZATION CRITERIA
	BDAM			
		JFCWVCBD	'80'	WRITE VALIDITY CHECK
		JFCOVER	'40'	TRACK OVERFLOW
		JFCEXT	'20'	EXTENDED SEARCH
		JFCFEED	'10'	FEEDBACK
		JFCACT	'08'	ACTUAL ADDRESSING
		JFCREL	'01'	RELATIVE TRACK ADDRESSING

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
	USASI/USASCII			
		JFCOPTQ	'08'	EBCDIC TO OR FROM USASCII TRANSLATION REQUESTED
	TCAM			
		JFCSDNAM	'80'	SOURCE OR DESTINATION NAME PRECEDES MESSAGE (AFTER CONTROL BYTE)
		JFCWUMSG	'40'	WORK UNIT IS A MESSAGE. DEFAULT WORK UNIT IS A RECORD
		JFCCBWU	'20'	CONTROL BYTE PRECEDES WORD UNIT
JFCPCI	PCI FLAG BYTE	JFCPCIX1	'80'	PCI=(X,) RECEIVE OPERATIONS
		JFCPCIX2	'40'	PCI= (,X) SEND OPERATIONS X INDICATES THAT AFTER THE FIRST BUFFER IS FILLED (ON RECEIVE OPERATIONS) OR EMPTIED (ON SEND OPERATIONS) A PCI OCCURS DURING THE FILLING OR EMPTYING OF THE NEXT BUFFER. THE FIRST BUFFER REMAINS ALLOCATED AND ANOTHER IS ALLOCATED
		JFCPCIA1	'20'	PCI= (A,) RECEIVE OPERATIONS
		JFCPCIA2	'10'	PCI= (,A) SEND OPERATIONS A INDICATES THAT AFTER THE FIRST BUFFER IS FILLED (ON RECEIVE OPERATIONS) OR EMPTIED (ON SEND OPERATIONS) A PCI OCCURS DURING THE FILLING OR EMPTYING OF THE NEXT BUFFER. THE FIRST BUFFER IS DEALLOCATED. A BUFFER IS ALLOCATED IN PLACE OF THE DEALLOCATED BUFFER
		JFCPCIN1	'08'	PCI= (N,) RECEIVE OPERATIONS
		JFCPCIN2	'04'	PCI= (,N) SEND OPERATIONS N INDICATES THAT NO PCIS ARE TAKEN DURING FILLING (ON RECEIVE OPERATIONS) OR EMPTYING (ON SEND OPERATIONS) OF BUFFERS. BUFFERS ARE DEALLOCATED AT THE END OF TRANSMISSION
		JFCPCIR1	'02'	PCI= (R,) RECEIVE OPERATIONS
		JFCPCIR2	'01'	PCI= (,R) SEND OPERATIONS R INDICATES THAT AFTER THE FIRST BUFFER IS FILLED (ON RECEIVE OPERATIONS) OR EMPTIED (ON SEND OPERATIONS) A PCI OCCURS DURING THE FILLING OR EMPTYING OF EACH SUCCEEDING BUFFER. THE COMPLETED BUFFER IS DEALLOCATED. BUT NO NEW BUFFER IS ALLOCATED TO TAKE ITS PLACE.
JFCUCSOP	OPERATION OF THE UCS IMAGE TO BE LOADED	JFCFOLD	'40'	UCS IMAGE TO BE LOADED IN THE FOLD MODE
		JFCVER	'10'	UCS IMAGE TO BE VERIFIED
		JFCFCBAL	'08'	FCB ALIGN
		JFCFCBVR	'04'	FCB VERIFY
JFCCPRI	PRIORITY BETWEEN SEND AND RECEIVE OPERATIONS (QTQM)	JFCSEND	'80'	SEND PRIORITY
		JFCEQUAL	'40'	EQUAL PRIORITY
		JFCRECV	'20'	RECEIVE PRIORITY
JFCBCTRI	SPACE PARAMETERS	JFCBABS	'00'	'ABSTR' REQUEST
		JFCBAVR	'40'	AVERAGE BLOCK LENGTH REQUEST
		JFCBTRK	'80'	'TRK' REQUEST
		JFCBCYL	'C0'	'CYL' REQUEST
		JFCONTIG	'08'	'CONTIG' REQUEST
		JFCMIXG	'04'	'MXIG' REQUEST
		JFCALX	'02'	'ALX' REQUEST
		JFCROUND	'01'	'ROUND' REQUEST
JFCFLGS1	FLAG BYTE (ICB488)	JFCBUAFF	'01'	UNIT AFFINITY SPECIFIED FOR THIS DATA SET

Job File Control Block Extension (JFCBX)

The Job File Control Block Extension (JFCBX) contains the volume serial numbers specified on the DD statements following the fifth DD statement. The JFCBX is created by the interpreter and deleted at job termination time.

JFCBX Storage Map



Fields in JFCBX — by displacement

DEC	HEX	FIELD
0	0	JFCBXTTR
4	4	JFCBXVOL

Alphabetical list of fields in JFCBX

FIELD	DEC	HEX
JFCBXTTR	0	0
JFCBXVOL	4	4

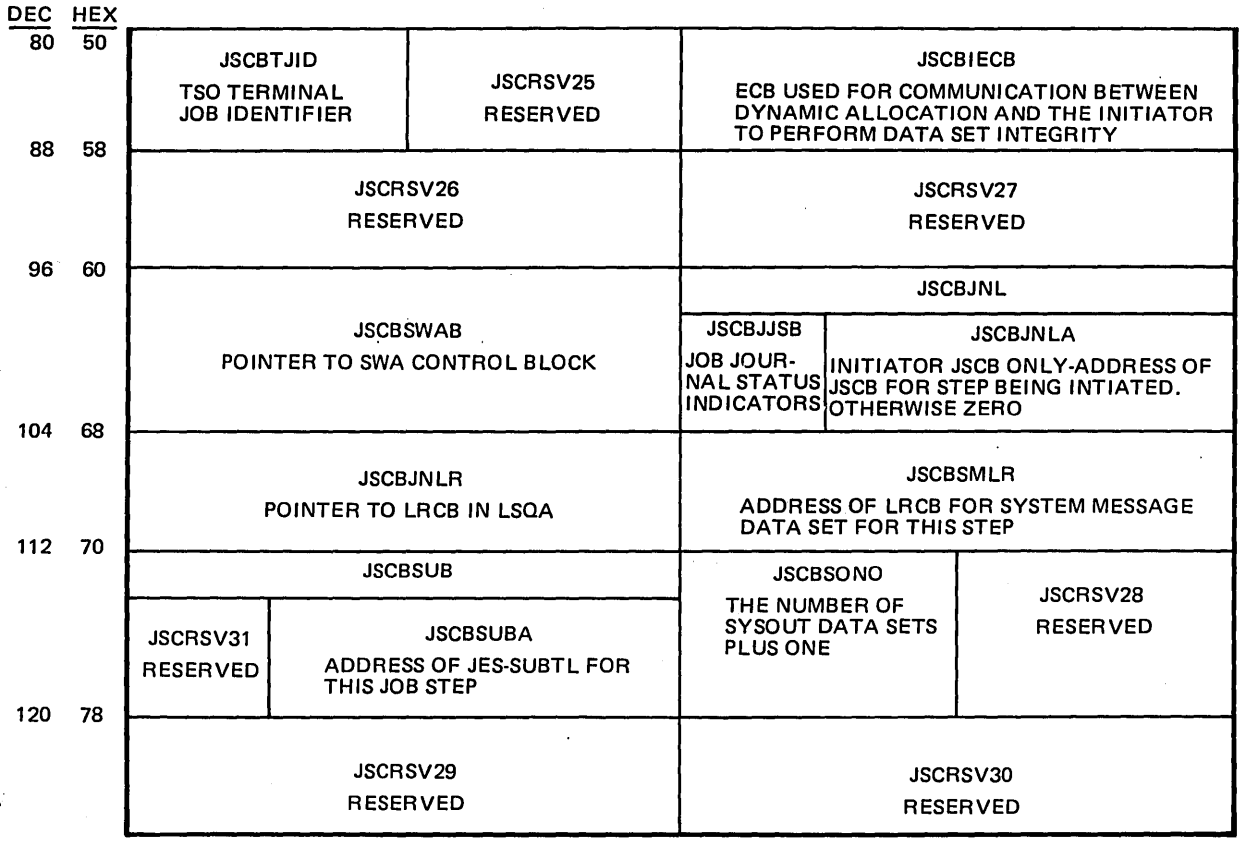
Job Step Control Block (JSCB)

A Job Step Control Block (JSCB) is built for every job whether it is a started job or a dequeued job. The initiator builds the JSCB with th GETJSCB macro after job interpretation and initializes it with job-related information. Data management routines use it as well as JECS and the initiator. The initiator frees the JSCB at job-termination time with the FREEJSCB macro. The JSCB occupies 128 bytes of storage.

JSCB Storage Map

DEC	HEX				
0	0	JSCRSV01 RESERVED		JSCHPCE	
				JSCRSV32 RESERVED	JSCHPCEA ADDRESS OF OPTIONAL JOB ENTRY SUBSYSTEM PROCESSOR CONTROL ELEMENT
8	8	JSCBSHR ADDRESS OF AMBL CHAIN		JSCBTCP ADDRESS OF TIOT CHAINING ELEMENT CHAIN	
16	10	JSCBPCC ADDRESS OF PRIVATE CATALOG CONTROL BLOCK CHAIN		JSCBTCBP ADDRESS OF INITIATOR'S TCB	
24	18	JSCBIJSC ADDRESS OF JSCB OF THE INITIATOR THAT ATTACHED THIS JOB STEP		JSCBDBTB ADDRESS OF THE DEB TABLE	
32	20	JSCBID JOB SERIAL NUMBER		JSCBDCB	
				JSCRSV02 RESERVED	JSCBDCBA ADDRESS OF DCB FOR DATA SET CONTAINING SCHEDULER TABLES FOR THIS JOB
40	28	JSCBSTEP CURRENT STEP NUMBER. CONTAINS 1 FOR FIRST STEP	JSCRSV03 RESERVED	JSCBSECB ECB FOR COMMUNICATION BETWEEN MAIN STORAGE SUPERVISOR AND INITIATOR	
48	30	JSCBOPTS FLAGS	JSCRSV10 RESERVED	JSCBTTTR JOB QUEUE ADDRESS (TTR) OF TIOT EXTENSION (VS2)	JSCBSWT1 STATUS SWITCHES (VS2)
56	38	JSCBQMPI ADDRESS OF THE QMPA FOR THE JOB'S INPUT QUEUE TABLE ENTRIES (VS2)		JSCBQMPO ADDRESS OF THE QMPA FOR THE JOB'S OUTPUT MSGCLASS QUEUE ENTRY (VS2)	
64	40	JSCBWTP WRITE-TO-PROGRAMMER DATA		JSCBCSCB ADDRESS OF CSCB USED FOR PROCESSING COMMANDS RECEIVED FOR JOB	
		JSCBWTFG FLAGS FOR WTP FUNCTION	JSCBWTSP NUMBER OF LAST STEP TO USE WTP	JSCBPMG NUMBER OF WTPs FOR STEP	
72	48	JSCBJCT		JSCBPSCB ADDRESS OF TSO PROTECTED STEP CONTROL BLOCK. (VS2)	
		JSCRSV24 RESERVED	JSCBJCTA TTR OF JOB'S JCT		

JSCB Storage Map (Contd.)



Fields in JSCB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	JSCRSV01	65	41	JSCBWTSP
4	4	JSCHPCE	66	42	JSCBPMG
4	4	JSCRSV32	68	44	JSCBCSCB
5	5	JSCHPCEA	72	48	JSCBJCT
8	8	JSCBSHR	72	48	JSCRSV24
12	C	JSCBTCP	73	49	JSCBJCTA
16	10	JSCBPCC	76	4C	JSCBPSCB
20	14	JSCBTCBP	80	50	JSCBTJID
24	18	JSCBIJSC	82	52	JSCRSV25
28	1C	JSCBDBTB	84	54	JSCBIECB
32	20	JSCBID	88	58	JSCBSV26
36	24	JSCBDCB	92	5C	JSCBSV27
36	24	JSCRSV02	96	60	JSCBSWAB
37	25	JSCBDCBA	100	64	JSCBJNL
40	28	JSCBSTEP	100	64	JSCBJJSB
41	29	JSCRSV03	101	65	JSCBJNLA
44	2C	JSCBSECB	104	68	JSCBJNLR
48	30	JSCBOPTS	108	6C	JSCBSMLR
49	31	JSCRSV10	112	70	JSCBSUB
52	34	JSCBTTTR	112	70	JSCRSV31
55	37	JSCBSWT1	113	71	JSCBSUBA
56	38	JSCBQMPI	116	74	JSCBSONO
60	3C	JSCBQMPO	118	76	JSCRSV28
64	40	JSCBWTP	120	78	JSCRSV29
64	40	JSCBWTFG	124	7C	JSCRSV30

Alphabetical list of fields in JSCB

FIELD	DEC	HEX	FIELD	DEC	HEX
JSCBCSCB	68	44	JSCBSUBA	113	71
JSCBDBTB	28	1C	JSCBSV26	88	58
JSCBDCB	36	24	JSCBSV27	92	5C
JSCBDCBA	37	25	JSCBSWAB	96	60
JSCBID	32	20	JSCBSWT1	55	37
JSCBIECB	84	54	JSCBTCBP	20	14
JSCBIJSC	24	18	JSCBTCP	12	C
JSCBJCT	72	48	JSCBTJID	80	50
JSCBJCTA	73	49	JSCBTTTR	52	34
JSCBJJSB	100	64	JSCBWTFG	64	40
JSCBJNL	100	64	JSCBWTP	64	40
JSCBJNLA	101	65	JSCBWTSP	65	41
JSCBJNLR	104	68	JSCHPCE	4	4
JSCBOPTS	48	30	JSCHPCEA	5	5
JSCBPCC	16	10	JSCRSV01	0	0
JSCBPMG	66	42	JSCRSV02	36	24
JSCBPSCB	76	4C	JSCRSV03	41	29
JSCBQMPI	56	38	JSCRSV10	49	31
JSCBQMPO	60	3C	JSCRSV24	72	48
JSCBSECB	44	2C	JSCRSV25	82	52
JSCBSHR	8	8	JSCRSV28	118	76
JSCBSMLR	108	6C	JSCRSV29	120	78
JSCBSONO	116	74	JSCRSV30	124	7C
JSCBSTEP	40	28	JSCRSV31	112	70
JSCBSUB	112	70	JSCRSV32	4	4

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JSCBOPTS	FLAGS	JSCBLONG	'20'	FAIL REDEFINE BECAUSE OF LONG RUNNING TASK (VS1)
		JSCBAUTH	'01'	STEP REPRESENTED BY THIS JSCB IS AUTHORIZED TO ISSUE THE MODESET MACRO
JSCBSWT1	STATUS SWITCHES (VS2)	JSCBPASS	'80'	WHEN THIS BIT IS SET TO ONE AND A CORRESPONDING BIT IN THE DCB IS SET TO ONE, OPEN WILL BYPASS PASSWORD PROTECTION FOR THE DATA SET BEING OPENED (VS2)
		JSCBPMSG	'01'	A MESSAGE HAS BEEN ISSUED BECAUSE THE DUMP DATA SET WAS NOT SUCCESSFULLY OPENED. PREVENTS USE OF MULTIPLE SMBs FOR MULTIPLE OPEN FAILURES IN THE JOB STEP (VS2)
JSCBWTFG	FLAGS FOR WTP FUNCTION	JSCBIOFG	'80'	PREVIOUS WTP I/O HAD I/O ERROR
JSCBJJSB	JOB JOURNAL STATUS INDICATORS	JSCBJNLN	'80'	NOTHING SHOULD BE WRITTEN IN JOURNAL
		JSCBJNLF	'40'	LIMIT SIZE OF JOURNAL REACHED
		JSCBJNLE	'20'	ERROR IN JOURNAL, DO NOT WRITE
		JSCBJSBJ	'10'	MODULE XEFXB500 TO PROCESS 'AFTER ALLOCATION'
		JSCBJSBI	'08'	JOB HAS NOT ENTERED ALLOCATION FOR THE FIRST TIME
		JSCBJSBA	'04'	JOB HAS ENTERED ALLOCATION
		JSCBJSBX	'02'	JOB HAS COMPLETED ALLOCATION
		JSCBJSBT	'01'	JOB HAS ENTERED TERMINATION

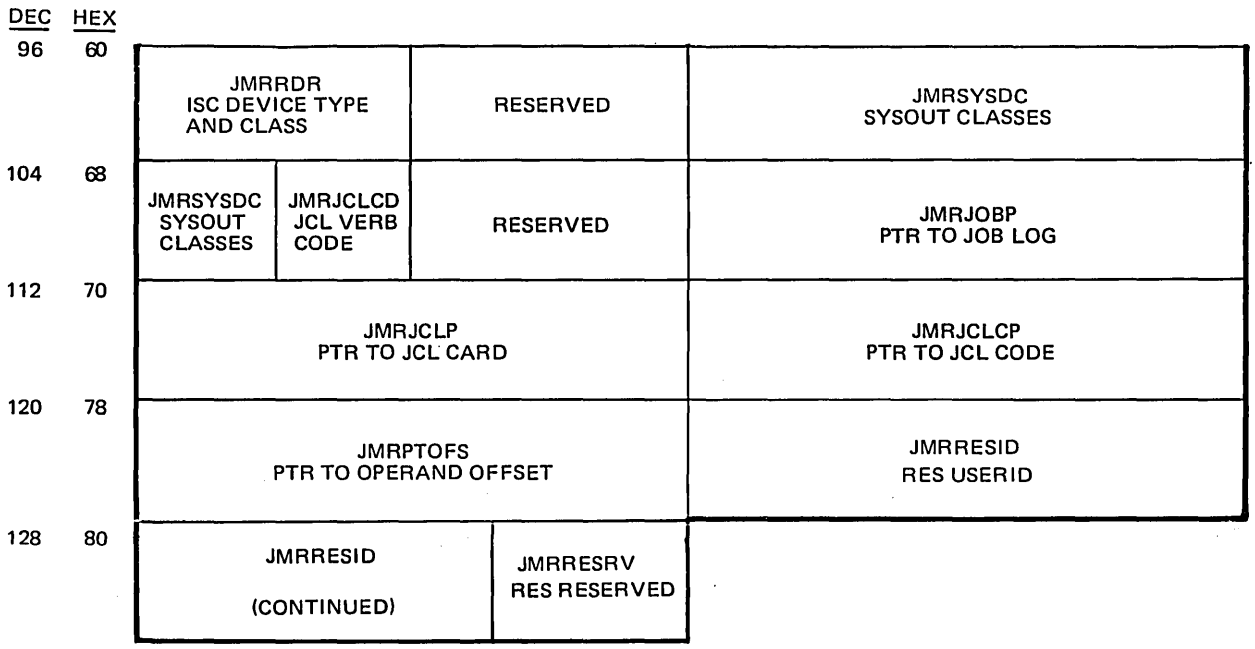
Job Management Record (JMR)

The Job Management Record (JMR) occupies 76 bytes of fixed PQA and contains job information accumulated by IBM-supplied data collection routines. JMR serves as an information source for SMF records and also contains parameters passed to user-exit routines. IEFVMB builds the JMR as the JCL for a job is being interpreted, and IEFOSC05 deletes it.

JMR Storage Map

DEC	HEX				
0	0	JMRTTR TTR OF THIS JMR	JMRID JMR ID	JMRNXT TTR OF NEXT RECORD	RESERVED
8	8	JMRJOB JOB NAME EBCDIC		JMRJOB JOB NAME EBCDIC	
16	10	JMRENTRY JOB ENTRY TIME IN 1/100 SEC.		JMREDATE ENTRY DATE 00YYDDDE	
24	18	JMRCPUID CPU-ID AND MDL FROM SMCA		JMRUSEID USER ID	
32	20	JMRUSEID USER ID		JMRSTEP STEP NUMBER	JMROPT OPTION SWITCHES	RESERVED
40	28	JMRUCOM USER COMMUNICATION INITIALIZED 0 (NOT USED BY THE SYSTEM)		JMRTNRDR REAL TIME IN THE ISC	
48	30	JMRNLRD NUMBER OF LINES READ		JMRJPR JOB PRIORITY	JMRJOBC JOB CLASSES
56	38	JMRTNPR REAL TIME FOR PRINT PROCESS		JMRLPR NUMBER OF LINES PRINTED	
64	40	JMRTNPU REAL TIME FOR PUNCH PROCESS		JMRLNPU NUMBER OF LINES PUNCHED	
72	48	JMRTNT REAL TIME FOR TAPE PROCESS		JMRLWT NUMBER OF LINES WRITTEN TO TAPE	
80	50	JMRUJVP ADDRESS OF IEFUJV/PTR TO SYS1.MAN RECORD		JMRDRSTP ISC STOP TIME AND DATE	
88	58	JMRDRSTP ISC STOP TIME AND DATE		JMRJOBIN JOB SYSIN COUNT	

JMR Storage Map (Cont.)



Fields in JMR — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	JMRTTR	44	2C	JMRTNRDR	84	54	JMRDRSTP
3	3	JMRID	48	30	JMRNLRD	92	5C	JMRJOBIN
4	4	JMRNXT	52	34	JMRJPR	96	60	JMRRDR
8	8	JMRJOB	54	36	JMRJOBC	100	64	JMRSYSDC
16	10	JMRENTRY	56	38	JMRTNPR	105	69	JMRJCLCD
20	14	JMREDATE	60	3C	JMRLPR	108	6C	JMRJOBP
24	18	JMRCPUID	64	40	JMRTNPU	112	70	JMRJCLP
28	1C	JMRUSEID	68	44	JMRLNPU	116	74	JMRJCLCP
34	24	JMRSTEP	72	48	JMRTNT	120	78	JMRPTOFS
35	25	JMROPT	74	4C	JMRLWT	124	7C	JMRRESID
40	28	JMRUCOM	80	50	JMRUJVP	131	83	JMRRESRV

Alphabetical list of fields in JMR

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
JMRCPUID	24	18	JMRJOBP	6C	108	JMRSTEP	34	24
JMRDRSTP	84	54	JMRJPR	52	34	JMRSYSDC	100	64
JMREDATE	20	14	JMRLNPU	68	44	JMRTNPP	56	38
JMRENTRY	16	10	JMRLPRR	60	3C	JMRTNPU	64	40
JMRID	3	3	JMRLWT	74	4C	JMRTNRDR	44	2C
JMRJCLCD	105	69	JMRNLRD	48	30	JMRTNT	72	48
JMRJCLCP	116	74	JMRNXT	4	4	JMRTTR	0	0
JMRJCLP	112	70	* JMROPT	35	25	JMRUCOM	40	28
JMRJOB	8	8	JMRPTOFS	96	60	JMRUSEID	28	1C
JMRJOBC	36	54	JMRRESID	124	7C	* JMRUJVP	80	50
JMRJOBIN	5C	92	JMRRESRV	131	83			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
JMRSTEP	STEP NUMBER	JMRLGEND	8	
		JMRLOGSZ	JMRLGEND-JMRJOB	SIZE OF JOB LOG COPIED TO DSB
JMROPT	OPTION SWITCHES	JMRJOBSW	80	JOB FCTN REQUESTED
		JMRSTPSW	40	STEP FCTN REQUESTED
		JMREXITS	20	USER EXITS REQUESTED
		JMRXONLY	10	EXITS ONLY SPECIFIED
JMRUJVP	ADDRESS OF IEFUJV/PTR TO SYS1.MAN RECORD	JMRSIZE	*-JMRJOB	SIZE OF JMR IN PQA (76 BYTES)

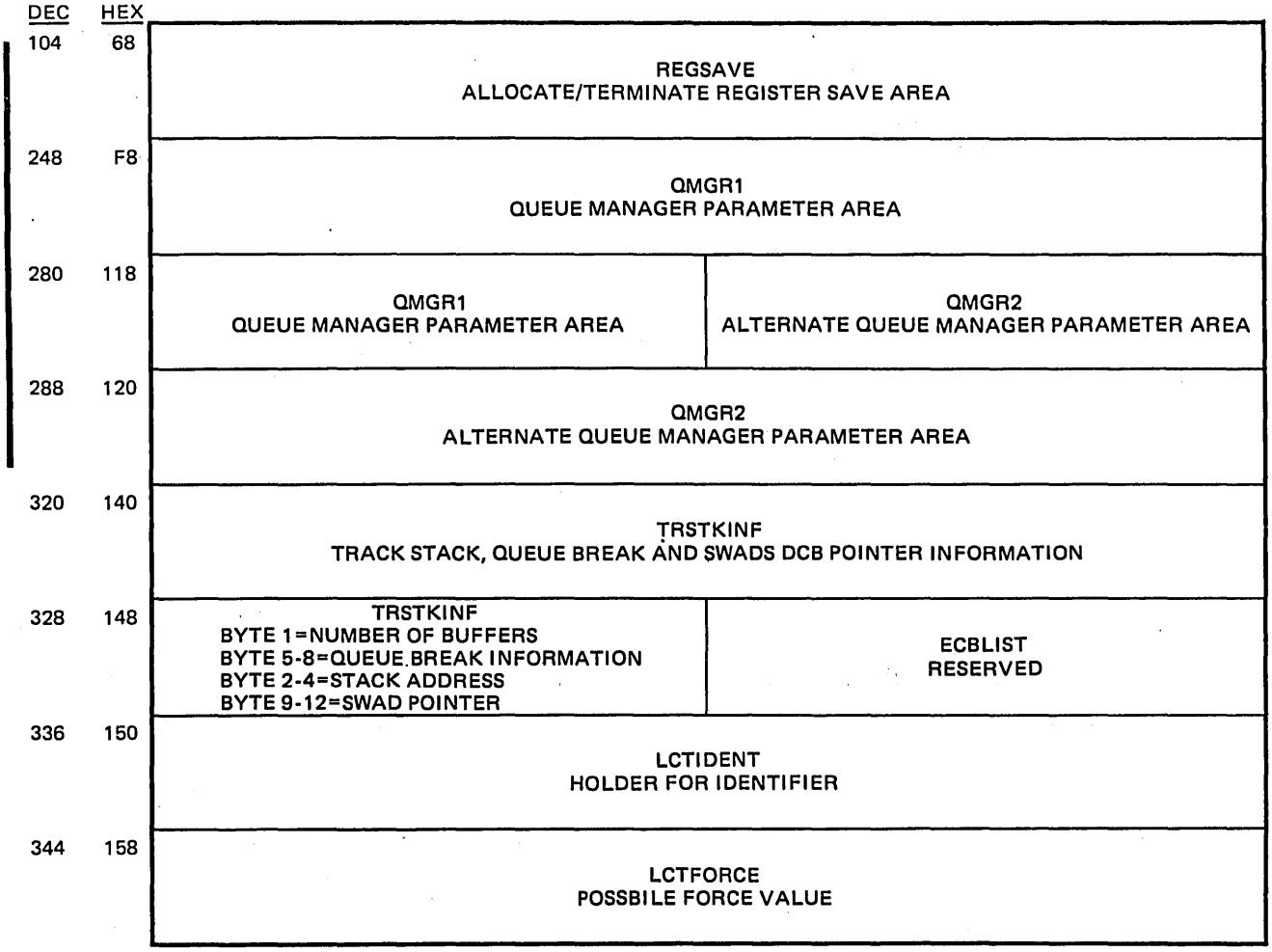
Linkage Control Table (LCT)

The Linkage Control Table (LCT), which occupies 424 bytes, is a communications area used by the initiator routines. IEFSD160 builds the LCT for all started tasks and for each initiator; IEFSD166 deletes it.

LCT Storage Map

DEC	HEX		
0	0	LCTQDRTY BITS 4-7=CONTAIN LPMOD VALUE. THE THREE LOW ORDER BYTES CONTAIN THE CSCB ADDRESS	LCTSRTAD SRT ADDRESS
8	8	LCTTCBAD JOB STEP TCB ADDRESS	LCTQENTY BIT1='NOSEP' BIT3=SPACE WAIT REC. BIT2=DEVICE WAIT RECOVERY
16	10	LCTJCTAD JCT ADDRESS	LCTSCTAD SCT ADDRESS
24	18	LCTSCTDA CURRENT SCT DISK ADDRESS	LCTPSPAR ADDRESS OF RESIDENT BLOCK OF STORAGE FOR MVT ALLOCATE
32	20	LCTERROR ERROR CODE	LCTPARM1 MULTI USE PARM FIELD
40	28	LCTPARM2 MULTI USE PARM FIELD	LCTPARM3 MULTI USE PARM FIELD
48	30	LCTPARM4 MULTI USE PARM FIELD	LCTCMCBA CORE ADDRESS OF CONTROL BYTES FOR CORE MANAGEMENT
56	38	LCTNSPAD NON SET-UP PAD-DING BYTE / LCTJFCBH JFCB HOUSE-KEEPING BYTE / LCTSNUMB CURRENT STEP NUMBER / LCTACTON ACTION CODE	LCTSMBAD STEP TIOT POINTER IN SSS SMB ADDRESS IN PPS AND PSS
64	40	LCTBATMN USED TO GENERATE VOL. SER. NO. WHEN USER SPECIFIES A PASSED DATA SET ON UNLABELED TAPE	LCTSOQMP ADDRESS OF MESSAGE CLASS QUEUE MANAGER PARAMETER AREA
72	48	LCTRTRN RETURN ADDR. TO MASTER SCHEDULER (FOR STOP INITIATOR)	LCTCSCB INITIATOR'S CSCB ADDRESS
80	50	LCTTMWRK TIMER WORK AREA CONSISTING OF 4 FULL WORDS USED AS FOLLOWS	LCTTMWRK STEP TIME
88	58	LCTTMWRK TIME REMAINING	LCTTMWRK TIME USED (STEP)
96	60	LCTJOBLB ADDRESS OF JOBLIB OR STEPLIB DCB	LCTATLST ADDRESS OF ALLOCATE-TERMINATE PARAMETER LISTS

LCT Storage Map (Contd.)



LCT Storage Map (Contd.)

DEC	HEX					
352	160	LCTLIMIT LIMIT VALUE	FRCPRTY FORCE PARITY HOLDER	INITPRTY INITIATOR'S PRIORITY	RESERVED	LCTENTR ADDRESS OF INITIATOR'S EXIT LIST
360	168	LCTCOM COMMUNICATIONS PARAMETER AREA POINTER				LCTJSCB ADDRESS OF JSCB
368	170	QMGR3 QMPA FOR INPUT (SYS1.SYSJOBQE)				
376	178	QMGR3 (9 FULLWORDS) QMPA FOR INPUT (SYS1.SYSJOBQE)				
384	184	QMGR3 QMPA FOR INPUT (SYS1.SYSJOBQE)				
404	194	LCTLRCB LRCB POINTER (ALLOCATION)				LCTDSDAD POINTER TO DATA SET DICTIONARY
412	19C	LCTIOTI TTR POINTER TO INITIATOR TIOT (SYSTEM RESTART)				LCTDERTR TTR POINTER TO DISK ENTRY RECORD ALSO IN-CORE ADDRESS
420	1A4	LOTSTATA FUNCTION OF INI-TIATOR	LOTSTATB FUNCTION OF INI-TIATOR	RESERVED		LCTLBWAP
		LCTSMF ABEND CODE FOR SMF				

Fields in LCT — by displacement

DEC	HEX	FIELD
0	0	LCTQDRTY
4	4	LCTSRTAD
8	8	LCTTCBAD
12	C	LCTQENTY
16	10	LCTJCTAD
24	18	LCTSCTDA
28	1C	LCTPSPAR
32	20	LCTERROR
36	24	LCTPARM1
40	28	LCTPARM2
44	2C	LCTPARM3
48	30	LCTPARM4
52	34	LCTCMCBA
56	38	LCTNSPAD
57	39	LCTJFCBH
58	3A	LCTSNUMB
59	3B	LCTACTON
60	3C	LCTSMBAD
64	40	LCTBATMN
68	44	LCTSOQMP
72	48	LCTRTRN
76	4C	LCTCSCB
80	50	LCTTMWRK
96	60	LCTJOBLB
100	64	LCTATLST
104	68	REGSAVE
248	F8	QMGR1
280	118	QMGR2
320	140	TRSTKINF
332	14C	ECBLIST
336	150	LCTIDENT
344	158	LCTFORCE
352	160	LCTLIMIT
353	161	FRCPRTY
354	162	INITPRTY
356	164	LCTENTR
360	168	LCTCOM
364	16C	LCTJSCB
368	170	QMGR3
404	194	LCTLRCB
408	198	LCTDSDAD
412	19C	LCTDERTR
416	1A0	LCTTIOTI
420	1A4	LOTSTATA
421	1A5	LOTSTATB
424	1A8	LCTLBWAP
428	1AC	LCTSMF

Alphabetical list of fields in LCT

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
ECBLIST	332	14C	LCTJCTAD	16	10	LCTSCTAD	24	18
FRCPRTY	353	161	* LCTJFCBH	57	39	* LCTSMBAD	60	3C
INITPRTY	354	162	LCTJOBLB	96	60	LCTSMF	428	1AC
LCTACTON	59	3B	LCTJSCB	364	16C	LCTSNUMB	58	3A
LCTATLST	100	64	LCTLBWAP	424	1A8	LCTSDQMP	68	44
LCTBATMN	64	40	LCTLIMIT	352	160	LCTSRTAD	4	4
LCTCMCBA	52	34	LCTLRCB	404	194	LCTTCBAD	8	8
LCTCOM	360	168	LCTNSPAD	56	38	LCTTIOTI	416	1A0
LCTCSCB	76	4C	LCTPARM1	36	24	LCTTMWRK	80	50
LCTDERTR	412	19C	LCTPARM2	40	28	* LOTSTATA	420	1A4
LCTDSDAD	408	198	LCTPARM3	44	2C	* LOTSTATB	421	1A5
LCTENTR	356	164	LCTPARM4	48	30	QMGR1	248	F8
LCTERROR	32	20	LCTPSPAR	28	1C	QMGR2	280	118
LCTFORCE	344	158	LCTQDRTY	0	0	QMGR3	368	170
LCTIDENT	336	150	* LCTQENTY	12	C	REGSAVE	104	68
			LCTRTRN	72	48	TRSTKINF	320	140

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
LCTQENTY	BIT1 OF HIGH ORDER BYTE USED IN CONJUNCTION WITH 'NOSEP.' BIT2-DEVICE WAIT RECOVERY	LCTERRM	1	BIT7-JOB TERMINATION STATUS-THE THREE LOW ORDER BYTES CONTAIN THE ADDRESS OF THE LINKER'S REGISTER SAVE AREA
LCTJFCBH	JFCB HOUSEKEEPING BYTE	LCTS2PEM	128	PDQ ENTRY IS INDICATOR
		LCTS2COP	64	CORE OBTAINED FOR PDQ TABLE
		LCTS2FES	32	FIRST ENTRY INTO PDQ PROC S/R
LCTSMBAD	STEP TIOT POINTER INSSS SMB ADDRESS IN PPS AND PSS	LCTREGSV	0	NEW DESIGN
		LCTQMPAM	144	Q-MGR PARAMETER AREA
		LCTCOREA	128	16 BYTE GETMAIN AREA
LCTINTSW	INITIATORS INTERNAL SWITCHES	LCTIHIER	128	RUN IN HIERARCHY ONE
		LCTSD0XX	32	ATTACH IEFSD0XX
		LCTMINRG	16	JOB FLUSH-USE MINPAR
		LCTSTART	8	TASKNAME NOT FOUND ON COMMAND
		LCTSTOP	4	INITIATOR INTERNAL STOP
		LCTABEND	2	EXECUTED PROGRAM ABENDED
LCTOPSW1	INITIATOR'S OPTION BYTE 1	LCTPKEYE	128	DON'T GET PROTECT KEY
		LCTPWFF	64	DON'T PROCESS DEDICATED WORK FILE
		LCTSTMDF	32	DON'T PROCESS STOP/MODIFY
		LCTMINPF	16	GET REGION SIZE SPECIFIED
		LCTCANF	8	ALLOW CANCEL ONLY AT ALLOCATION
		LCTONEJF	4	PROCESS ONLY ONE JOB
		LCTICMDF	2	DON'T PROCESS INIT. COMMANDS
LCTOPSW2	INITIATOR OPTIONS BYTE 2	LCTTIMEF	128	
		LCTCRF	64	DON'T ALLOW CHECK/RESTART
		LCTDSOF	32	DON'T PROCESS DSO
		LCTINTH0	16	INIT IN HIERARCHY ZERO
		LCTINTH1	8	INIT IN HIERARCHY ONE
		LCTENQU	1	DON'T WAIT FOR DATA SETS
LCTOPSW3	INIT. OPTION BYTE THREE	LCTTRSTK	4	INDICATES TRACK STACKING IS IN

Flags and Masks (Contd.)

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
LOTSTATA	THIS STATUS FIELD DEFINES FUNCTIONS TO BE PERFORMED BY THE INITIATOR	LOTSUSPD	80	THE JOB SELECT MODULE WILL CAUSE THE INIT. TO SUSPEND
		LOTSNOWK	40	THE INIT. SUSPENDS TO IEFMF105 IF THERE IS NO WORK TO BE DONE
		LOTBTJOB	20	JOB SELECT MODULE SUSPENDS THE INITIATOR BETWEEN JOBS
		LOTNECBL	10	THE INIT. INITIALIZATION MODULE IS NOT TO CONSTRUCT AN ECB LIST
		LOTJCPIB	08	INIT. GETS JOB INFO. FROM PIB
		LOTNOSDP	04	INIT. IS TO BYPASS STEP DISPATCHING PRIORITY CODE
		LOTNOGCB	02	THIS BIT INDICATES THAT GCB PROCESSING IS TO BE BYPASSED
		LOTCPART	01	THE INIT WILL CHECK PARTITION BOUNDARIES IF A STEP IS BEING RESTARTED FROM A CHECKPOINT.
LOTSTATB	THIS STATUS FIELD DEFINES FUNCTIONS TO BY THE INITIATOR	LOTECBPB	80	THE INIT INITIALIZATION MODULE WILL PUT THE ECB LIST POINTER INTO THE PIB
		LOTNOREG	40	THE INITIATOR IS TO BYPASS REGION DETERMINATION CODE
		LOTNOATC	20	THE INITIATOR WILL BYPASS ATTACH/DETACH CONSIDERATIONS
		LOTWRITE	10	THE STEP SELECT MODULE WILL WRITE THE LOT WITH THE TIOT
		LOTNREAD	08	THE STEP DELETE MODULE WILL NOT READ THE JCT AND SCT.
		LOTSBPOL	04	INIT WILL GET IN SUBPOOL 255.
		LOTNPKEY	02	PROGRAM IS TO RUN IN KEY ZERO.
		LOTMFTIO	01	A COPY OF IEEMFTIO WILL BE USED FOR TERMINATING THIS PROGRAM.

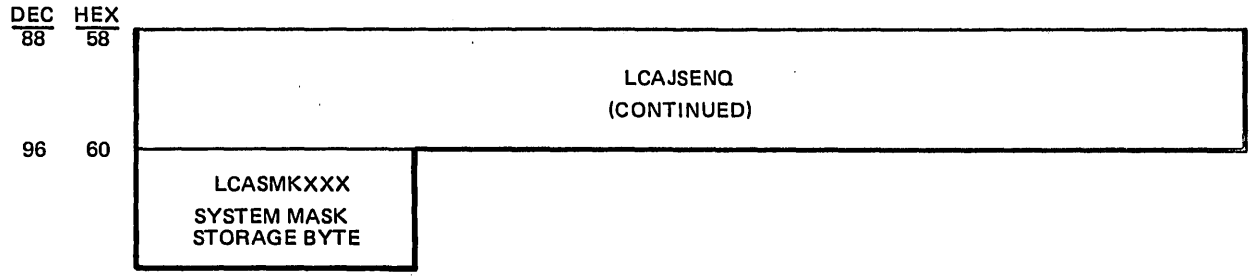
Log Control Area (LCA)

The system log uses the Log Control Area (LCA) as a work area, which is initialized at master scheduler initialization (MSI) time by module IEEVLIN. This work area is used for the life of an IPL and can be found in the virtual=real address space. The LCA occupies 98 bytes.

LCA Storage Map

DEC	HEX					
0	0	LCALGDER TTR OF LOG DER			LCAXDSB TTR OF LOG X DSB	
8	8	LCAYDSB TTR OF LOG Y DSB			LCARPL POINTER TO REQUEST PARAMETER LIST	
16	10	LCALRCB POINTER TO LOG LRCB – USED IN JECS INTERFACE			LCALECB POINTER TO BALOGECB IN MRDA	
24	18	LCAWTLRS OUTLIM VALUE FOR EACH LOG DATA SET			LCAJOBNO JOB NUMBER FOR CURRENT LOG DATA SET	
32	20	LCADER POINTER TO LOG ECB/IOB FOR QMGR			LCAXCTL POINTER TO XCTL NAME	
40	28	DCB POINTER (ALWAYS ZERO)			LCAXNAM XCTL NAME	
48	30	LCAXNAM (CONTINUED)			LCCLASS CLASSNAME FOR WRITELOG COMMAND	LCDEFCLS SYSGEN WRITELOG DEFAULT CLASSNAME LOGCONID MCS CONSOLE ID OR ZEROS
56	38	RESERVED	LCADSTAT LOG DATA SET STATUS FIELD	LCAPOSTT OPERATION ID FIELD	LCAJECSS JECS RESOURCE SWITCH	
64	40	LCAJSECB JECS RESOURCE ECB			LCAJSEC1 TCB SAVE AREA FOR RESOURCE EQUEUE	
72	48	LCALABLQ JECS RESOURCE QUEUE NAME				
80	50	LCALABLR JECS RESOURCE MINOR QUEUE NAME			LCAJSENQ JECS RESOURCE ENQUEUE LIST	

LCA Storage Map (Contd.)



Fields in LCA — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	LCALGDER	53	35	LCDEFCLS
4	4	LCAXDSB	54	36	LOGCONID
8	8	LCAYDSB	58	3A	LCADSTAT
12	C	LCARPL	59	3B	LCAPOSTT
16	10	LCALRCB	60	3C	LCAJECSS
20	14	LCALECB	64	40	LCAJSECB
24	18	LCAWTLRS	68	44	LCAJSEC1
28	1C	LCAJOBNO	72	48	LCALABLQ
32	20	LCADER	80	50	LCALABLR
36	24	LCAXCTL	84	54	LCAJSENQ
44	2C	LCAXNAM	96	60	LCASMKXXX
52	34	LCCLASS			

Alphabetical list of fields in LCA

FIELD	DEC	HEX	FIELD	DEC	HEX
LCADER	32	20	LCAPOSTT	59	3B
LCADSTAT	58	3A	LCARPL	12	C
LCAJECSS	60	3C	LCASMKXXX	96	60
LCAJOBNO	28	1C	LCAWTLRS	24	18
LCAJSECB	64	40	LCAXCTL	36	24
LCAJSEC1	68	44	LCAXDSB	4	4
LCAJSENQ	84	54	LCAXNAM	44	2C
LCALABLQ	72	48	LCAYDSB	8	8
LCALABLR	80	50	LCCLASS	52	34
LCALECB	20	14	LCDEFCLS	53	35
LCALGDER	0	0	LOGCONID	54	36
LCALRCB	16	10			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
LCADSTAT	LOG DATA SET CURRENT STATUS FIELD	LGDSXOPN	'40'	LX DATA SET CURRENTLY OPEN
		LGDSXENQ	'20'	LX DATA SET QUEUED FOR SYSOUT
		LGDSYOPN	'10'	LY DATA SET CURRENTLY OPEN
		LGDSYENQ	'08'	LY DATA SET QUEUED FOR SYSOUT
		LGXYENQD	'04'	BOTH LOG DATA SETS QUEUED FOR SYSOUT
LCAPOSTT	OPERATION ID FIELD	POSTWLOG	'80'	WRITE LOG ISSUED
		POSTCLOZ	'40'	HALT OR WRITE LOG CLOSE ISSUED
		POSTFULL	'20'	OUTPUT LIMIT REACHED
		POSTINIT	'10'	LOG INITIALIZATION INDICATOR
		POSTIOER	'08'	I/O ERROR ON SYS1.SYSPOOL
		POSTNODS	'04'	NO DATA SET AVAILABLE FOR RECORDING

Logical Record Control Block (LRCB)

The spool manager uses the LRCB control table for each requester. The table, built by IEFSMODS in storage obtained by the requester, describes the data set being accessed. The LRCB occupies 120 bytes and is built as the requester does an ODS. It is deleted by IEFSMEND. Storage is freed by the requester.

LRCB Storage Map

DEC	HEX							
0	0	(LRCBDTNM) LRCBREQ REQUESTER ID		(LRCBDTNM) LRCBJOB JOB NUMBER		(LRCBDTNM) LRCBUNIQ UNIQUE NUMBER		
8	8	LRCBCTT TT FIELD		LRCBCR R FIELD	LRCBCL L FIELD	LRCBBUF BUFFER ADDRESS		
16	10	LRCBPTBF LOGICAL RECORD POINTER IN BUFFER				LRCBSTRT START ADDRESS OF DS		
24	18	LRCBSTOP END ADDRESS OF DS				LRCBRLNG TOTAL RECORD LENGTH		
32	20	LRCBWAPT PTR TO WAA REQUEST BLOCK				LRCBFLG1 REQUESTER ID	LRCBFLG2 TYPE OF REQUEST	LRCBFLG3	LRCBFLG4
40	28	LRCBUTPT USER TABLE POINTER				RESERVED	LRCBRCNT RECORD COUNTER	
48	30	RESERVED	LRCBOTLM OUTLIM VALUE			LRCBWKSZ WORK AREA AVAILABLE	LRCBRCSZ LENGTH OF RECORD WRITTEN	
56	38	LRCBWMRB WORK AREA MANAGER REQUEST BLOCK						
64	40	LRCBWMRB WORK AREA MANAGER REQUEST BLOCK						
72	48	LRCBBFRB BUFFER MANAGER REQUEST BLOCK						
80	50	LRCBBFRB BUFFER MANAGER REQUEST BLOCK						
88	58	LRCBMLNG				LRCBMADR		
96	60	LRCBMMOD	LRCBMSPN	LRCBSPAN SPANNING FLAGS	LRCBPTCI POINT REQ. COUNTER	LRCBBFLG BUFFER FLAGS	LRCBIDRQ REQUESTER ID	LRCBCFLG I/O COMPLETION FLAG	LRCBEQDQ ENQ/DER PROCESSING
104	68	LRCBPECB PRE–EMPT I/O ECB				LRCBACWA AREAS TO BE FREED	RESERVED	LRCBCWAL LENGTH OF CWA	
112	70	LRCBDSD POINTER TO CURRENT DSD ENTRY				LRCBCWA POINTER TO CWA		

Field in LRCB – By Displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	LRCBDTNM	36	24	LRCBFLG1	96	60	LRCBMMOD
0	0	LRCBREQ	37	25	LRCBFLG2	97	61	LRCBMSPN
2	2	LRCBJOB	38	26	LRCBFLG3	98	62	LRCBSPAN
6	6	LRCBUNIO	39	27	LRCBFLG4	99	63	LRCBPTCT
8	8	LRCBCTT	40	28	LRCBUTPT	100	64	LRCBBFLG
10	A	LRCBCR	45	2D	LRCBRCNT	101	65	LRCBIDRQ
11	B	LRCBCL	49	31	LRCBOTLM	102	66	LRCBCFLG
12	C	LRCBBUF	52	34	LRCBWKSZ	103	67	LRCBEQDQ
16	10	LRCBPTBF	54	36	LRCBRCSZ	104	68	LRCBPECB
20	14	LRCBSTRT	56	38	LRCBWMRB	108	6C	LRCBACWA
24	18	LRCBSTOP	72	48	LRCBBFRB	110	6E	LRCBCWAL
28	1C	LRCBRLNG	88	58	LRCBMLNG	112	70	LRCBDSD
32	20	LRCBWAPT	92	5C	LRCBMADR	116	74	LRCBCWA

Alphabetical List of Fields in LRCB

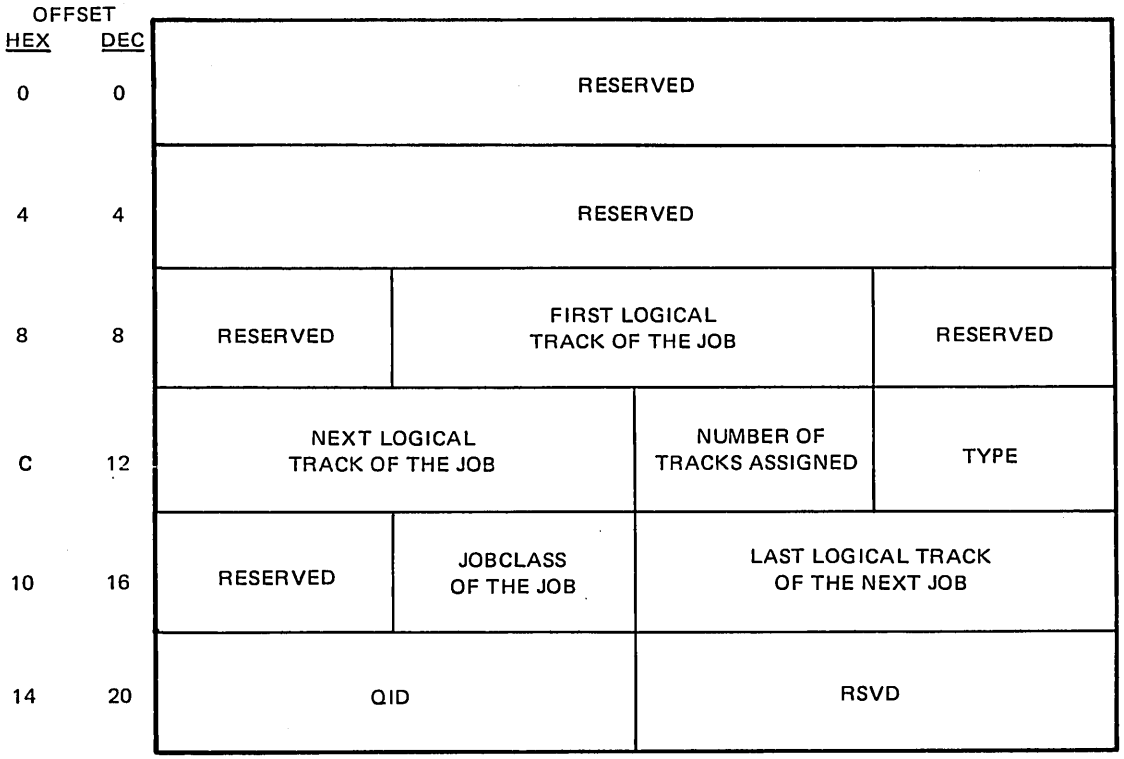
FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
*LRCBACWA	108	6C	LRCBFLG1	36	24	LRCBPTCT		
*LRCBBFLG	100	64	LRCBFLG2	37	25	LRCBRCNT	45	2D
LRCBBFRB	72	48	*LRCBFLG3	38	26	LRCBRCSZ	54	36
LRCBBUF	12	C	*LRCBFLG4	39	27	LRCBREQ	0	0
*LRCBCFLG	102	66	LRCBIDRQ	101	65	LRCBRLNG	28	1C
LRCBCL	11	B	LRCBJOB	2	2	LRCBSPAN	98	62
LRCBCR	10	A	LRCBMADR	92	5C	LRCBSTOP	24	18
LRCBCTT	8	8	LRCBMLNG	88	58	LRCBSTRT	20	14
LRCBCWA	116	74	LRCBMMOD	96	60	LRCBUNIQ	6	6
LRCBCWAL	110	6E	LRCBMSPN	97	61	LRCBUTPT	40	28
LRCBDSD	112	70	LRCBOTLM	49	31	LRCBWAPT	32	20
LRCBDTNM	0	0	LRCBPECB	104	68	LRCBWKSZ	52	34
*LRCBEQDQ	103	67	LRCBPTBF	16	10	LRCBWMRB	56	38

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
LRCBFLG3	FLAG FOR LRCB	LRCBPFLG	80	PUTREC USING BUFFER
LRCBFLG4	FLAG FOR LRCB	LRCBCWAS	80	SUBPOOL FOR CWA ON-241 OFF-254
LRCBBFLG	BUFFER FLAGS	LRCBNPMT	80	DO NOT PRE-EMPT MASK
LRCBCFLG	I/O COMPLETION FLAG	LRCBCTST	80	I/O COMPLETION MASK
LRCBEQDQ	ENQ/DEQ PROCESSING	LRCBQTST	80	END/DEQ TEST
LRCBACWA	WORK AREA TO BE FREED	LRCBFWAM	80	WAM WORKAREA TO BE FREED

Logical Track Header (LTH) Records on SYS1.SYSJOBQE

One Logical Track Header (LTH) record of 176 bytes exists on SYS1.SYSJOBQE for each job on an input/output system queue. The Queue Control Records (QCR) contain pointers to the LTH.



LOGICAL TRACK HEADER (LTH) RECORD FORMAT

Master Scheduler Resident Data Area (BASEA and BASEB)

The scheduler uses the master scheduler resident data area to contain information needed across scheduler components.

It is assembled into the nucleus at sysgen time and is never deleted. It occupies 192 bytes of storage in the nucleus.

BASEA Storage Map

DEC	HEX						
0	0	BACHN HANDLE TO COMMAND SCHEDULING CHAIN			BATRM GROUP QUEUE POINTER		
8	8	BALAD ECB FOR ADDED CHAIN ELEMENT			BAIPL ECB FOR IPL COMMUNICATIONS TASK		
16	10	BAQ POINTER TO JOB QUEUE UCB			BAPRC POINTER TO PROCLIB UCB		
24	18	BACV SYSTEM RESTART INDICATOR AND AUTO= SET COMMAND AND IEEVIPL			BALOG POINTER TO LOG CONTROL TABLE		
32	20	BASFL	BATRST NO. OF TRK. IN INITIA- TOR STACK	BAICTR INTERPRETER COUNTER	BAPKES MASK OF INITIATOR PROTECT KEYS	BAMINPAR MINIMUM INITIATOR PARTITION	
40	28	BAMIPAR2 MINIMUM/MINIMUM PARTITION SIZE	MSLOGST LOG STATUS	BASPBYTE MSTR SCH INIT CPL	BALOCECB LOG ECB		
48	30	BADSO ORIGIN OF DSO CONTROL BLOCK CHAIN			BAMONITR MONITOR FLGS.	BAMONTR2 RESERVED FOR MON- ITOR	BABCMAX MAXIMUM NUMBERS OF BROADCAST MESSAGES
56	38	BADUMPID DUMPID FOR STAE IN MASTER	BASP5 SPARE		BAPNLO LOW BOUNDARY PARTITION SPACE		
64	40	BASP7 SPARE			BASP8 SPARE		
72	48	BASP9 SPARE			BASP10 SPARE		
80	50	BASP11 SPARE			BASP12 SPARE		
88	58	RESERVED FOR BASEB					

BASEA Storage Map (Contd.)

DEC	HEX								
136	88	MSNTAL INITIALI-ZATION BYTE	MSSB SSS SYS-TEM EX-CLUSIVE BYTE 2	MSPFG PENDING FLAGS	MSECBFL ECB FLAGS	MSSSA SSS RESIDENT SWITCHES	MSFHF FETCH FLAGS	MSVRB COMMAND VERB	
144	90	RESERVED				RESERVED		MSPASS VARIABLE COMMUNI-CATION FIELD	
152	98	RESERVED				RESERVED		MSERM MESSAGE GENERA-TION CONTROL	
160	A0	MSPBP FP POINTER. POINTS TO CHAR. BEFORE LIST.				MSECB MASTER ECB			
168	A8	MSSJQ ADDR OF ECB IN SJQ ENTRY OF JOB USING CONSOLE				MSBOBECB ECB FOR ALLOCATION INTERNAL USE. UCB POINTERS.			
176	B0	MSUCBPR PRIMARY UCB				MSUCBAL ALTERNATE UCB			
184	B8	MSABL PSEUDO-DISABLE SWITCH				MSSPARE NEXT TO HIGH PRIORITY PROB PGM TCB PGM TCB			
192	C0	MSTPTCB (MSSPARE) HIGHEST PRIORITY PROBLEM PROG TCB							

BASEB OS/VS1 II AREA

DEC	HEX				
88	58	BAMTTRCB MOUNT TRCB CHAIN POINTER	BASENFE SPARE	BADEFID ID OF CON-SOLE ISSU-ING COM.	BADEFSP DEFINE IN TRAN. INT-FACE
96	60	BALCSBND LCS LOW BOUND	BAFTBBX SUBPOOL 255 BOUNDARY BOX		
104	68	BAFTLO LOW BOUNDARY POINTER	BAFTHI HIGH BOUNDARY POINTER		
112	70	BASRPND SYSTEM ASSIGNED TRCB POINTER	BASFL2 OPTION 2 SWITCHES	BACSCB TRANSIENT READER CSCB PTR	

BASEA Storage Map (Contd.)

DEC	HEX		
120	78	BAIWATTR TRANSIENT READER TTR SAVE FOR ROLLOUT	BADEFINE DEFINE CONTROL INFORMATION
128	80	BASCSZ RESERVED	BATECBP MFT PT TO ECB CHAIN OF READERS WAITING FOR SPACE

Fields in BASEA – by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	BACHN	56	38	BADUMPID	124	7C	BADEFINE
4	4	BATRM	57	39	BASP5	128	80	BASCSZ
8	8	BALAD	60	3C	BAPNLO	132	84	BATECBP
12	C	BAIPL	64	40	BASP7	136	88	MSNTAL
16	10	BAQ	68	44	BASP8	137	89	MSSSB
20	14	BAPRC	72	48	BASP9	138	8A	MSPFG
24	18	BACV	76	4C	BASP10	139	8B	MSECBFL
28	1C	BALOG	80	50	BASP11	140	8C	MSSSA
32	20	BASFL	84	54	BASP12	141	8D	MSFHF
33	21	BATRST	88	58	BAMTTRCB	142	8F	MSVRB
34	22	BAICTR	92	5C	BASENFE	150	96	MSPASS
36	24	BAPKES	94	5E	BADEFID	158	9E	MSERM
38	26	BAMINPAR	95	5F	BADEFSP	160	A0	MSPBP
40	28	BAMIPAR2	96	60	BALCSBND	164	A4	MSFCB
42	2A	MSLOGST	100	64	BAFTBBX	168	A8	MSSJQ
43	2B	BASPBYTE	104	68	BAFTLO	172	AC	MSBOBECB
44	2C	BALOCECB	108	6C	BAFTHI	176	B0	MSUCBPR
48	30	BADSO	112	70	BASRPND	180	B4	MSUCBAL
52	34	BAMONITR	116	74	BASFL2	184	B8	MSABL
53	35	BAMONTR2	117	75	BACSCB	188	BC	MSSPARE
54	36	BABCMAX	120	78	BAIWATTR	192	C0	MSTPTCB

Alphabetical list of fields in BASEA

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
BABCMAX	54	36	* BAMONITR	34.	52	BATRST	33	21
BACHN	0	0	BAMONTR2	53	35	MSABL	184	B8
BACSCB	117	75	BAMTTRCB	88	58	MSBOBECB	172	AAC
BACV	24	18	BAPKES	36	24	MSECB	164	A4
BADEFID	94	5E	BAPNLO	60	3C	* MSECBFL	139	8B
BADEFINE	124	7C	BAPRC	20	14	MSERM	158	9E
BADEFSP	95	5F	BASCSZ	128	80	* MSFHF	8D	141
BADSO	48	30	BASENFE	92	5C	* MSLOGST	42	2A
BADUMPID	56	38	* BASFL	32	20	* MSNTAL	136	88
BAFTBBX	100	64	BASFL2	116	74	MSPARE	188	BC
BAFTLO	104	68	* BASPBYTE	43	2B	MSPASS	96	150
BAFTHI	108	6C	BASP5	57	39	MSPBP	A0	160
BAICTR	34	22	BASP7	64	40	* MSPFG	8A	138
BAIPL	12	C	BASP8	68	44	MSSJQ	168	A8
BAIWATTR	120	78	BASP9	72	48	MSSSA	140	8C
BALAD	8	8	BASP10	76	4C	* MSSSB	137	89
BALCSBND	96	60	BASP11	80	50	MSTPTCB	192	C0
BALOG	28	1C	BASP12	84	54	MSUCBAL	180	B4
BALOCECB	44	2C	BASRPND	112	70	MSUCBPR	±176	B0
BAMINPAR	38	26	BATECBP	132	84	MSVRB	142	8E
BAMIPAR2	40	28	BATRM	4	4			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
BASFL	JOB FLAGS	BAIN	128	IPL FLAG
		BAJN	64	JOBNAMES FLAG
		BAINTSET	32	INDICATES INTERNAL SET FOR TOD

Flags and Masks (Contd.)

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
		BAVU	16	VARY/UNLOAD SUMMARY FLAG
		BAHR	8	Q HOLD-RELEASE FLAG
		BADSP	4	DISPLAY A PROCEDURE FLAG
		BANOSET	2	TOD CLOCK SHOULD NOT BE SET
		BADSET	1	INVALID SET COMMAND INDICATOR FOR TOD CLOCK
MSLOGST	LOG STATUS	MSLOGENQ	128	LOG DATA SET SCHEDULED TO BE QUEUED TO SYSOUT WRITER
		MSLOGTHD	64	LOG NOT SUPPORTED BIT
		MSLOGCOM	32	SIGNAL FOR COM TASK TO STOP ISSUING WTLS
		MSGLODAR	16	CONTROL BIT TO IDENTIFY RE-ENTRY OF IEELOGWR FROM DAR (DUE TO 80A ABEND)
BASPBYTE	MSTR SCH INIT CPL	BAMSCPLT	80	MASTER SCHED INITIALIZATION IS COMPLETE BIT-SET ON ENTRY TO IEEVWAIT
BAMONITR	MONITOR FLAGS (TERMINAL-TJB CHAIN, CONSOLES-UCME CHAIN)	BAMJIN	128	JOBNAMES FLAG FOR TERMINALS
		BAMDSN	64	DSNAME FLAG FOR TERMINALS
		BAMSPACE	32	SPACE FLAG FOR TERMINALS
		BAMSTAT	16	STATUS FLAG FOR TERMINALS
		BAMSESST	8	SESSIONS FLAGS FOR TERMINALS
		BAMSFSSC	4	SESSIONS FLAGS FOR CONSOLES
MSNTAL	INITIALIZATION BYTE	MSNIP	128	IPL SWITCH
			64	SYSOUT IPL
			32	SYSOUT JOB START
		MSCURE34	4	34 SECURITY BIT
		MSQNIP	2	Q INITIALIZED
		MSPNIP	1	PROCEDURE CATALOG INITIALIZED
MSSSB	SSS SYSTEM EXCLUSIVE BYTE 2	MSCONFLG	128	CONSOLE FLAG
		MSCANFLG	64	CANCEL FLAG FOR ABEND
		MSROLFLG	32	ROLL-OUT FLAG
		MSSO	16	SPINOFF FLAG (CANCEL)
		MSBTN	MSSSB	TIME NOTE BIT IS BIT 6
		MSTN	2	VALUE TO TURN ON TIME NOTE
		MSSSDSN	8	DISPLAY DATASET NAME
		MSSSPACE	4	DISPLAY SPACE
MSPFG	C PENDING FLAGS	MSDATE	128	IPL DATE
		MSPNB	64	PARTITION BUSY
		MSCMC	32	COMMAND MOVE COMPLETED
		MSICR	16	INTERPRETER COMMAND RETURN
		MSSYN	8	SYSTEM INPUT CONTROL PURGE REQUEST
		MSSYT	4	SYSTEM OUTPUT CONTROL PURGE REQUEST
		MSBSP	2	BLANK START PENDING (REQ=1, START BLANK=0)
		MSCCS	1	CONSOLE COMMAND SUPPRESSED
MSECBFL	ECB FLAGS	MSEXT	128	EXTERNAL INTERRUPT
		MSWTO	64	WRITE TO OPERATOR
		MSWTL	32	WRITE TO LOG
		MSATTN	16	CONSOLE ATTENTION
		MYMSYSIN	8	SYSTEM INPUT
		MSYSOUT	4	SYSTEM OUTPUT
		MSMCR	2	MASTER COMMAND ROUTINE
		MSSUM	1	SUMMARY BIT, VARY UCB SCAN REQUIRED
MSTUS	STATUS FLAGS	MSINLSW	128	MASTER INITIALIZATION SWITCH
		MSSSSIPL	128	IPL
		MSWRPEN	64	WTO PENDING
		MSNUPSW	32	CONSOLE USAGE PRINCIPLE OR ALTERNATE

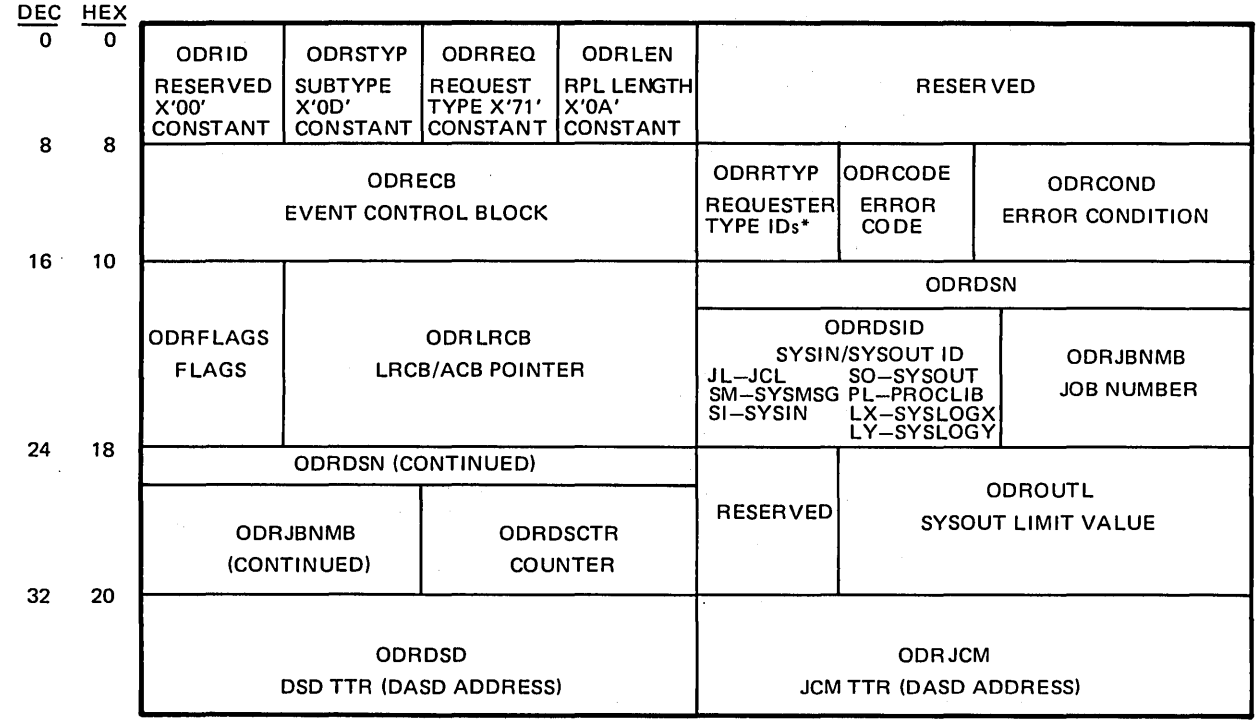
Flags and Masks (Contd.)

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
		MSWRLOG	16	LOG PURGE REQUEST
		MSREOF	8	READER END OF FILE
		MSSRDR	8	START READER
		MSNRP	4	NEW READER PENDING
		MSNWP	2	NEW WRITER PENDING
		MSYOUT	2	NEW WRITER PENDING
		MSJNF	1	JOB NOTIFICATION FLAG (1=YES)
MSFHF	FETCH FLAGS	MSNMF	128	NAMED FETCH
		MSCSD	64	CURRENT COMMAND EXECUTION SEQUENCE DEFER.
		MSTTT	32	TCB TREE TRACE FETCH (LOCATE)
		MSFAX	16	AUXILIARY FETCH GIVEN
		MSREPLYB	8	REPLY BIT TO REQUEST ATTENTION
		MSPSDT	4	PSEUDO-SYSOUT FLAG
		MSDISPST	2	STATUS NOTIFICATION FLAG
		MSQHR	1	Q HOLD-RELEASE
BADEFSP	DEFINE IN TRAN-SIENT INTERFACE	BASPQSCE	80	PARTITIONS QUIESCED FOR DEFINE
		BASPDEFN	20	DEFINITIONS IN PROGRESS
BASFL2	OPTION 2 SWITCHES	BACTIV	128	TRANSIENT READER ACTIVE
		BARDIN	64	TRANSIENT READER IN CORE (OFF = NOT IN CORE)
		BAPEND	32	TRANSIENT START READER PENDING = ON
		BAOPT2	16	OPTION 2 ENVIRONMENT
		BASRR	8	SYSTEM ASSIGNED READER IS RUNNING
		BALCS	04	LCS IN SYSTEM
BADEFINE	DEFINE CONTROL INFO.	BAON	80	ON = DEFINE OPERATION. OFF = IPL TIME
		BALIST	40	LIST REQUESTED
		BASTP	20	ON = ADJACENT PARTITION CHECK
		BADYNDEF	10	DYNAMIC DEFINITION ALLOWABLE
		BACHANGE	08	IPL TIME REQUEST TO CHANGE PARTITION
		BASPWT	04	SMALL PARTITION CANNOT TERMINATE BECAUSE OF DEFINE OPERATION.
		BADFCMD	02	A DEFINE COMMAND HAS BEEN ISSUED
		BAPFK	01	PROTECTION INDICATOR

Open Data Set Request Parameter List (ODRPL)

The user passes the Open Data Set Request Parameter List (ODRPL) to JECS for the open data set function (IEFSMODS). The user allocates, initializes, and frees the storage required by the ODRPL. ODRPL requires 40 bytes of storage.

ODRPL Storage Map



* The following are valid requester type IDs:

X'01' - Interpreter
X'02' - Allocation
X'03' - WTP
X'04' - Termination
X'05' - Transient
X'06' - Restart reader in problem program partition
X'07' - DSO
X'08' - Data Management
X'09' - Initiator
X'0A' - Checkpoint/Restart
X'0B' - System restart
X'0C' - SWADS
X'10' - JES writer on behalf of user writer
X'40' - Special data set
X'90' - JES writer
X'00' - JES reader

Fields in ODRPL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	ODRID	17	11	ODRLRCB
1	1	ODRSTYP	20	14	ODRDSN
2	2	ODRREQ	20	14	ODRDSID
3	3	ODRLEN	22	16	ODRJBNMB
8	8	ODRECB	26	1A	ODRDSCTR
12	C	ODRRTYP	29	1D	ODROUTL
13	D	ODRCODE	32	20	ODRDSD
14	E	ODRCOND	36	24	ODRJCM
16	10	ODRFLAGS			

Alphabetical list of fields in ODRPL

FIELD	DEC	HEX	FIELD	DEC	HEX
ODRCODE	13	D	ODRJBNMB	22	16
ODRCOND	14	E	ODRJCM	36	24
ODRDSCTR	26	1A	ODRLEN	3	3
ODRDSD	32	20	ODRLRCB	17	11
ODRDSN	20	14	ODROUTL	29	1D
ODRSID	20	14	ODRREQ	2	2
ODRECB	8	8	ODRRTYP	12	C
ODRFLAGS	16	10	ODRSTYP	1	1
ODRID	0	0			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
ODRFLAG	FLAGS	ODRSOUT	'40'	SYSOUT DATA SET
		ODROUT	'20'	OUTPUT DATA SET
		ODRINPT	'10'	INPUT DATA SET
		ODRSQA	'08'	GET DSD FROM SQA
		ODRWNEW	'04'	WRITE TO NEW DATA SET
		ODRWOLD	'02'	WRITE TO OLD DATA SET
		ODRREAD	'01'	READ FROM OLD DATA SET

Partition Information Block (PIB)

The Partition Information Block (PIB) describes what job classes can run in a partition and provides information about the status of an initiator in that partition. It also provides information about the task operating in that partition. The PIB, built at SYSGEN time—one per each user partition, is never deleted and occupies eighty bytes of storage.

PIB Storage Map

DEC	HEX				
0	0	SD33ITTR THIS FIELD CONTAINS THE TTR OF THE SUSPENDED INITIATOR'S TIOT		SD33LOTP CONTAINS POINTER TO LCT USED FOR PROBLEM PROGRAMS	
8	8	SD33WECB ECB IS POSTED BY DEFINE, START/STOP COMMAND BY 510, BEFORE AN XCTL		SD33STAT	SD33ECBL THIS FIELD IS FOR STATUS INFORMATION
16	10	SD33BBTS	SDD33TRCN THIS FIELD CONTAINS STATUS BITS	SD33CSCB THIS FIELD CONTAINS THE ADDRESS OF THE CSCB FOR THE CURRENT TASK IN THIS PARTITION	
24	18	SD33GRP THIS FIELD CONTAINS FROM ONE TO FIFTEEN JOB CLASSES BEGINNING AT THE LOW ORDER BYTE. THE 1ST BYTE CONTAINS A PROTECT KEY SET BY THE DEFINE COMMAND AT IPL TIME. THE OTHER LEADING BYTES CONTAIN BINARY ZEROS OR CLASS CODES.			
32	20	SD33GRP			
40	28	SD33RTTR THIS FIELD CONTAINS THE ADDRESS OF THE INITIATOR'S CSCB		SD33DSO THIS FIELD CONTAINS THE ADDRESS OF THE DSOCB CHAIN	
48	30	SD33INTQ INTERNAL QUEUE POINTER		SD33JTQE POINTER TO THE TQE FOR THE JOB/STEP TIMING FUNCTION. HIGH ORDER BYTE IS USED FOR FLAGS	
56	38	SD33JPAQ BYTE0—NO. OF ATTACHES IN JOB STEP. BYTES 1-3 PTR. TO LOADED RE-ENTRANT MODULES SHARED BY JOB		SD33IECB THIS FIELD CONTAINS THE ADDRESS OF THE INITIATOR'S ECB LIST	
64	40	SD33JBNM JOB NAME OF JOB BEING SCHEDULED			
72	48	SD33IJBN JOB NAME OF THE INITIATOR ACTIVE IN THIS PARTITION			

Fields in PIB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SD33ITTR	16	10	SD33TRCN	52	34	SD33JTQE
4	4	SD33LOTP	20	14	SD33CSCB	56	38	SD33JPAQ
8	8	SD33WECB	24	18	SD33GRP	60	3C	SD33IECB
12	C	SD33STAT	40	28	SD33RTTR	64	40	SD33JBNM
12	C	SD33ECBL	44	2C	SD33DSO	72	48	SD33IJBN
16	10	SD33BBTS	48	30	SD33INTQ			

Alphabetical list of fields in PIB

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
* SD33BBTS	16	10	SD33IJBN	72	48	SD33JPAQ	56	38
SD33CSCB	20	14	SD33INTQ	48	30	SD33LOTP	4	4
SD33DSO	44	2C	SD33ITTR	0	0	SD33RTTR	40	28
SD33GRP	24	18	SD33JBNM	64	40	* SD33STAT	12	C
SD33IECB	60	3C	SDTQE	52	34	SD33WECB	8	8

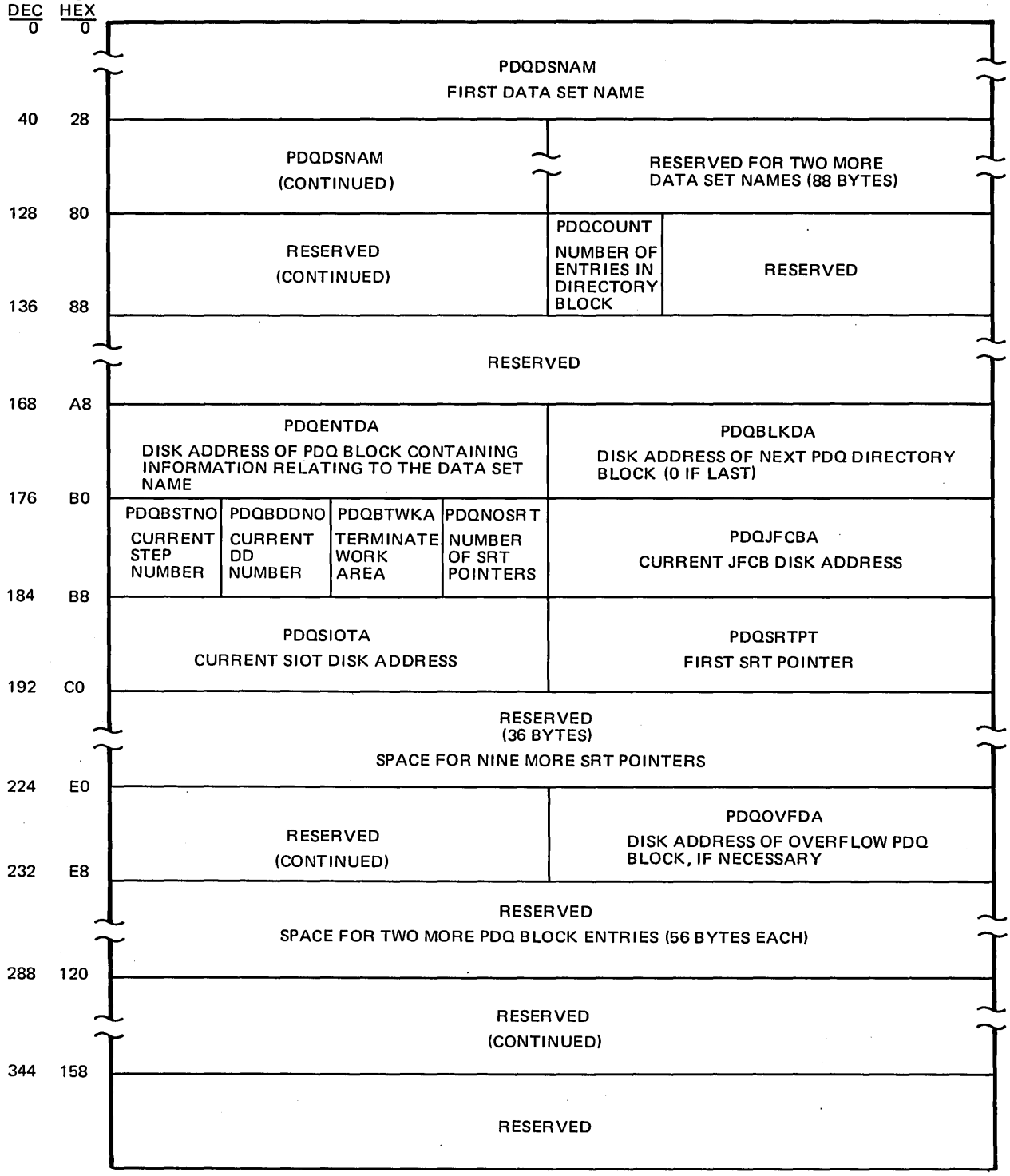
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SD33STAT	THIS FIELD IS FOR STATUS INFORMATION	SD33INIT	80	BIT0-INITIATOR HAS BEEN STARTED IN THIS PARTITION
		SD33GENS	40	BIT1-ON MEANS A GENERALIZED START JOB HAS BEEN STARTED IN THIS PARTITION
		SD33SUSP	20	BIT2-ON WHEN INITIATOR SUSPENDED
		SD33RSTR	10	BIT3-ON WHEN INITIATOR RESTORED BIT4-RESERVED
		SD33STRT	08	BIT0-MEANS START OUTSTANDING FOR INITIATION. OFF MEANS STOP OUTSTANDING
		SD33DFIN	04	BIT5-PARTITION BEING REDEFINED
		SD33WROP	02	BIT6-ON MEANS INIT. HAS BEEN SUSPENDED AT LEAST ONCE
		SD33PPGM	01	BIT7-PROBLEM PROGRAM IS RUNNING
SD33BBTS	THIS FIELD CONTAINS STATUS BITS	SD33NTSK	8	BITS0-3= RESERVED BIT4-ON-UNENDING TASK IN PARTITION BIT5,6=RESERVED
		SD33RTAM	01	BIT7-ON DURING RTAM START-OFF AFTER SUCCESSFUL START
SD33JTQE	POINTER TO THE TQE FOR THE JOB/STEP TIMING FUNCTION. THE HIGH-ORDER BYTE IS USED FOR FLAGS.	SD33TENQ	80	BIT0-INDICATES WHEN THE JOB/STEP TQE HAS BEEN ENQUEUED ON THE TIMER QUEUE.
		SD33TIME	40	BIT1-TURNED ON WHEN STIMER IS ISSUED FOR A P/P STEP.

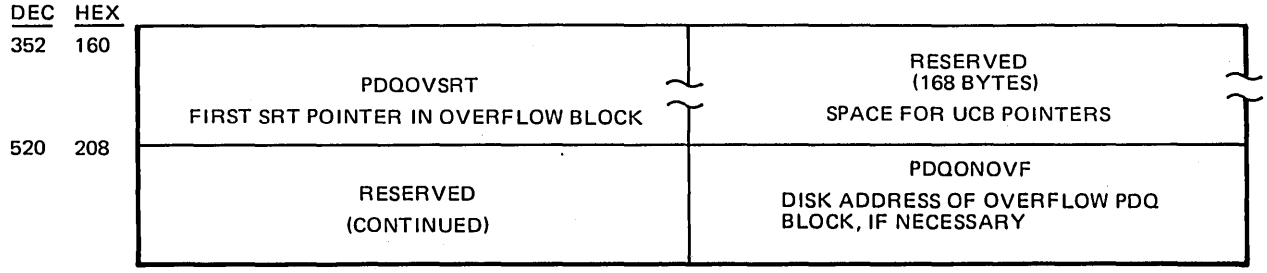
Passed Data Set Queue (PDQ)

The Passed Data Set Queue (PDQ) is created by allocation and contains the information pertaining to passed data sets. The PDQ requires 528 bytes of storage and is deleted at job-termination time.

PDQ Storage Map



PDQ Storage Map (Contd.)



Fields in PDQ — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	PDQDSNAM	179	B3	PDQNOSRT
132	84	PDQCOUNT	180	B4	PDQJFCBA
168	A8	PDQENTDA	184	B8	PDQSIOTA
172	AC	PDQBLKDA	188	BC	PDQSRTPT
176	B0	PDQBSTNO	228	E4	PDQOVFDA
177	B1	PDQBDDNO	352	160	PDQOVSRT
178	B2	PDQBTWKA	524	20C	PDQONOVF

Alphabetical list of fields in PDQ

FIELD	DEC	HEX	FIELD	DEC	HEX
PDQBDDNO	177	B1	PDQJFCBA	180	B4
PDQBLKDA	172	AC	PDQNOSRT	179	B3
PDQBSTNO	176	B0	PDQONOVF	524	20C
PDQBTWKA	178	B2	PDQOVFDA	228	E4
PDQCOUNT	132	84	PDQOVSRT	352	160
PDQDSNAM	0	0	PDQSIOTA	184	B8
PDQENTDA	168	A8	PDQSRTPT	188	BC

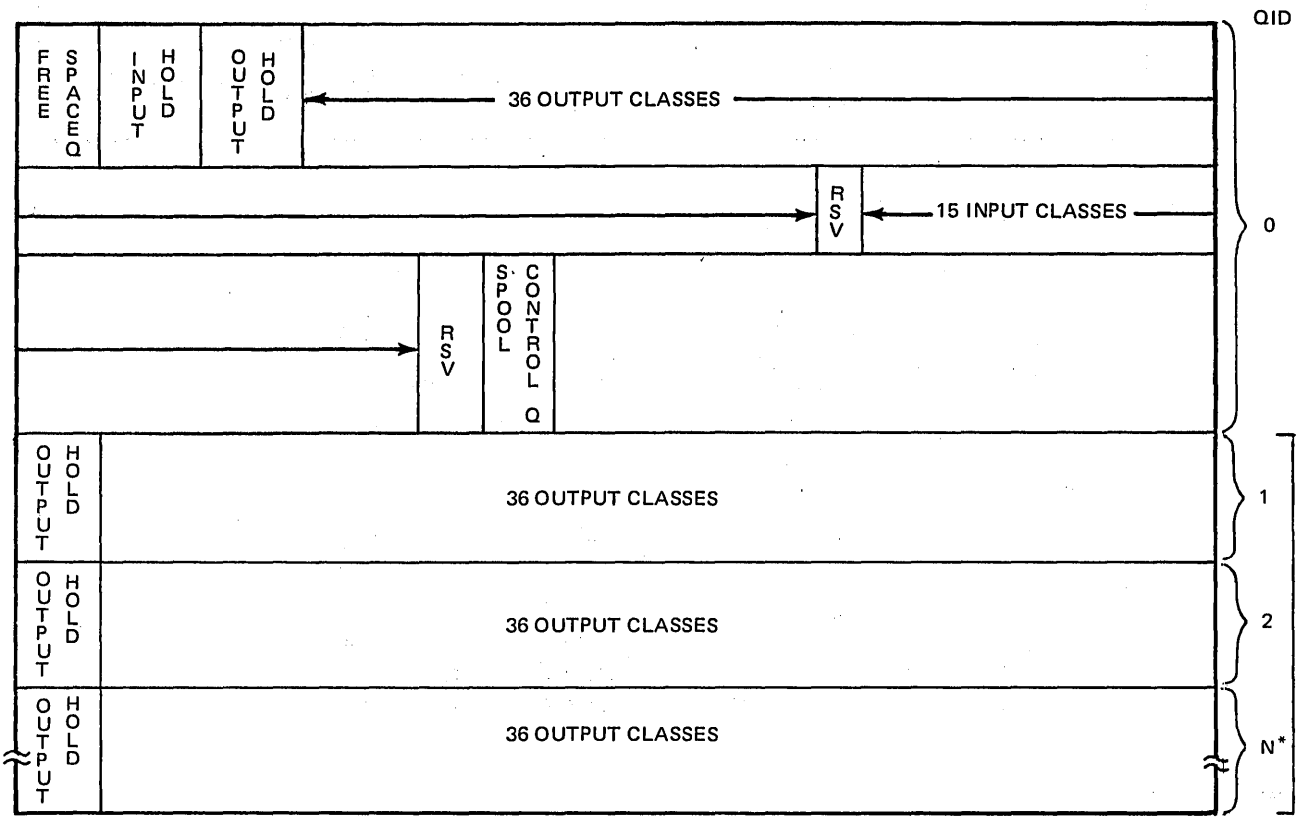
Flags and Masks

FLAG FIELD	CONTAINS
PDQDSNAM	FIRST DATA SET NAME
PDQBTWKA	TERMINATE WORK AREA

MASK NAME	VALUE	MEANS
PDQDSNML	'2C'	LENGTH OF DS NAME FIELD
PDQISTAT	'80'	INITIAL STATUS INDICATOR (OLD IF ON)
PDQCSTAT	'40'	CURRENT STATUS INDICATOR (OLD IF ON)
PDQPASIN	'20'	PASS SATISFIED INDICATOR

Queue Control Records (QCR) on SYS1.SYSJOBQE

The Queue Control Records (QCR) occupy 36 bytes. A QCR exists for each class of information residing on SYS1.SYSJOBQE. Additional QCRs are present if RTAM support is in the system. The QCR contains pointers to the Logical Track Header (LTH) records.



QCR LENGTH = 36 BYTES

*SYS1.SYSJOBQE HAS ADDITIONAL QCR's WHEN RTAM SUPPORT IS IN THE SYSTEM.

OFFSET

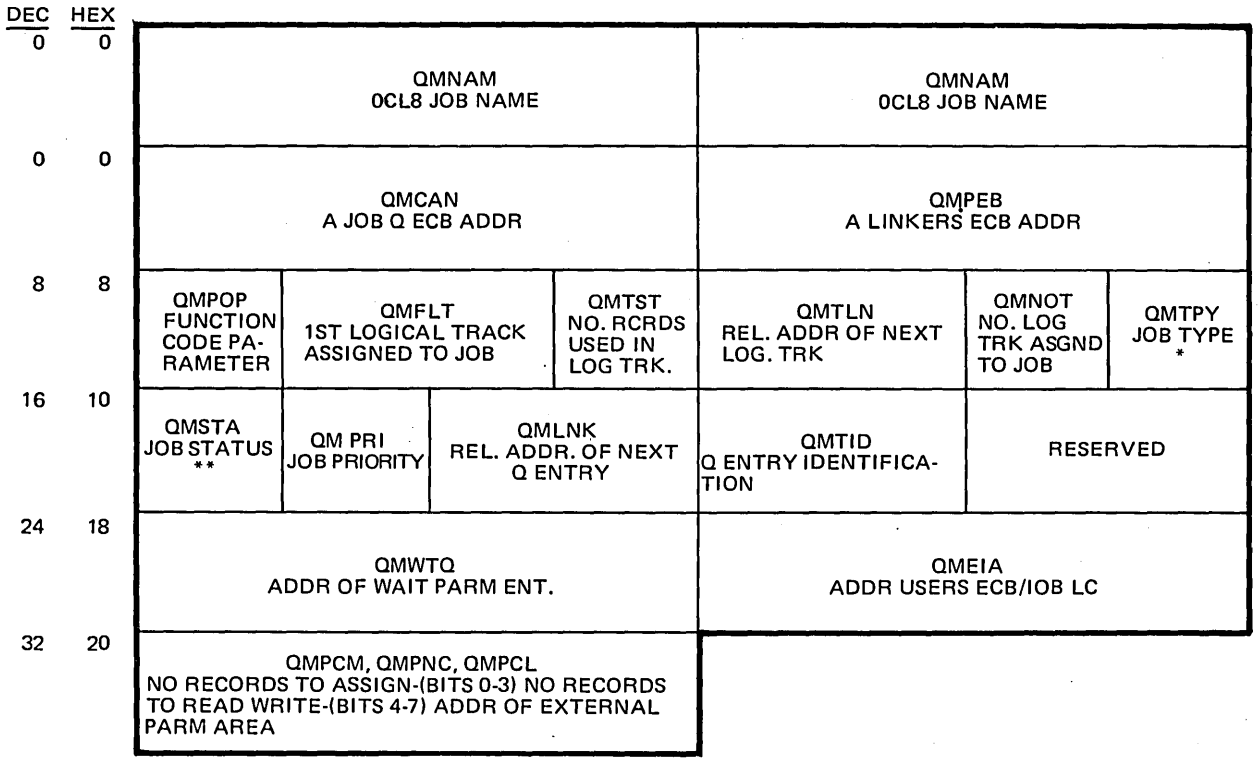
HEX	DEC				
0	0	ADDRESS OF LAST LTH OF HIGHEST PRIORITY ENTRY ON QUEUE	2	14	2
4	4	13	2	12	2
8	8	11	2	10	2
C	12	9	2	8	2
10	16	7	2	6	2
14	20	5	2	4	2
18	24	3	2	2	2
1C	28	1	2	0	2
20	32	HOLD QUEUE	HIGHEST PRIORITY [1]	ADDRESS OF ECB FOR FIRST REQUESTING WORK	3

QCR RECORD FORMAT

Queue Manager Parameter Area (QMPA)

The Queue Manager Parameter Area (QMPA), having a length of 36 bytes, describes the parameter list passed to the queue manager for the following functions: read, write, assign, assign/start, dequeue, enqueue, and delete. Whoever requests one of these queue manager functions can build and delete QMPA whenever one of these functions is required.

QMPA Storage Map



* 1=HELDINPUT, 2=NOT USED, 3-38=SYSOUT CLASS, 39=NOT USED, 40-54=INPUT CLASSES
** 0=READY, 1=CANCEL, 2=PRIORITY CHANGE

Fields in QMPA — by displacement

DEC	HEX	FIELD
0	0	QMNAM
4	4	QMCAN
8	8	QMPEB
9	9	QMPOP
11	B	QMFLT
12	C	QMTST
14	E	QMTLN
15	F	QMNOT
16	10	QMTPY
17	11	QMSTA
18	12	QMPRI
20	14	QMLNK
24	18	QMTID
26	1C	QMWTQ
32	20	QMEIA
32	20	QMPCM
32	20	QMPNC
32	20	QMPCL

Alphabetical list of fields in QMPA

FIELD	DEC	HEX
QMCAN	0	0
QMEIA	26	1C
QMFLT	9	9
QMLNK	18	12
QMNAM	0	0
QMNOT	14	E
QMPCL	32	20
QMPCM	32	20
QMPEB	4	4
QMPNC	32	20
* QMPOP	8	8
* QMPRI	17	11
* QMSTA	16	10
QMTID	20	14
QMTLN	12	C
QMTPY	15	F
QMTST	11	B
QMWTQ	24	18

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
QMPOP	FUNCTION CODE PARAMETER	QMASGN	1	ASSIGN
		QMASGS	0	ASSIGN/START WITH ASSIGN
		QMWRTA	2	WRITE AND ASSIGN
		QMWRTE	3	WRITE
		QMREAD	4	READ
		QMWREN	5	WRITE/END
		QMDTOP	6	DEQUEUE TOP
		QMDTYP	7	DEQUEUE BY TYPE
		QMDELE	8	DELETE
		QMSTRT	9	ASSIGN/START-NO ASSIGN
		QMWRTV	10	WRITE-VARIABLE LENGTH RECORD
		QMREDV	11	READ-VARIABLE LENGTH RECORD
QMSTA	JOB STATUS	QMSTDS	64	SCHEDULER WORK AREA DATA SET INDICATOR
			32	RETURN ON QUEUE FULL INSTEAD OF WAITING FOR SPACE
QMPRI			64	STARTED TASK REQUEST
			32	IF SWADS REQUEST MEANS INTERPRETER IS THE REQUESTER IF JOB QUEUE REQUEST MEANS INITIATOR OR A JOB MANAGEMENT COMPONENT CALLED BY THE INITIATOR IS THE REQUESTER.

Queue Manager Resident Area (QMRES)

IEFQMRES is the mapping macro which describes the queue manager resident data area. The storage for QMRES is defined in the link edit of the nucleus and is pointed to from the CVT (label CVTJOB). QMRES is never deleted and has a length of 152 bytes.

QMRES Storage Map

DEC	HEX				
		IEFQMRES			
0	0	QMRDCB QUEUE MANAGER DCB			
8	8	H'0'	QMWTPLMT OR QMINIT INIT.COUNT	X'00'	8'F0'
16	10				A (QMRDEB) DEB POINTER
48	30	F'0'			QMRDEB 6'F0' QUEUE MANAGER DEB
56	38			X'0F' DEB ID	AL3 (QMRDCB) DCB POINTER
80	50	QMRAPG APPENDAGE POINTER			QMRUCB UCB POINTER
88	58	H'0'	QMRSCC Q-MGR EXTENT START (CC)	QMRSHH Q-MGR EXTENT START (HH)	QMRECC Q-MGR EXTENT END (CC)
96	60	QMREHH Q-MGR EXTENT END (HH)	QMRNTR Q-MGR NO. OF ASSIGNED TRACKS	QMHDA DISK ADDR. OF M.H. (MBBCCHHR)	
104	68	QMHDA DISK ADDR. OF M.H. (MBBCCHHR)		X'00'	QMFLTM NN OF FIRST LTRK AVAILABLE QMQBK QUEUE BREAKING INFO.
112	70	QMTLNM TOTAL NO. OF LTRKS IN Q EXTENT	QMNOTM NUMBER OF LTRKS AVAILABLE	QMHKT THRSHLD OF LTRKS FOR OVERFLOW	QMTBT TOTAL THRSHLD OF LTRKS + K BIGT
120	78	QMTIDM NN OF LAST LTRK AVAILABLE	QMKTT NN OF 1ST LTRK OF ALL JOBQ	QMHPT NUMBER OF HANDLES PER PHYS. TRACK	QMRPT NUMBER OF RECORDS PER PHYS. TRACK
128	80	QMLPT NUMBER OF RECORDS PER LOG. TRK.	QMTRS THRSHLD OF LTRKS SAVED PER INIT	QMNHM NUMBER OF HANDLES ON MIXED TRK	QMFQR NN OF 1ST JOBQ RECORD (QMKTT)

QMRES Storage Map (Cont.)

DEC	HEX					
136	88	QMECBA ECB FOR NO SPACE		QMFIBLIM MAX LIMIT OF TRKS TSO FIELDS-NOT USED IN VS1	QMFIBMAX MAX NO. OF TRKS TSO FIELDS-NOT USED IN VS1	
144	90	QMFIBCUR CURRENT NO ENTRIES IN SUBMIT Q (TSO FIELDS-NOT USED IN VS1	QMTJID TJID OF JOB WAITING FOR WORK (TSO FIELDS-NOT USED IN VS1	RPS INDICATOR (X'10')	RPS DEVICE TYPE (DEVTYPE MACRO)	RESERVED

Fields in QMRES– by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	QMRDCB	109	6D	QMFLTM	130	82	QMTRS
10	A	QMWTPLMT	111	6F	QMQBK	132	84	QMNHM
52	34	QMRDEB	112	70	QMTLNM	134	86	QMFQR
80	50	QMRAPG	114	72	QMNOTM	136	88	QMECBA
84	54	QMRUCB	116	74	QMHKT	140	8C	QMFIBLIM
90	5A	QMRSCC	118	76	QMTBT	142	8E	QMFIBMAX
92	5C	QMRSHH	120	78	QMTIDM	144	90	QMFIBCUR
94	5E	QMRECC	122	7A	QMKTT	146	92	QMTJID
96	60	QMREHH	124	7C	QMHPT	148	94	QMRPS
98	62	QMRNTR	126	7E	QMRPT			
100	64	QMHDA	128	80	QMLPT			

Alphabetical list of fields in QMRES

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
QMECBA	136	88	QMNHM	132	84	QMRPT	126	7E
QMFIBCUR	144	90	QMNOTM	114	72	QMRSCC	90	5A
QMFIBLIM	140	8C	* QMQBK	111	6F	QMRUCB	84	54
QMFIBMAX	142	8E	QMRAPG	80	50	QMTBT	118	76
QMFLTM	109	6D	QMRDCB	0	0	QMTIDM	120	78
QMFQR	134	86	QMRDEB	52	34	QMTJID	146	92
QMHDA	100	64	QMRECC	94	5E	QMTLNM	112	70
QMHKT	116	74	QMREHH	96	60	QMTRS	130	82
QMHPT	124	7C	QMRNTR	98	62	QMWTPLMT	10	A
QMKTT	122	7A	QMRPS	148	94			
QMLPT	128	80	QMRSHH	92	5C			

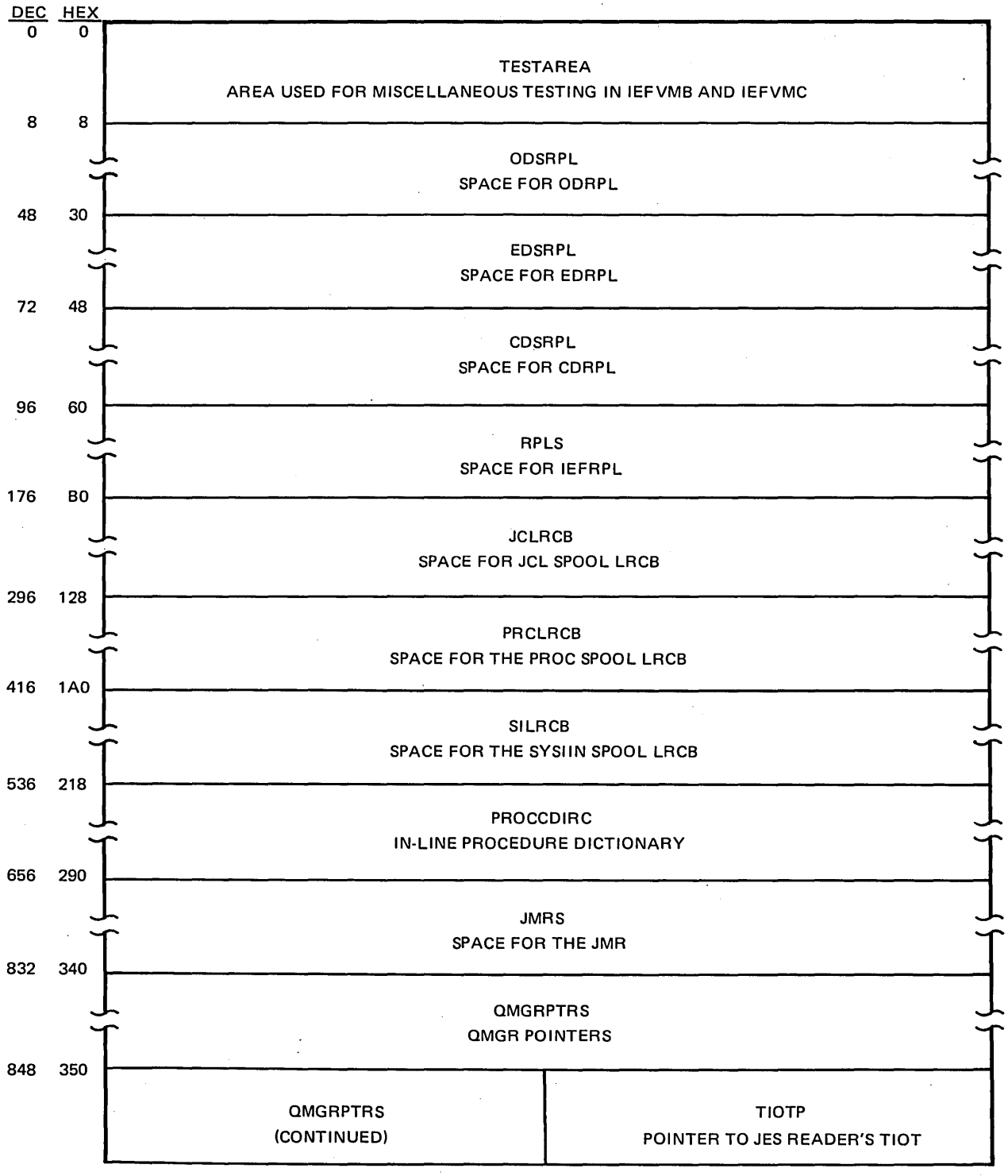
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
QMQBK	QUEUE BREAKING INFORMATION	QMRSV	01	BIT7=ON-SPACE RESERVED TO START INITIATOR

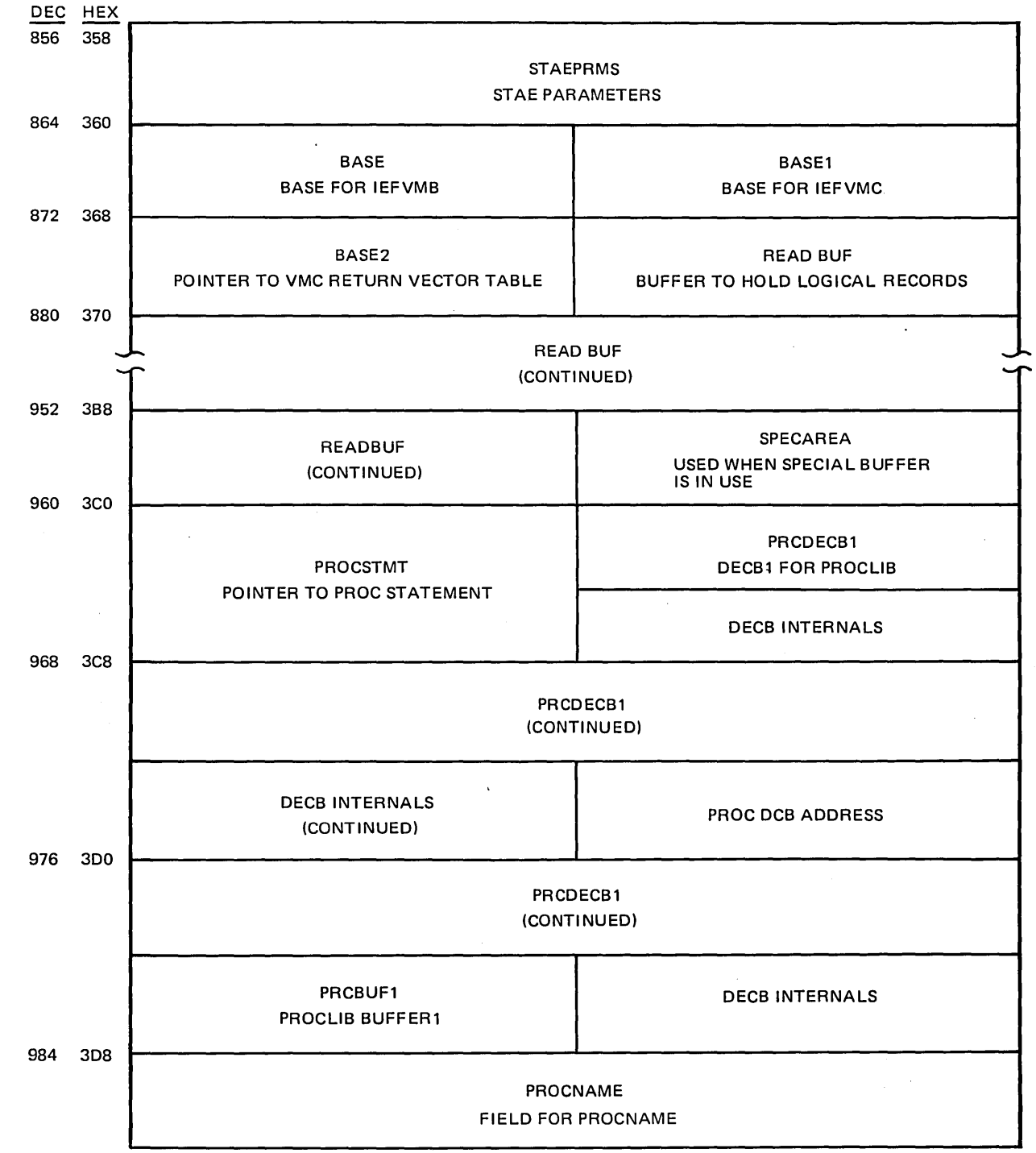
Reader Work Area

The Reader Work Area is allocated with the writer work area at master sheduler initialization (MSI) time according to the JESPARMS RDR value R. The module IEFVMA allocates and deallocates the work area at reader start/stop time. This work area is always on a 2K boundary. It contains all information used by the reader.

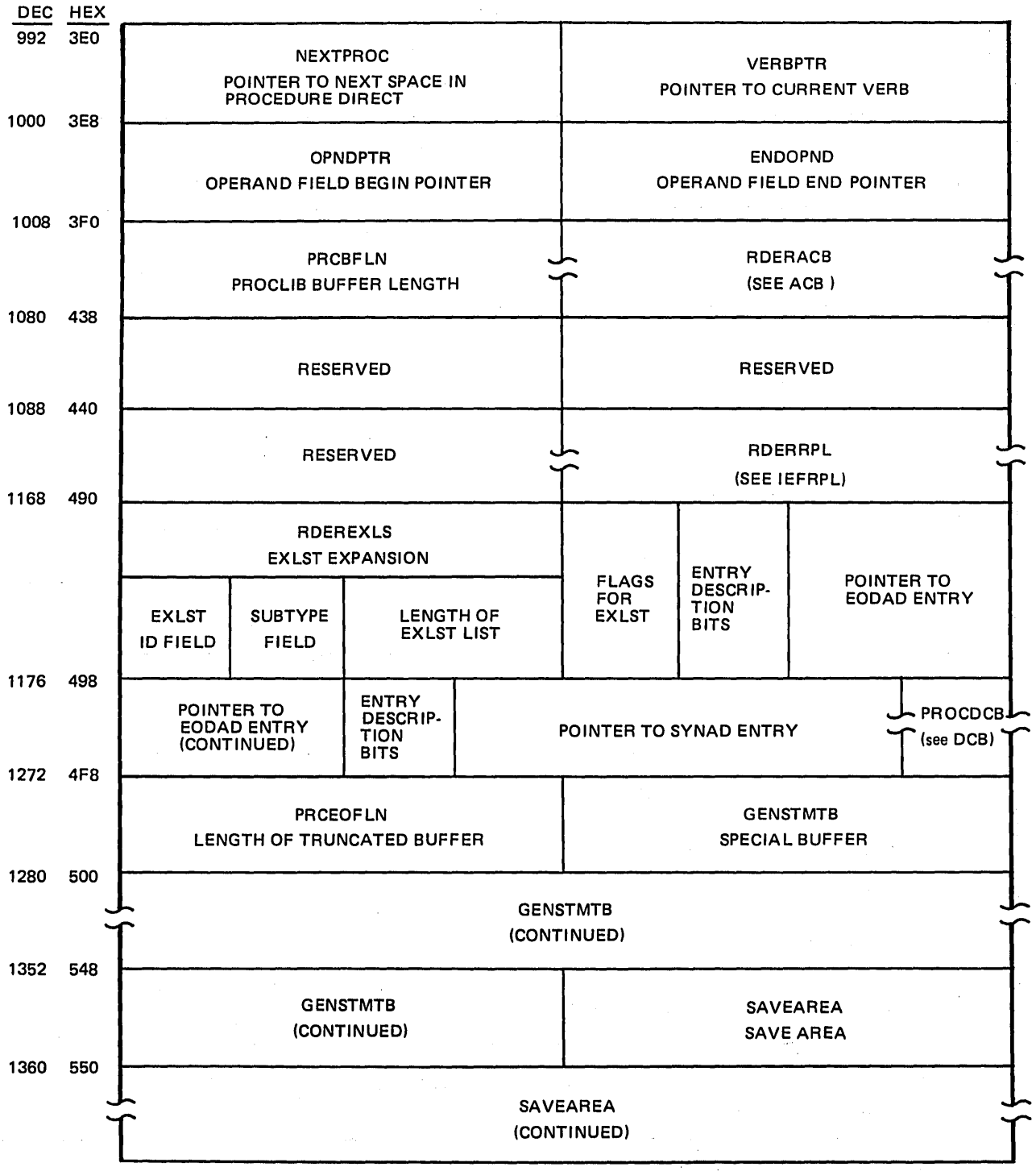
Reader Work Area Storage Map



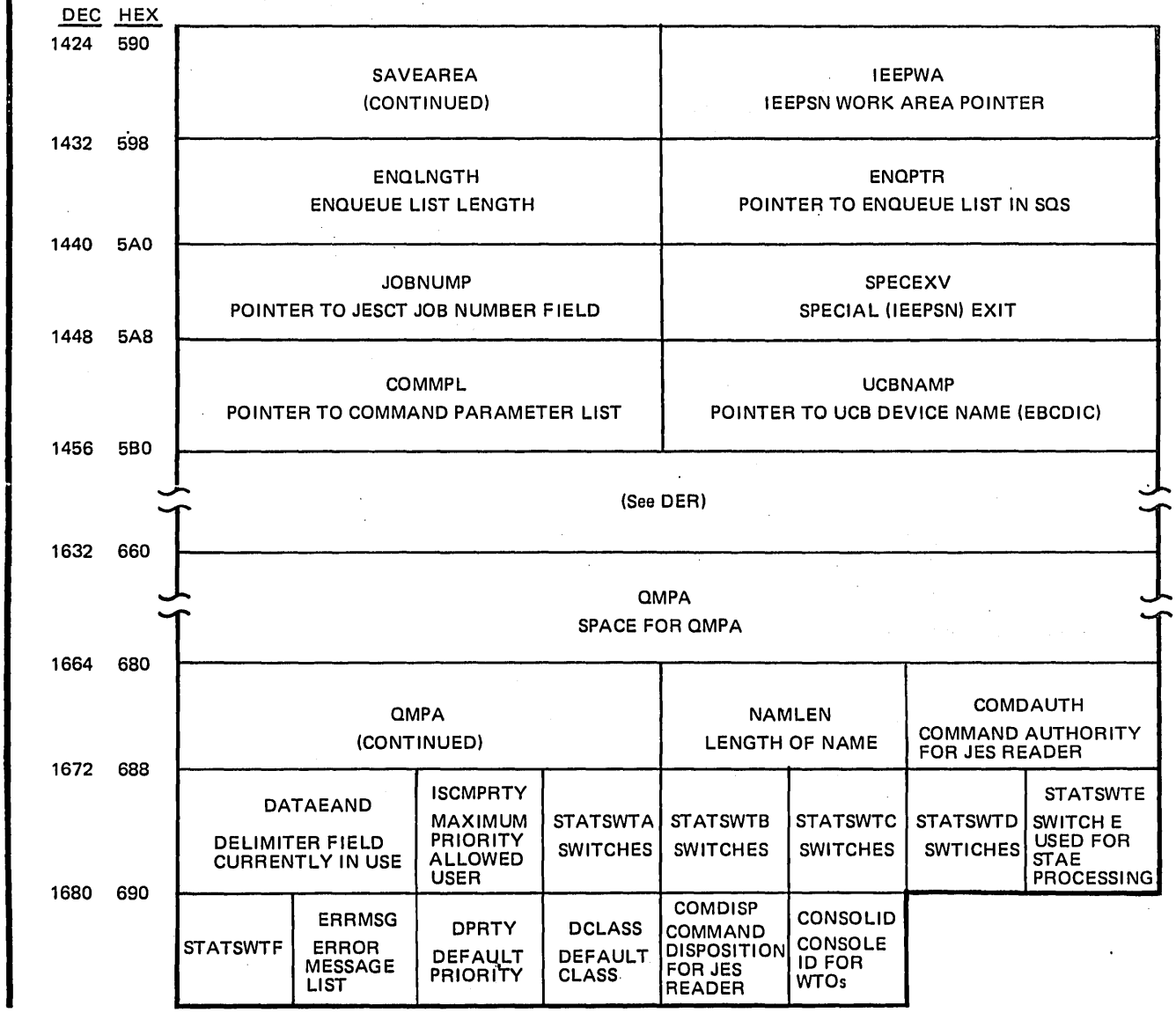
Reader Work Area Storage Map (Contd.)



Reader Work Area Storage Map (Contd.)



Reader Work Area Storage Map (Contd.)



Fields in Reader Work Area — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	TESTAREA	1168	490	RDEREXLS
8	8	ODSRPL	1183	49F	PROCDCB
48	30	EDSRPL	1272	4F8	PRCEOFLN
72	48	CDSRPL	1276	4FC	GENSTMTB
96	60	RPLS	1356	54C	SAVEAREA
176	B0	JCLRCB	1428	594	IEEPWA
296	128	PRCLRCB	1432	598	ENQLNGTH
416	1A0	SILRCB	1436	59C	ENQPTR
536	218	PROCCDIRC	1440	5A0	JOBNUMP
656	290	JMRS	1444	5A4	SPECEXV
832	340	OMGRPTRS	1448	5A8	COMMPL
852	354	TIOTP	1452	5AC	UCBNAMP
856	358	STAEPRMS	1632	660	QMPA
864	360	BASE	1668	684	NAMLEN
868	364	BASE1	1670	686	COMDAUTH
872	368	BASE2	1672	688	DATAEND
876	36C	READBUF	1674	68A	ISCMPRTY
956	3BC	SPECAREA	1675	68B	STATSWTA
960	3C0	PROCSTMT	1676	68C	STATSWTB
964	3C4	PRCDECB1	1677	68D	STATSWTC
976	3D0	PRCBUF1	1678	68E	STATSWTD
984	3D8	PROCNAME	1679	68F	STATSWTE
992	3E0	NEXTPROC	1680	690	STATSWTF
996	3E4	VERBPTR	1681	691	ERRMSG
1000	3E8	OPNDPTR	1682	692	DPRTY
1004	3EC	ENDOPND	1683	693	DCLASS
1008	3F0	PRCBFLN	1684	694	COMDISP
1012	3F4	RDERACB	1685	695	CONSOLID
1092	444	RDERRPL			

Alphabetical list of fields in Reader Work Area

FIELD	DEC	HEX	FIELD	DEC	HEX
BASE	864	360	PRCEOFLN	1272	4F8
BASE1	868	364	PRCLRCB	296	128
BASE2	872	368	PROCCDIRC	536	218
CDSRPL	72	48	PROCDCB	1183	49F
COMDAUTH	1670	686	PROCNAME	984	3D8
COMDISP	1684	694	PROCSTMT	960	3C0
COMMPL	1448	5A8	QMGRPTRS	832	340
CONSOLID	1685	695	QMPA	1632	660
DATAEND	1672	688	RDEREXLS	1168	490
DCLASS	1683	693	RDERRPL	1092	444
DPRTY	1682	692	RDERACB	1012	3F4
EDSRPL	48	30	READBUF	876	36C
ENDOPND	1004	3EC	RPLS	96	60
ENQLNGTH	1432	598	SAVEAREA	1356	54C
ENQPTR	1436	59C	SILRCB	416	1A0
ERRMSG	1681	691	SPECAREA	956	3BC
GENSTMTB	1276	4FC	SPECEXV	1444	5A4
IEEPWA	1428	594	STAEPRMS	856	358
ISCMPRTY	1674	68A	STATSWTA	1675	68B
JCLRCB	176	B0	STATSWTB	1676	68C
JMRS	656	290	STATSWTC	1677	68D
JOBNUMP	1440	5A0	STATSWTD	1678	68E
NAMLEN	1668	684	STATSWTE	1679	68F
NEXTPROC	992	3E0	STATSWTF	1680	690
ODSRPL	8	8	TESTAREA	0	0
OPNDPTR	1000	3E8	TIOTP	852	354
PRCBFLN	1008	3F0	UCBNAMP	1452	5AC
PRCBUF1	976	3D0	VERBPTR	996	3E4
PRCDECB1	964	3C4			

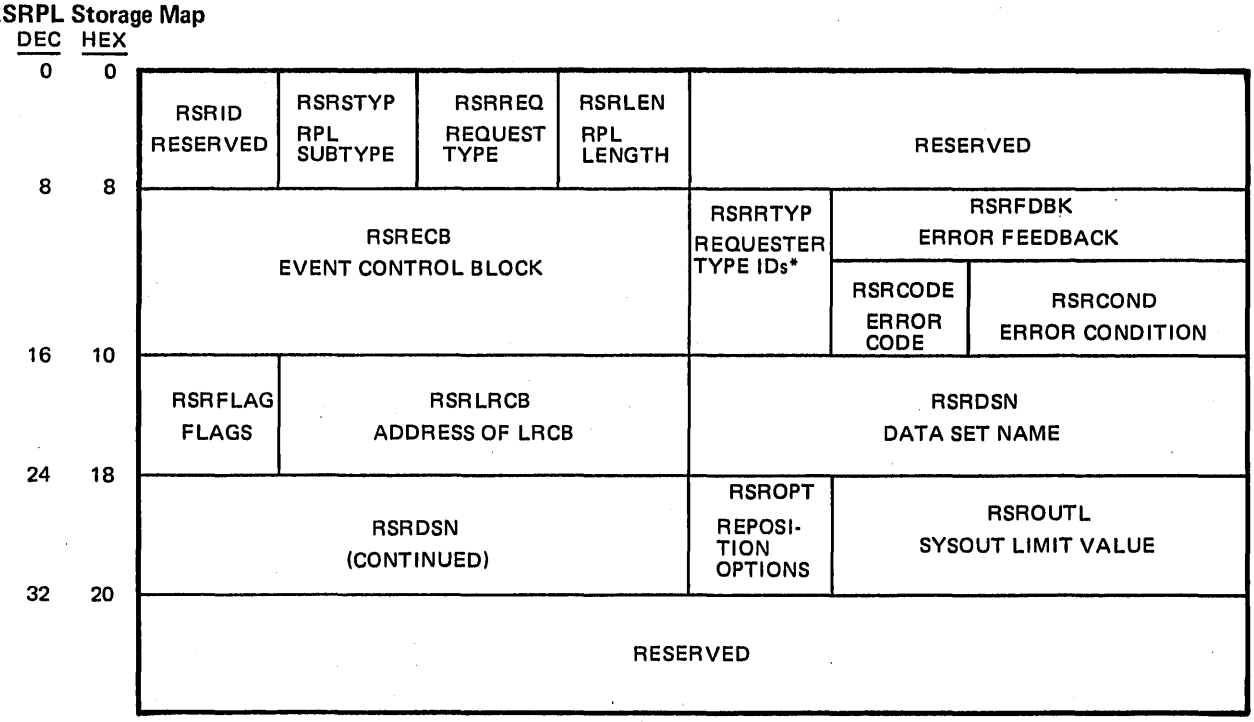
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
ACBID	ACB IDENTIFIER	ACBIDVAL	'A0'	IDENTIFIER VALUE
GENSTMTB	SPECIAL BUFFER	DSNAME	'04FC'	TEMPORARY SPOOL DSNAME THIS DD
		SYSINCT	'0504'	COUNT OF DATA RECORDS THIS DD
STATSWTA		STARTJOB	'80'	BEGIN PROCESSING AT USER GIVEN JOB NAME
		FLUSH	'40'	FLUSH ALL SYSIN
		QOPEN	'20'	JOBQ OPEN
		JCLDATA	'10'	JCL DATA WITHIN SYSIN ALLOWED
		INPROC	'08'	IN-LINE PROC BEING PROCESSED
		HOLD	'04'	MUST PUT JOB ON HOLD QUEUE
		EXECONT	'01'	EXEC CONTINUATION BYPASS SW
STATSWTB		SPLRTN	'80'	RETURNING FROM SPOOL
		EOD	'40'	SPOOL EOD
		NTYPRUN	'20'	NEED TYPRUN INFORMATION
		NCLASS	'10'	NEED CLASS INFORMATION
		NPRTY	'08'	NEED PRTY INFORMATION
		NDLM	'04'	NEED DLM INFORMATION
		QUOTE	'02'	IN QUOTE SWITCH
		SPACEX	'01'	HAVE SPECIAL (IEEPSN) EXIT
STATSWC		CMCONTEX	'80'	COMMENT CONTINUATION EXPECTED ON NEXT STATEMENT
		OPCONTEX	'40'	OPERAND CONTINUATION EXPECTED ON NEXT STATEMENT
		CMCONT	'20'	COMMENT CONT'D FROM LAST STMNT
		OPCONT	'10'	OPERAND CONT'D FROM LAST STMNT
		JCL	'08'	HAVE JCL STATEMENT
		COMMENT	'04'	A COMMENT STMNT (//*)
		NULL	'02'	A NULL STMNT (//)
		INVALID	'01'	INVALID FORMAT ON JCL STMNT (NO OPERAND)
STATSWTD		ATOPND	'80'	SEARCHING FOR STMNT OPERAND
		GNSTMT	'40'	CURRENT RECORD IS GENERATED STM
		DDSW	'20'	SPOOL SW
		JOBCONT	'10'	JOB CARD TO BE CONT'D
		WTRSTRT	'08'	ALLOW ONLY WTR TO START
		PROCBUF1	'04'	NEED READ FROM PROCLIB INTO BUF1
		PRCTIME1	'02'	FIRST READ FOR PROCLIB
		RDREOF	'01'	READER INPUT EOF SW
STATSWTE	SWITCH E (USED FOR STAE PROCESS)	RETRY	'80'	TASK IN RETRY
		ODSJL	'40'	JL DS OPEN
		ODSPL	'20'	PL DS OPEN
		CDS	'10'	CDS TO BE DONE
		ODSSI	'08'	SI DS OPEN
		RECUR	'04'	RECURSIVE ABEND SW
		DEL	'02'	CYL DELETE TO BE DONE
		SMFEX	'01'	SMF EXIT TO BE TAKEN
STATSWF		RDROPEN	'80'	RDR DCB OPEN
		PRCOPEN	'40'	PROC DCB OPEN
		DSINTG	'20'	DS INTEGRITY DONE
		PRCSPOOL	'10'	PROCEDURES ARE TO BE SPOOLED
		WTRSTRTS	'08'	SPOOL ALLOW WRITER TO START
		INTSTRTQ	'04'	QUEUE ALLOW INIT TO START
		RTRDR	'02'	REMOTE INPUT DEVICE

Reposition Data Set Parameter List (RSRPL)

The user passes the Reposition Data Set Parameter List (RSRPL) to JECS to request the repositioning function (IEFSMREP). The user allocates, initializes, and frees the storage required by the RSRPL. RSRPL requires 40 bytes of storage.

RSRPL Storage Map



* The following are valid requester type IDs:

X'01' - Interpreter
X'02' - Allocation
X'03' - WTP
X'04' - Termination
X'05' - Transient
X'06' - Restart reader in problem program partition
X'07' - DSO
X'08' - Data Management
X'09' - Initiator
X'0A' - Checkpoint/Restart
X'0B' - System restart
X'0C' - SWADS
X'10' - JES writer on behalf of user writer
X'40' - Special data set
X'90' - JES writer
X'00' - JES reader

Fields in RSRPL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	RSRID	13	D	RSRCODE
1	1	RSRSTYP	14	E	RSRCOND
2	2	RSRREQ	16	10	RSRFLAG
3	3	RSRLEN	17	11	RSRLRCB
8	8	RSRECB	20	14	RSRDSN
12	C	RSRRTYP	28	1C	RSROPT
13	D	RSRFDBK	29	1D	RSROUTL

Alphabetical list of fields in RSRPL

FIELD	DEC	HEX	FIELD	DEC	HEX
RSRCODE	13	D	RSRLEN	3	3
RSRCOND	14	E	RSRLRCB	17	11
RSRDSN	20	14	RSROPT	28	1C
RSRECB	8	8	RSROUTL	29	1D
RSRFDBK	13	D	RSRREQ	2	2
RSRFLAG	16	10	RSRRTYP	12	C
RSRID	0	0	RSRSTYP	1	1

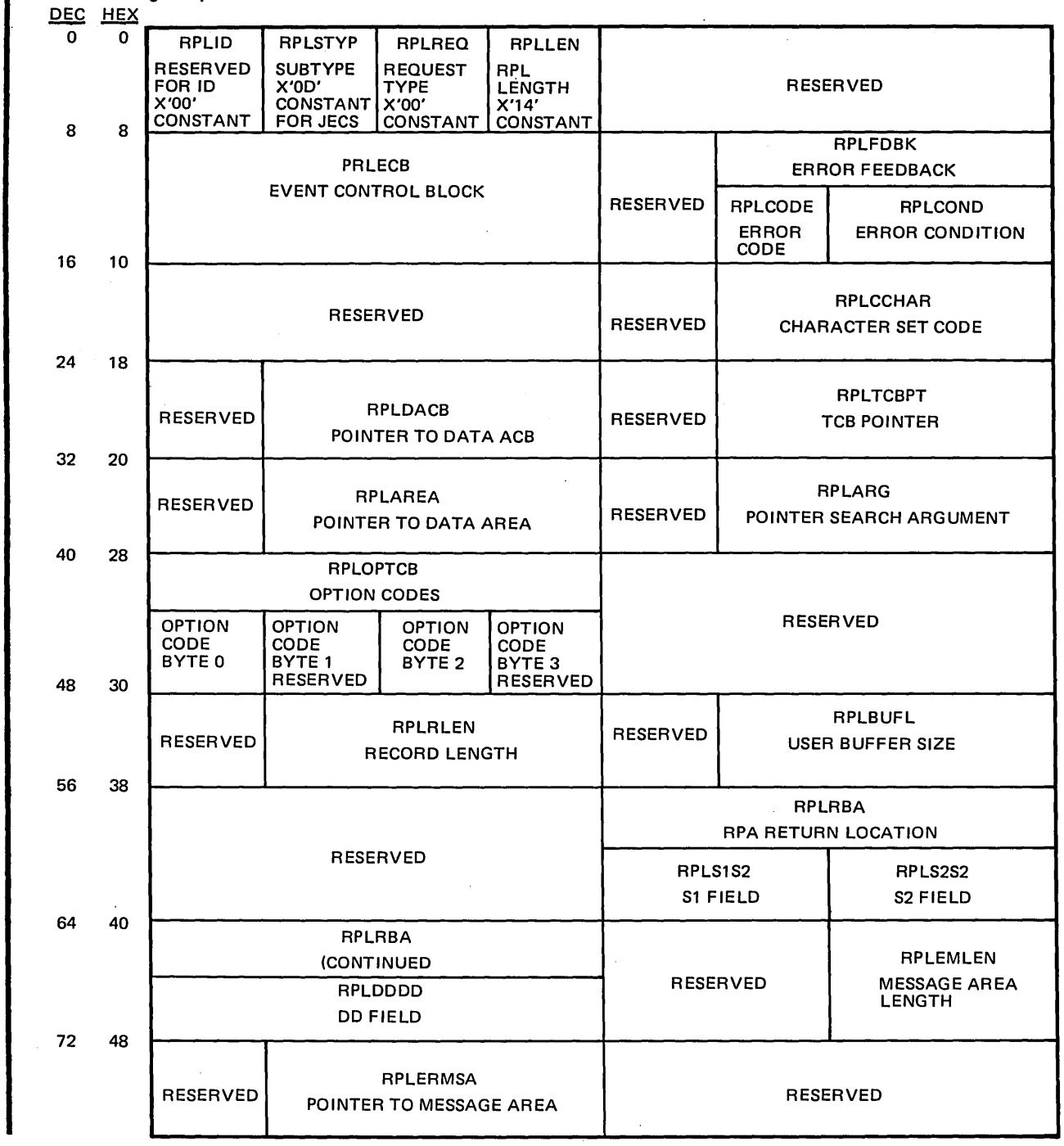
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
RSRFLAG	FLAGS	RSROUT	'20'	1 = OUTPUT
		RSRINPT	'10'	1 = INPUT
RSROPT	REPOSITION OPTIONS	RSRRST	'80'	1 = IMMEDIATE RESTART 0 = DEFERRED RESTART

Request Parameter List (IEFRPL)

The user passes the Request Parameter List (IEFRPL) to JECS to request a GET (IEFSMGET), PUT (IEFSMPUT), or POINT (IEFSMGET) function. A spool user allocates and frees the space required by this list before going to the JECS request router (IEFSMIFC). IEFRPL requires 80 bytes of storage.

IEFRPL Storage Map



Fields in IEFRPL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	RPLID	33	21	RPLAREA
1	1	RPLSTYP	37	25	RPLARG
2	2	RPLREQ	40	28	RPLOPTCB
3	3	RPLLEN	49	31	RPLRLEN
8	8	RPLECB	53	35	RPLBUFL
13	D	RPLFDBK	60	3C	RPLRBA
13	D	RPLCODE	60	3C	RPLS1S2
14	E	RPLCOND	62	3E	RPLS2S2
21	15	RPLCCHAR	64	40	RPLDDDD
25	19	RPLDACB	70	46	RPLEMLEN
29	1D	RPLTCBPT	73	49	RPLERMSA

Alphabetical list of fields in IEFRPL

FIELD	DEC	HEX	FIELD	DEC	HEX
RPLAREA	33	21	RPLFDBK	13	D
RPLARG	37	25	RPLID	0	0
RPLBUFL	53	35	RPLLEN	3	3
RPLCCHAR	21	15	RPLOPTCB	40	28
RPLCODE	13	D	RPLRBA	60	3C
RPLCOND	14	E	RPLREQ	2	2
RPLDACB	25	19	RPLRLEN	49	31
RPLDDDD	64	40	RPLSTYP	1	1
RPLECB	8	8	RPLS1S2	60	3C
RPLEMLEN	70	46	RPLS2S2	62	3A
RPLERMSA	73	49	RPLTCBPT	29	1D

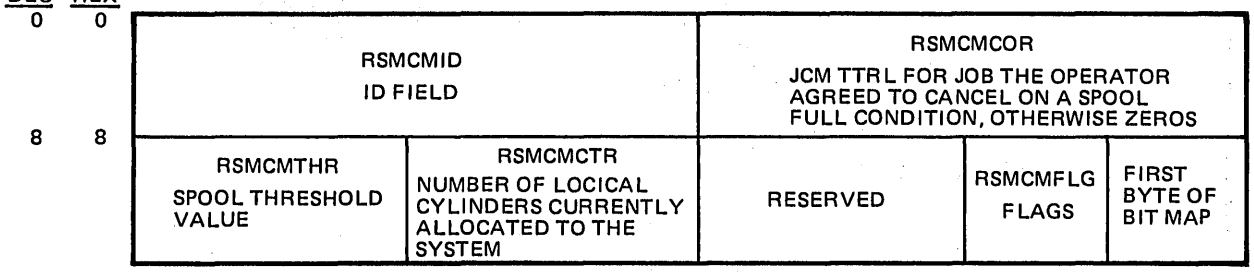
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
RPLOPTCD	OPTION CODE	RPLSEQ	'20'	ADDRESS SEQUENTIAL REQUEST
		RPLASY	'08'	ASYNCHRONOUS
		RPLEODS	'80'	END OF DATA SET INDICATOR
		RPLSFORM	'40'	SPECIAL POINTER FORMS
		RPLBLK	'20'	BLOCK DATA CHECKS
		PPLVFY	'10'	VERIFY BUFFER LOAD
		RPLFLD	'08'	LOAD IN FOLD MODE
		RPLFMT	'02'	UCS/FCB LOAD
		RPLALIGN	'01'	FCB ALIGN

Resident Master Cylinder Map (RMCM)

The Resident Master Cylinder Map (RMCM) contains information reflecting the current status of the spool data set. The bit map portion reflects the logical cylinders currently in use by all jobs. RMCM is created by IEFWAMIN and is never deleted. The RMCM requires a 15-byte header and a variable number of bytes for the map. (One bit is required in the map for each logical cylinder.)

RMCM Storage Map



Fields in RMCM — by displacement

DEC	HEX	FIELD
0	0	RSMCMID
4	4	RSMCMCOR
8	8	RSMCMTHR
10	A	RSMCMCTR
14	E	RSMCMFLG

Alphabetical list of fields in RMCM

FIELD	DEC	HEX
RSMCMCOR	4	4
RSMCMCTR	10	A
RSMCMFLG	14	E
RSMCMID	0	0
RSMCMTHR	8	8

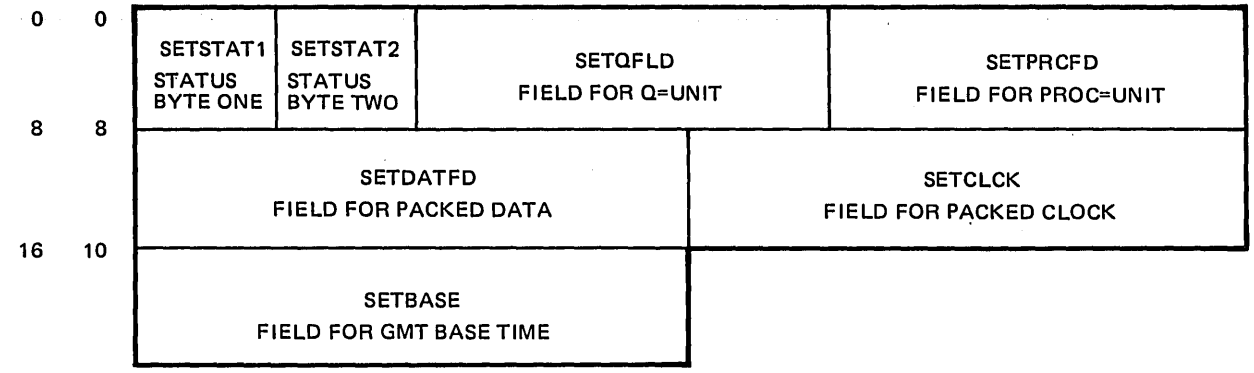
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
RSMCMFLG	FLAGS	RSMCM001	'80'	START INITIATOR/WRITER MESSAGE HAS BEEN ISSUED
		RSMCM003	'20'	JOB QUEUE IS BEING HELD

Set Parameter List (SETLT)

The Set Parameter List (SETLT) is created during the execution of the nucleus initialization program (NIP). It is used to pass the set parameters to master scheduler initialization (MSI) and the set parameter processor. The SETLT, which is 20 bytes long, is deleted at the end of MSI.

SETLT Storage Map



Fields in SETLT — by displacement

DEC	HEX	FIELD
0	0	SETSTAT1
1	1	SETSTAT2
2	2	SETQFLD
5	5	SETPRCFD
8	8	SETDATFD
12	C	SETCLCK
16	10	SETBASE

Alphabetical list of fields in SETLT

FIELD	DEC	HEX
SETBASE	16	10
SETCLCK	12	C
SETDATFD	8	8
SETPRCFD	5	5
SETQFLD	2	2
SETSTAT1	0	0
SETSTAT2	1	1

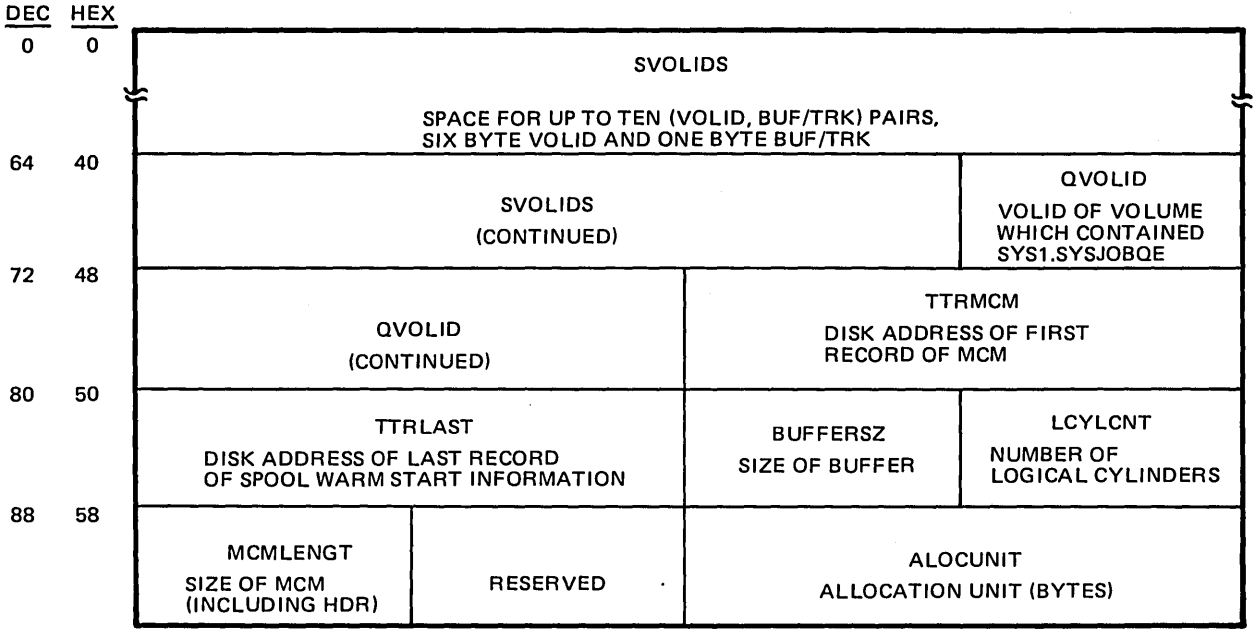
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SETSTAT1	STATUS BYTE NO. 1	SETQKYWD	'80'	INDIC Q KYWD SPECIFIED
		SETPROC	'40'	INDIC PROC KYWD SPECIFIED IN NIP
		SETSPOOL	'20'	INDIC SPOOL KYWD SPECIFIED
		SETQPRMS	'10'	INDIC QPARM KYWD SPECIF IN MEMB
		SETAUTO	'08'	INDIC AUTO/RDR KYWDS SPEC IN NIP
		SETDATE	'04'	INDIC DATE KYWD SPECIFIED IN NIP
		SETCLOCK	'02'	INDIC CLOCK KYWD SPECIFIED IN NIP
		SETGMT	'01'	INDIC GMT KYWD SPECIFIED IN NIP
SETSTAT2	STATUS BYTE NO. 2	SETQUNIT	'80'	INDIC Q=UNIT SPECIFIED IN NIP
		SETQF	'40'	INDIC Q=F SPECIFIED IN NIP
		SETCHNG	'20'	INDIC SPOOL=CHNG SPECIF IN NIP
		SETSPOLF	'10'	INDIC SPOOL=F SPECIF IN NIP
		SETTIME	'08'	INDIC DATE MUST BE SPECIF IN NIP
		SETPRMRD	'04'	INDIC PROCESSING A MEMBER

Spool Configuration Control Record (SCCR)

The Spool Configuration Control Record (SCCR) contains information about the system spool data sets. This information is required to rebuild the system at system restart time. The SCCR is built at initialization time and read at system restart time. The SCCR occupies 96 bytes of storage.

SCCR Storage Map



Fields in SCCR — by displacement

DEC	HEX	FIELD
0	0	SVOLIDS
70	46	QVOLID
76	4C	TTRMCM
80	50	TTRLAST
84	54	BUFFERSZ
86	56	LCYLCNT
88	58	MCMLENGT
92	5C	ALOCUNIT

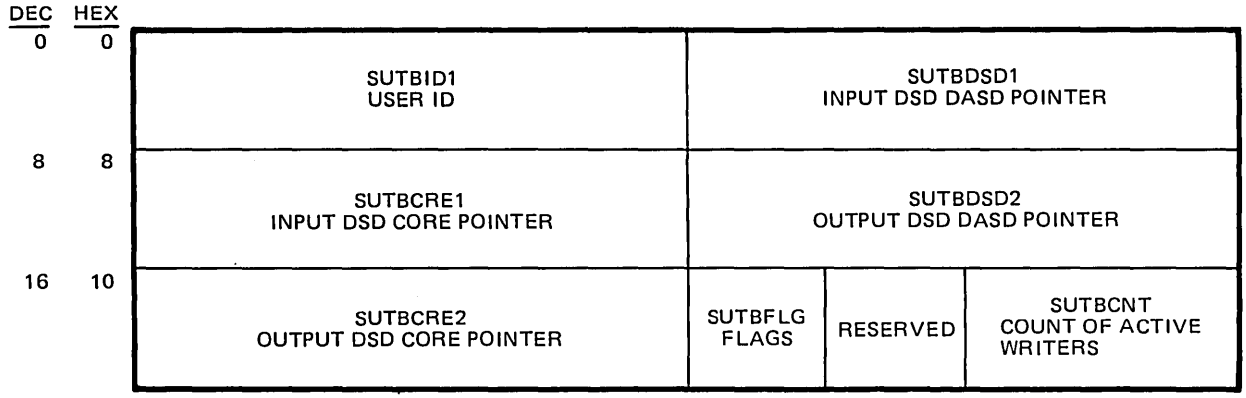
Alphabetical list of fields in SCCR

FIELD	DEC	HEX
ALOCUNIT	92	5C
BUFFERSZ	84	54
LCYLCNT	86	56
MCMLENGT	88	58
QVOLID	70	46
SVOLIDS	0	0
TTRLAST	80	50
TTRMCM	76	4C

Spool User Table (SUTBL)

The Spool User Table (SUTBL) provides an active list of spool users and pointers to control blocks set up on their behalf. IEFSMINT obtains the space, and the routine is never deleted. SUTBL occupies 20 bytes of storage (it expands with multiple users).

SUTBL Storage Map



Fields in SUTBL – by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SUTBID1	12	C	SUTBDSD2	20	14	SUTBFLG
4	4	SUTBDSD1	16	10	SUTBCRE2	22	16	SUTBCNT
8	8	USTBCRE1						

Alphabetical list of fields in SUTBL

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
SUTBCNT	22	16	SUTBDSD1	4	4	SUTBTD1	0	0
SUTBCRE1	8	8	SUTBDSD2	12	C	* SUTBFLG	20	14
SUTBCRE2	16	10						

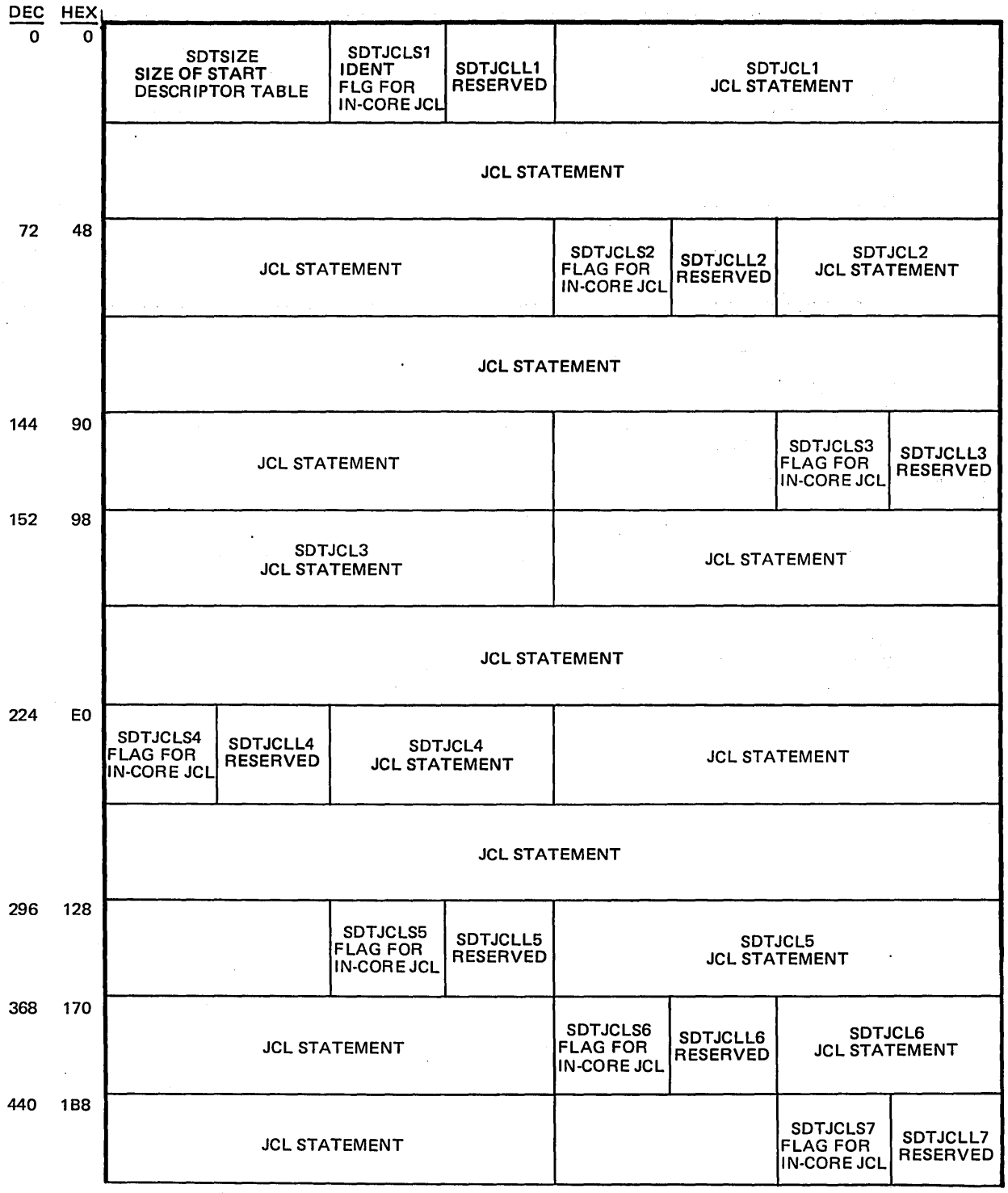
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SUTBFLG	FLAGS	SUTBFLG1	80	BIT0-ENTRY INCOMPLETE
		SUTBFLG2	40	BIT1-I/O FOR DSD FAILED

Start Descriptor Table (SDT)

The Start Descriptor Table (SDT), occupying 532 bytes of storage, consists of up to seven entries, each of which contains a JCL statement constructed by the syntax-check routine of the system task control routine. The SDT is built for each command by the IEEVSTAR routine, and it is deleted by IEEVJCL.

SDT Storage Map



SDT Storage Map (Contd.)

DEC	HEX		
448	1C0	SDTJCL7 JCL STATEMENT	JCL STATEMENT

Fields in SDT – by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SDTSIZE	151	97	SDTJCLL3	300	12C	SDTJCL5
2	2	SDTJCLS1	152	98	SDTJCL3	372	174	SDTJCLS6
3	3	SDTJCL11	224	E0	SDTJCLS4	374	176	SDTJCLL6
4	4	SDTJCL1	225	E1	SDTJCLL4	375	177	SDTJCL6
76	4C	SDTJCLS2	226	F2	SDTJCL4	446	1BE	SDTJCLS7
77	4D	SDTJCLL2	298	12A	SDTJCLS5	447	1BF	SDTJCLL7
78	4E	SDTJCL2	299	12B	SDTJCLL5	448	1C0	SDTJCL7
150	96	SDTJCLS3						

Alphabetical list of fields in SDT

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
SDTJCL1	4	4	SDTJCLL2	77	4D	* SDTJCLS2	76	4C
SDTJCL2	78	4E	SDTJCLL3	151	97	* SDTJCLS3	150	96
SDTJCL3	152	98	SDTJCLL4	225	E1	* SDTJCLS4	224	E0
SDTJCL4	226	E2	SDTJCLL5	299	12B	SDTJCLS5	298	12A
* SDTJCL5	300	12C	SDTJCLL6	374	176	SDTJCLS6	372	174
* SDTJCL6	375	177	SDTJCLL7	447	1BF	SDTJCLS7	446	1BE
* SDTJCL7	448	1C0	* SDTJCLS1	2	2	SDTSIZE	0	0
SDTJCLL1	3	3						

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SDTJCLS1	IDENTIFIER FLAGS FOR IN-CORE JCL	SDTJOB1	128	BIT0-JOB STATEMENT
		SDTEXEC1	64	BIT1-EXEC STATEMENT
		SDTDD1	32	BIT2-DD STATEMENT
		SDTCONT1	16	BIT3-DD STATEMENT CONTINUATION
SDTJCLS2	IDENTIFIER FLAGS FOR IN-CORE JCL	SDTJOB2	128	BIT0-JOB STATEMENT
		SDTEXEC2	64	BIT1-EXEC STATEMENT
		SDTDD2	32	BIT2-DD STATEMENT
		SDTCONT2	16	BIT3-DD STATEMENT CONTINUATION
SDTJCLS3	IDENTIFIER FLAG	SDTJOB3	128	BIT0-JOB STATEMENT
		SDTEXEC3	64	BIT1-EXEC STATEMENT
		SDTDD3	32	BIT2-DD STATEMENT
		SDTCONT3	16	BIT3-DD STATEMENT CONTINUATION
SDTJCLS4	IDENTIFIER FLAG FOR IN-CORE JCL	SDTJOB4	128	BIT0-JOB STATEMENT
		SDTEXEC4	64	BIT1-EXEC STATEMENT
		SDTDD4	32	BIT2-DD STATEMENT
		SDTCONT4	16	BIT3-DD STATEMENT CONTINUATION
SDTJCL5	IDENTIFIER FLAG FOR IN-CORE JCL	SDTJOB5	128	BIT0-JOB STATEMENT
		SDTEXEC5	64	BIT1-EXEC STATEMENT
		SDTDD5	32	BIT2-DD STATEMENT
		SDTCONT5	16	BIT3-DD STATEMENT CONTINUATION
SDTJCL6	IDENTIFIER FLAG FOR IN-CORE JCL	SDTJOB6	128	BIT0-JOB STATEMENT
		SDTEXEC6	64	BIT1-EXEC STATEMENT
		SDTDD6	32	BIT2-DD STATEMENT
		SDTCONT6	16	BIT3-DD STATEMENT CONTINUATION

Flags and Masks (Contd.)

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SDTJCL7	IDENTIFIER FLAG FOR IN-CORE JCL	SDTJOB	128	BIT0-JOB STATEMENT
		SDTEXEC	64	BIT1-EXEC STATEMENT
		SDTDD	32	BIT2-DD STATEMENT
		SDTCONT	16	BIT3-DD STATEMENT CONTINUATION
		SDTEXCON	8	BIT4-EXEC STATEMENT CONTINUATION

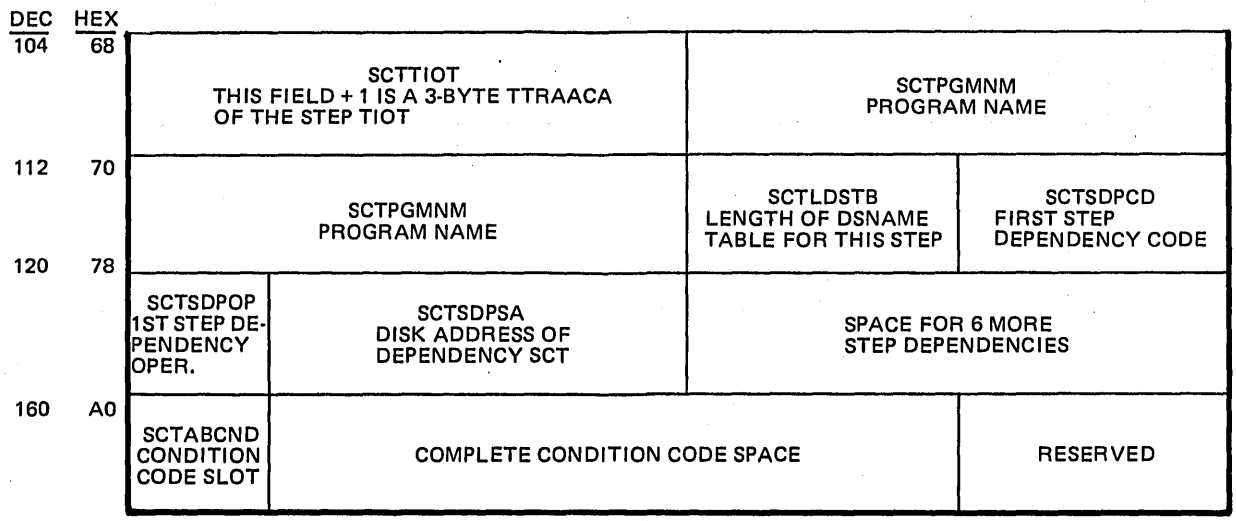
Step Control Table (SCT)

The Step Control Table (SCT) is 176 bytes long and contains information regarding the job step. The Interpreter Module (IEFVEA) builds the SCT as part of the processing of the EXEC statement. It is deleted at termination time.

SCT Storage Map

DEC	HEX				
0	0	SCTDISKA DISK ADDRESS OF SCT	SCTTBLID TABLE ID OF SCT = 4	SCTSSTAT INTERNAL STEP STATUS	SCTSTIME MAXIMUM STEP RUNNING TIME
8	8	SCTSEXEC STEP CODE PASSED TO INIT. AT TERMINATE	SCTLALOC LENGTH OF ALLOC. WK AREA & NO. OF GOOD DDCARDS	SCTFSIOT DISK ADDRESS OF FIRST SIOT	
16	10	SCTAALOC DISK ADDRESS OF ALLOCATION WORKAREA		SCTANSCT DISK ADDRESS OF NEXT SCT	
24	18	SCTAFSMB DISK ADDRESS OF FIRST SMB FOR THIS STEP + 1 *		SCTADSMB DISK ADDRESS OF DUMMY SMB ALLOCATED	
32	20	SCTAFACT DISK ADDRESS OF FIRST ACT FOR THIS STEP		SCTVOLTB DISK ADDRESS OF VOLUME TABLE	
40	28	SCTADSTB DISK ADDRESS OF DSNAME TABLE FOR THIS STEP		SCTSCLPC NAME OF STEP THAT CALLED PROCEDURE	
48	30	STCSCLPC NAME OF STEP THAT CALLED PROCEDURE		SCTSNAME STEPNAME	
56	38	SCTSNAME STEPNAME		SCTRPACT RELATIVE POINTER TO STEP ENTRY IN ACT.	SCTVOLTL LENGTH OF VOLUME TABLE
64	40	SCTNSIOT NO. OF SIOTS IN THIS STEP	SCTNSMSG NO. OF SET UP MESSAGES	SCTNJFCB NO. OF JFCBS TO ALLOCATE	SCTSTYPE STEP TYPE
72	48	SCTMSADR ADDRESS OF REGION IN MAIN STORAGE X'00' IN 1ST BYAACA		SCTLCSAD ADDRESS OF REGION IN LCS – X'01' IN 1ST BYTE	
80	50	SCTCRWTP TTR OF FIRST WTP SMB FOR AUTOMATIC C/R USE	SCTCRCNT COUNT OF WTP SMB'S FOR STEP	RESERVED	SCTTEXT TTR FOR TIOT EXTENSION TABLE
88	58	SCTMSSZE SIZE OF REGION IN MAIN STORAGE	SCTLCSSZ SIZE OF REGION IN LCS	RESERVED	SCTSDP STEP DISPATCHING PRIORITY SET IN IEFVEA
96	60	SCTSMF STEP SYSIN COUNT FOR SMF		SCTGOTTR TTR OF PGM = *.SIOT	

SCT Storage Map (Contd.)



* - If there is no next step, this address points to a job queue record which was assigned but not written into. This field will be zeroed by job termination.

Fields in SCT — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SCTDISKA	44	2C	SCTSCLPC	88	58	SCTMSSZE
3	3	SCTTBLID	52	34	SCTSNAME	90	5A	SCTLCSSZ
4	4	SCTSSTAT	60	3C	SCTRPACT	94	5E	SCTSDP
5	5	SCTSTIME	62	3E	SCTVOLTL	96	60	SCTSMF
8	8	SCTSEXEC	64	40	SCTNSIOT	100	64	SCTGOTTR
A	A	SCTLALOC	65	41	SCTNSMSG	104	68	SCTTIOT
C	C	SCTFSIOT	66	42	SCTNJFCB	108	6C	SCTPGMNM
10	10	SCTAALOC	67	43	SCTSTYPE	116	74	SCTLDSTB
14	14	SCTANSCT	68	44	SCTXBTTR	118	76	SCTSDPCD
18	18	SCTAFSMB	72	48	SCTMSADR	120	78	SCTSDPOP
28	1C	SCTADSMB	76	4C	SCTLCSAD	121	79	SCTSDPSA
32	20	SCTAFACT	80	50	SCTCRWTP	160	A0	SCTABCND
36	24	SCTVOLTB	83	53	SCTCRCNT			
40	28	SCTADSTB	85	55	SCTTEXT			

Alphabetical list of fields in SCT

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
SCTAALOC	16	10	SCTLCSSZ	90	5A	SCTSEXEC	8	8
SCTABCND	160	A0	SCTLDSTB	116	74	SCTSMF	96	60
SCTADSMB	28	1C	SCTMSADR	72	48	SCTSNAME	52	34
SCTADSTB	40	28	SCTMSSZE	88	58	* SCTSSTAT	4	4
SCTAFACT	32	20	SCTNJFCB	66	42	SCTSTIME	5	5
SCTAFSMB	24	18	SCTNSIOT	64	40	* SCTSTYPE	67	43
SCTANSCT	20	14	SCTNSMSG	65	41	SCTTBLID	3	3
* SCTCRCNT	83	53	SCTPGMNM	108	6C	SCTTEXT	85	55
SCTCRWTP	80	50	SCTRPACT	60	3C	* SCTTIOT	104	68
SCTDISKA	0	0	SCTSCLPC	44	2C	SCTVOLTB	36	24
SCTFSIOT	12	C	* SCTSDP	94	5E	SCTVOLTL	62	3E
SCTGOTTR	100	64	SCTSDPCD	118	76	SCTXBTTR	68	44
SCTLALOC	10	A	SCTSDPOP	120	78			
SCTLCSAD	76	4C	SCTSDPSA	121	79			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SCTSSTAT	INTERNAL STEP	EAADDRBT	128	BIT0=ADDRSPC BIT BIT1=AVAILABLE FOR VS1
		SCTNORST	32	BIT2=NO RESTART TO BE DONE
		SCTNOCKP	16	BIT3=NO CHECKPOINT TO BE TAKEN
		SCTDORST	8	BIT4=DO RESTART IF NECESSARY
		SCTKEYO	4	BIT5=FOR GRAPHICS
		SCTGRPH	2	BIT6=FOR GRAPHICS
		TNCMSSTS	1	BIT7=STEP FAILED
SCTSTYPE	STEP TYPE	SCTGOSTP	128	BIT0=1 IF PGM=* (GO) STEP (FETCH DCB)
			64	BIT1=1 IF SYSIN IS SPECIFIED (DD*)
			32	BIT2=1 PARAMETER SPECIFIED THE MESSAGE CLASS
		SCTSJFHK	16	BIT3=JFCB H/K COMPLETE BITS 4, 5, AND 6 ARE USED BY THE INITIATOR 000-USE ACTION CODE 001-GO TO AVR MODULE 010-GO TO SPACE REQUEST 011-GO TO EXTERNAL ACTION SETUP 100-GO TO EXTERNAL ACTION VERIFY 110-NULL 111-NULL BIT7-RESERVED
SCTCRCNT	COUNT OF WTP SMB'S FOR STEP	SCTSNUMB	SCTCRCNT	VS1 COUNTER USED BY IEFVHH & ALLOCATION
SCTSDP	STEP PRIORITY	SCTFSTEP	32	BIT2-FIRST STEP TO BE EXECUTED
SCTTIOT	3-BYTE TTR OF STEP TIOT	SCTDSOCL	64	SET BY IEFDSOAL IF IEFDSOWR IS CALLED
		SCTMCVOL	32	ALLOC. FOR CVOL-BIT3 RESERVED
		SCTSTPLB	8	BIT4-STEPLIB PRESENT
		SCTSPSYS	4	BIT5=1 IF SPOOLED SYSIN FOR STEP
		SCTJBEND	2	JOB ENDED BIT
		SCTBCTU	1	BIAS COUNT TABLE-INFO. ON GDG TABLE
SCTABCND	8TH CONDITION CODE SLOT	SCTABCAN	16	STEPCANCEL-PRIOR ABEND-NO EVEN/ONLY
		SCTONLYC	8	STEPCANCEL-ONLY WITH NO PRIOR ABENDS
		SCTABEND	4	THIS STEP ABENDED
		SCTEVEN	2	COND=EVEN WAS SPECIFIED
		SCTONLY	1	COND=ONLY WAS SPECIFIED

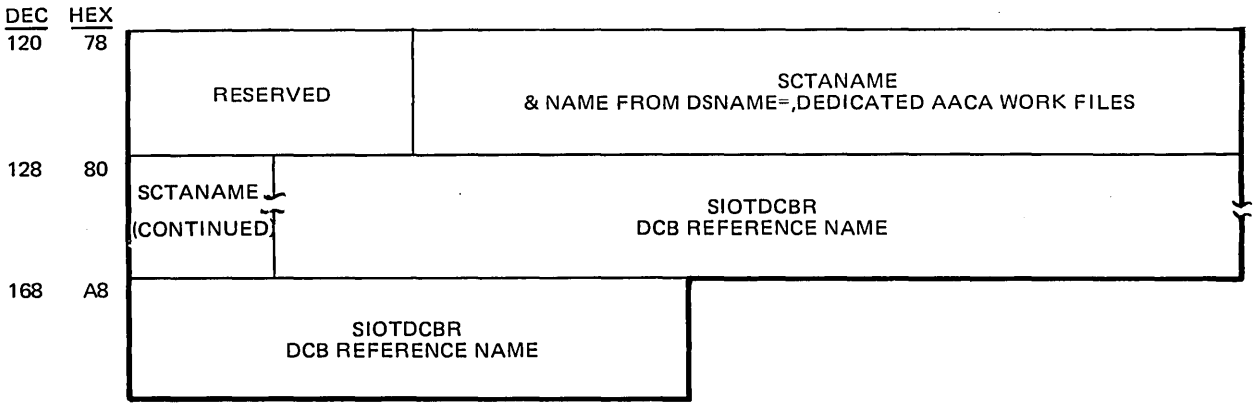
Step Input/Output Table (SIOT)

The Step Input/Output Table (SIOT) is 176 bytes long and contains information regarding the data set defined by a DD statement. The Interpreter Module (IEFVDA) builds the SIOT as part of the processing of the DD statement. It is deleted at Job Termination time.

SIOT Storage Map

DEC	HEX								
0	0	SIOTDSKA DISK ADDRESS OF SIOT			SIOTTYPE TABLE ID OF SIOT=3	SCTDDNAM DDNAME FROM THE DD CARD			
8	8	SCTDDNAM DDNAME FROM THE DD CARD				SCTCSADD INTERNAL NUMBERS FOR CHANNEL SEPARATION OR AFFINITY			
16	10	SCTCSADD INTERNAL NUMBERS FOR CHANNEL SEPARATION OR AFFINITY				SCTUSADD INTERNAL NUMBERS FOR UNIT SEPARATION OR AFFINITY			
24	18	SCTUSADD INTERNAL NUMBERS FOR UNIT SEPARATION OR AFFINITY				SCTPSIOT D. A. OF NEXT SIOT IN CHAIN			
32	20	SCTPJFCB DISK ADDRESS OF JFCB				SIOTVRSB DISK ADDRESS OF SIOT FOR VOLREF OR SUBALLOCATE			
40	28	SIOTSTOP DISK ADDRESS OF SIOT SYSTEM OUTPUT/DEPENDENCY BLOCK				RESERVED			SIOTTSTC INDICATOR FOR TIME SHARING
48	30	SCTSPOOL INTERNAL NO. OF POOL DD	SCTVOLCT NO. VOL. THAT BELONG TO DATES.	SCTVLPTR PTR TO VOLUME SERIALS OR VOLUME REFERENCE		SCTDDINO INTERNAL NO. OF DD STATEMENT	SCTNMBUT NO. UNITS FOR DATA SET	SIOTVLCT VALUE OF VOL COUNT (=JFCBVCCT)	SCTSDISP DISP OF DATA SET
56	38	SCTSBYT1 INDICATOR BYTE NO. 1	SCTSBYT2 INDICATOR BYTE NO. 2	SCTSBYT3 INDICATOR BYTE NO. 3	SCTSBYT4 INDICATOR BYTE NO. 4	SCTUTYPE SYSIN EXPECTED OF THE DATA SET—THE LOW-ORDER 2 BYTES CONTAIN UCB ADDR.			
64	40	SCTUTYPE SYSIN EXPECTED OF THE DATA SET—THE LOW-ORDER 2 BYTES CONTAIN UCB ADDR.				SCTOUTNM SYSTEM OUTPUT PROGRAM NAME			
72	48	SCTOUTNM SYSTEM OUTPUT PROGRAM NAME				SCTOUTNO FORM NO. OF CARD OR PAPER TO BE USED FOR PRINTING			
80	50	SCTOUTPN OUTPUT CLASS NAME	SCTDDDUP DD STATEMENT DUPLICATE NO.	SIOTDPCD NOT USED AT PRESENT		SIOTDPOP CREATED FOR STEP TERM. APPLICABLE ONLY IF SYSOUT BIT IS SET.			SIOTDSCT RESERVED
88	58	SIOTNDSB TTR OF NEXT DSB			SCTSBYT5 STATUS BYTE NO. 5	SIOTALTD CONDITIONAL DISPOSITION	SIOTPDQ TTR FOR THE SIOT BEING PASSED		
96	60	SIOTOUTC NO. OF COPIES TO BE PRINTED	SIOTOUTR MEMBER NAME OF TEST PATTERN FOR REG. DATA	SIOTOPUC RESERVED FOR FUTURE WRITER SUPPORT					RESERVED

SIOT Storage Map (Contd.)



Fields in SIOT — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SIOTDSKA	52	34	SCTDDINO	82	52	SIOTDPCD
3	3	SIOTTYPE	53	35	SCTNMBUT	84	54	SIOTDPOP
4	4	SCTDDNAM	54	36	SIOTVLCT	87	57	SIOTDSCT
12	C	SCTCSADD	55	37	SCTSDISP	88	58	SIOTNDSB
20	14	SCTUSADD	56	38	SCTSBYT1	91	5B	SCTSBYT5
28	1C	SCTPSIOT	57	39	SCTSBYT2	92	5C	SIOTALTD
32	20	SCTPJFCB	58	3A	SCTSBYT3	93	5D	SIOTPDQ
36	24	SIOTVRSB	59	3B	SCTBYT4	96	60	SIOTOUTC
40	28	SIOTSTOP	50	3C	SCTUTYPE	97	61	SIOTOUTR
47	2F	SIOTTSTC	68	44	SCTOUTNM	99	63	SIOTOPUC
48	30	SCTSPOOL	76	4C	SCTOUTNO	122	7A	SCTANAME
49	31	SCTVOLCT	80	50	SCTOUTPN	129	81	SIOTDCBR
50	32	SCTVLPTR	81	51	SCTDDDUP			

Alphabetical list of fields in SIOT

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
SCTANAME	122	7A	* SCTSBYT2	57	39	SIOTDSCT	87	57
*SCTBYT4	59	3B	* SCTSBYT3	58	3A	SIOTDSKA	0	0
SCTCSADD	12	C	* SCTSBYT5	91	5B	SIOTNDSB	88	58
SCTDDDUP	81	51	* SCTSDISP	55	37	SIOTOPUC	99	63
SCTDDINO	52	34	SCTSPOOL	48	30	SIOTOUTC	96	60
SCTDDNAM	4	4	SCTUSADD	20	14	SIOTOUTR	97	61
SCTNMBUT	53	35	SCTUTYPE	50	3C	SIOTPDQ	93	5D
SCTOUTNM	68	44	SCTVLPTR	50	32	SIOTSTOP	40	28
SCTOUTNO	76	4C	SCTVOLCT	49	31	* SIOTTSTC	47	2F
SCTOUTPN	80	50	* SIOTALTD	92	5C	SIOTTYPE	3	3
SCTPJFCB	32	20	SIOTDCBR	129	81	SIOTVLCT	54	36
SCTPSIOT	28	1C	SIOTDPCD	82	52	SIOTVRSB	36	24
* SCTSBYT1	56	38	SIOTDPOP	84	54			

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SIOTTSTC	INDICATORS FOR TIME SHARING AND TCAM	SIOTDYNA	128	BIT0-TSO DYNAMIC ALLOCATION-DD DYNAM PARAMETER-SET BY IEFVDA.
		SIOTTERM	64	BIT1-TSO TERMINAL BIT-DD TERM=TS PARARM. SET BY IEFVDA.
		SIOTTLMC	32	BIT2-FOR TSO USE.=1 IF LAST MESSAGE CLASS SYSOUT SIOT. SET BY IEFVMLS1. CHECKED BY IEFWCIMP, IEFYTVMS. BIT3-6 ARE NOT USED.
		SIOTCNAM	1	BIT7-FOR TCAM USE=1 IF QNAME=ON, DD STATE. SET BY IEFVDA, TESTED BY ALLOCATION.

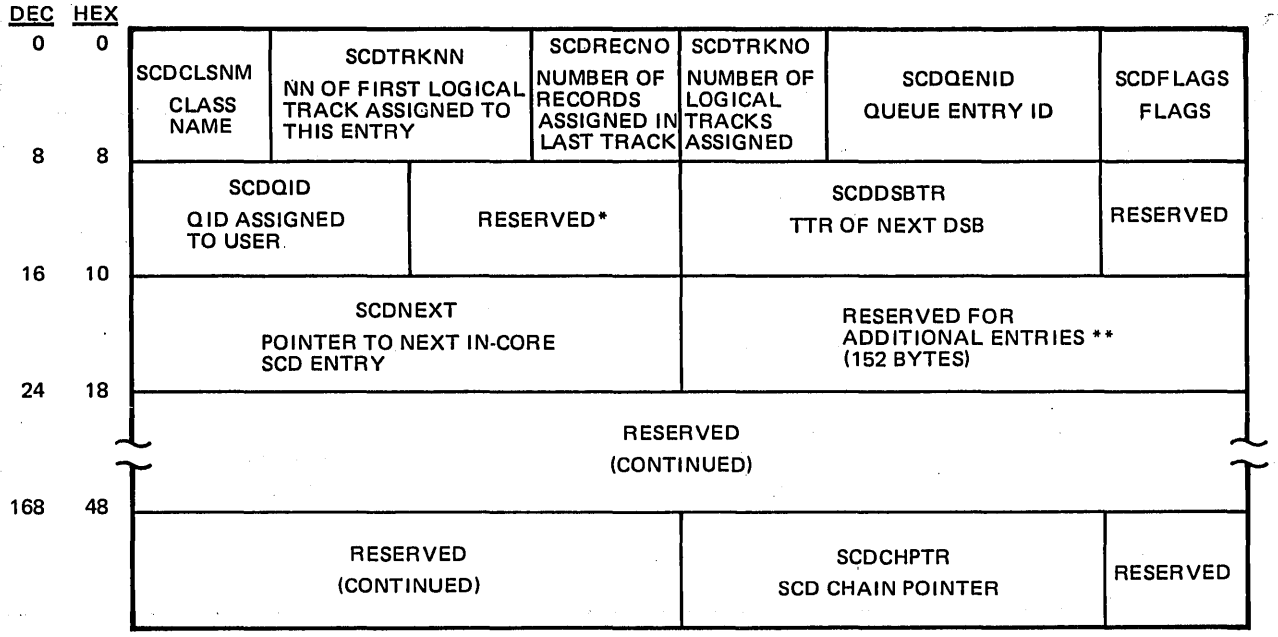
Flags and Masks (Contd.)

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SCTSDISP	SCHEDULER DISPOSITION OF THE DATA SET (AT END OF STEP OR JOB)	RETAIN	64	BIT1-REAAIN
		PRIVATE	32	BIT2-PRIVATE VOLUME
		PASS	16	BIT3-PASS THE DATA SET
		KEEP	8	BIT4-KEEP THE DATA SET
		DELETE	4	BIT5-DELETE THE DATA SET
		CATALOG	2	BIT6-CATALOG THE DATA SET
		UNCATLG	1	BIT7-UNCATALOG THE DATA SET
SCTSBYT1	INDICATOR BYTE NUMBER 1	SCTDUMMY	128	BIT0-DUMMY DATA SET
		SCTSYSIN	64	BIT1-SYSIN DATA SET
		SCTSPLTP	32	BIT2-SPLIT (PRIMARY)
		SCTSPLTS	16	BIT3-SPLIT (SECONDARY)
		SCTSBALC	8	BIT4-SUBALLOCATE-(BIT5-PARALLEL MOUNT INDICATOR)
		SCTUNAFF	2	BIT6-UNIT AFFINITY
		SCTUNSEP	1	BIT7-UNIT SEPARATION
SCTSBYT2	INDICATOR BYTE NUMBER 2	SCTCHAFF	128	BIT0-CHANNEL AFFINITY
		SCTCUSEP	64	BIT1-CHANNEL SEPARATION
		SCTVCLAF	32	BIT2-VOLUME AFFINITY
		SCTJOBLB	16	BIT3-JOBLIB DD STMNT
		SCTUNLBD	8	BIT4-UNLABELED
		SCTLABEL	4	BIT5-NONSTANDARD LABEL
		SCTDEFER	2	BIT6-DEFER MOUNTING
		SCTRECVD	1	BIT7-RECEIVED DATA SET
SCTSBYT3	INDICATOR BYTE NUMBER 3	SCTDSNRF	128	BIT0-VOLUME REFERENCE DSNAME PRESENT
		SCTSYSNE	64	BIT1-SYSIN EXPECTED (PROCEDURES ONLY)
		SCTSNVYV	32	BIT2-AWT VOLUME BLOCK INDICATOR
			16	BIT3-VOLUME REFERENCE IN STEP
		SYSOUT	8	BIT4-SYSOUT SPECIFIED
		NEW	4	BIT5-NEW DATA SET
		MOD	2	BIT6-MODIFIED DATA SET
		OLD	1	BIT7-OLD DATA SET
SCTSBYT4	INDICATOR BYTE4		128	BIT0-SET BY R/I TO INDICATE GDG SINGLE
		SIOTGDGA	64	BIT1-SIOT WAS CREATED IN RESPONSE TO A GDG-ALL REQUEST
			32	BIT2-NO PDQ PROCESSING
		SIOTASC1	16	BIT3-USASCII TAPE LABEL. SET BY IEFVDA, TESTED BY IEFWA000.
			8	BIT4-STEP PROCESSED
			4	BIT5-INTRA-STEP VOLUME AFFINITY
			2	BIT6-DATA SET IS IN PDQ
			1	BIT7-1=OLD OR MODIFIED DATA SET 0=NEW DATA SET
SCTSBYT5	STATUS BYTE NO. 5	SIOTDSO	80	BIT0=THIS SYSOUT DATA SET IS TO BE PROCESSED BY DSO. USED BY ALLOCATION, SET BY IEFVMLS1 AND TESTED IN IEFWA000.
SIOTALTD	CONDITIONAL DISPOSITION	SIOTNPRV	16	BITS0-2 ARE RESERVED. BIT3-THIS BIT IS SET AT RESTART TIME TO INDICATE THAT THIS DD IS NON-PRIVATE EVEN THOUGH IT MAY NOW APPEAR TO BE PRIVATE.
		KEEP	8	BIT4-DELETE DATA SET IS ABEND
		DELETE	4	BIT5-DELETE DATA SET IS ABEND
		CATLG	2	BIT6-CATALOG DATA SET IF ABNORMAL TERMINATION
			1	BIT7-UNCATALOG DATA SET IF ABNORMAL TERM.

SYSOUT Class Directory (SCD)

The SYSOUT Class Directory (SCD) contains 17 entries. Each entry consists of data compressed from the QMPA and used by termination to enqueue output to the proper class. The SCD is created by the interpreter, stored on SYS1.SYSJOBQE, and deleted at job termination time. The SCD occupies 176 bytes of storage.

SCD Storage Map



* The fields between locations 'A' - '14' apply only to in-core form of the SCD.
** The fields between locations '14' - 'B0' apply only to the DASD form of the SCD.

Fields in SCD — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	SCDCLSNM	7	7	SCDFLAGS
1	1	SCDTRKNN	8	8	SCDQID
3	3	SCDRECNO	12	C	SCDDSBTR
4	4	SCDTRKNO	16	10	SCDNEXT
5	5	SCDQENID	172	AC	SCDCHPTR

Alphabetical list of fields in SCD

FIELD	DEC	HEX	FIELD	DEC	HEX
SCDCHPTR	172	AC	SCDQENID	5	5
SCDCLSNM	0	0	SCDQID	8	8
SCDDSBTR	12	C	SCDRECNO	3	3
SCDFLAGS	7	7	SCDTRKNN	1	1
SCDNEXT	16	10	SCDTRKNO	4	4

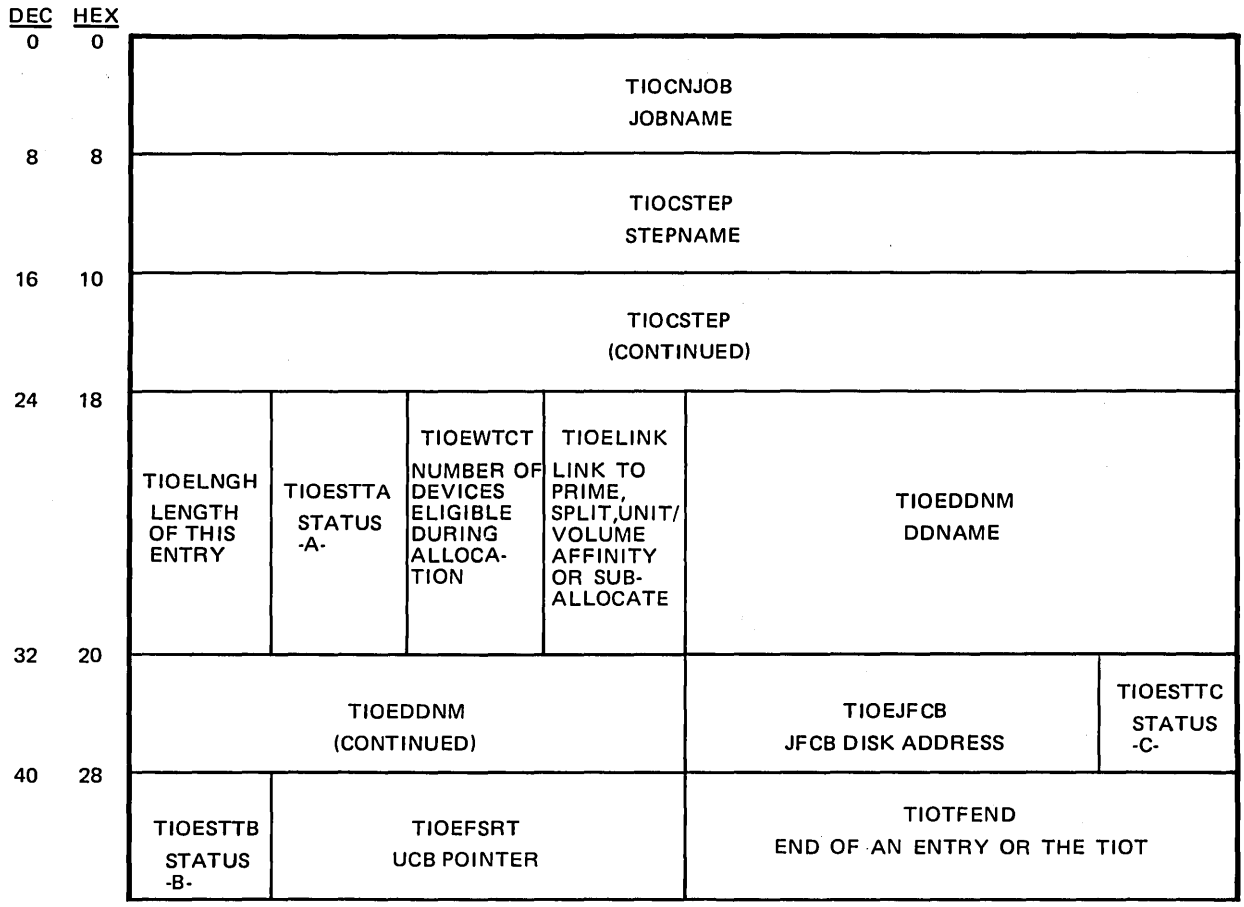
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
SCDFLAGS	FLAGS	SCDHOLD	'80'	HOLD FLAG
		SCDOUTPT	'40'	OUTPUT EXISTS FOR THIS CLASS

Task Input/Output Table (TIOT)

The Task Input/Output Table (TIOT) is built by allocation and provides the data management routines with pointers to the JFCBs and the UCBs of allocated devices. The TIOT, which has a variable length, is deleted at step termination time.

TIOT Storage Map



Fields in TIOT — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	TIOCNJOB	28	1C	TIOEDDNM
8	8	TIOCSTEP	36	24	TIOEJFCB
24	18	TIOELNGH	39	27	TIOESTTC
25	19	TIOESTTA	40	28	TIOESTTB
26	1A	TIOEWTCT	41	29	TIOEFSRT
27	1B	TIOELINK	44	2C	TIOTFEND

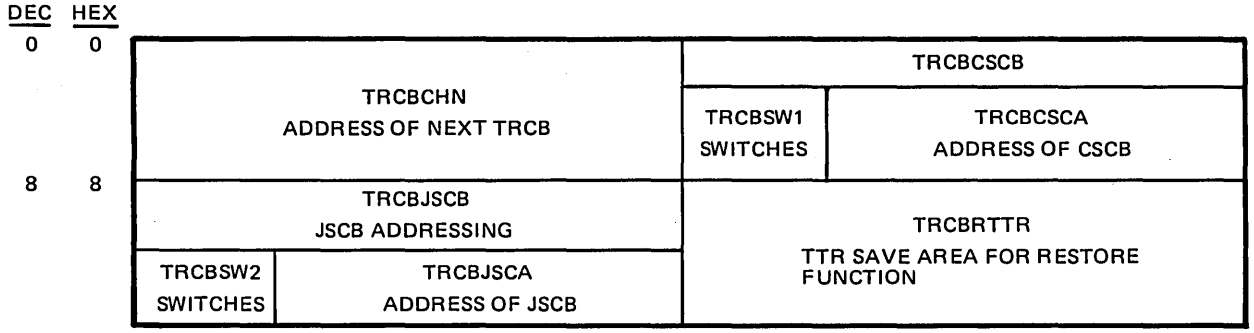
Alphabetical list of fields in TIOT

FIELD	DEC	HEX	FIELD	DEC	HEX
TIOCNJOB	0	0	TIOELNGH	24	18
TIOCSTEP	8	8	TIOESTTA	25	19
TIOEDDNM	28	1C	TIOESTTB	40	28
TIOEFSRT	41	29	TIOESTTC	39	27
TIOEJFCB	36	24	TIOEWTCT	26	1A
TIOELINK	27	1B	TIOTFEND	44	2C

Transient Routine Control Block (TRCB)

The Transient Routine Control Block (TRCB) is built whenever a START command is issued. The START command handler builds the TRCB and the initiator initializes it. A primary use of the TRCB is for task identification. The TRCB, which requires 16 bytes, is freed by the initiator when the started task terminates.

TRCB Storage Map



Fields in TRCB — by displacement

DEC	HEX	FIELD
0	0	TRCBCHN
4	4	TRCBCSCB
4	4	TRCBSW1
5	5	TRCBCSCA
8	8	TRCBJSCB
8	8	TRCBSW2
9	9	TRCBJSCA
12	C	TRCBRTTR

Alphabetical list of fields in TRCB

FIELD	DEC	HEX
TRCBCHN	0	0
TRCBCSCA	5	5
TRCBCSCB	4	4
TRCBJSCA	9	9
TRCBJSCB	8	8
TRCBRTTR	12	C
TRCBSW1	4	4
TRCBSW2	8	8

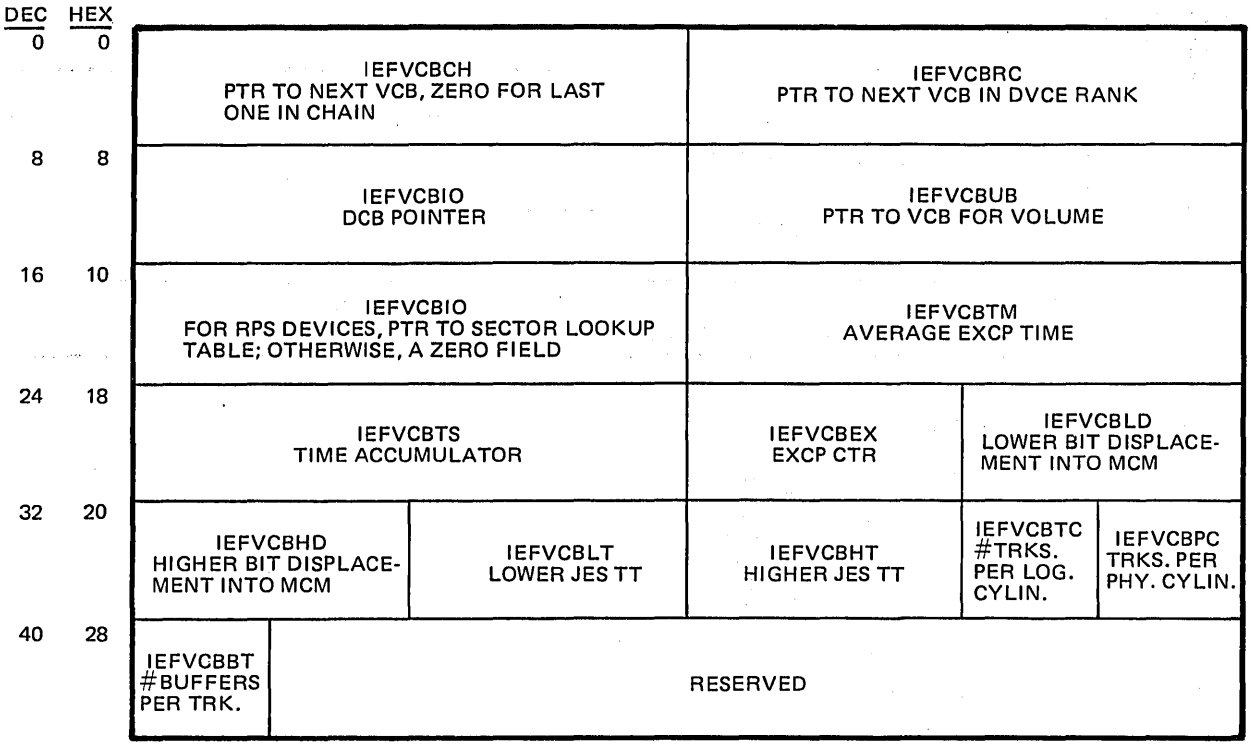
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
TRCBSW1	SWITCHES	TRCBINIT	'80'	INITIATOR
		TRCBRDR	'40'	READER/INTERPRETER
		TRCBDSO	'20'	DIRECT SYSOUT
		TRCBRSUS	'10'	READER IS RESTORED/ SUSPENDED
		TRCBSTOP	'08'	INTERNAL STOP DUE TO SCHEDULER ABEND
		TRCBMON	'04'	ON=READER STARTED OFF=WRITER STARTED
		TRCBSYS	'02'	ON MEANS READER/WRITER STARTED
		TRCBPTN	'01'	ON MEANS PARTITION SPECIFIED
TRCBSW2	SWITCHES	TRCBALL	'80'	'ALL' SPECIFIED ON START COMMAND
		TRCBRTAM	'40'	RTAM BEING STARTED
		TRCBGEN	'08'	GENERALIZED START TERMINATING
		TRCBEOF	'02'	END OF FILE CONDITION FOR ISC/OSC
		TRCBCHK	'01'	ON MEANS CSCB IN PROCESS

Volume Control Block (VCB)

The Volume Control Block (VCB) contains information related to a volume of the spool configuration. The VCBs (one for each volume) are part of the IEFJECS load module and the space occupied is fixed by WAMIN. The VCB is never deleted and occupies 30 bytes of storage.

VCB Storage Map



Fields in VCB — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFFVCBCH	20	14	IEFVCBTM	34	22	IEFVCBLT
4	4	IEFVCBRC	24	18	IEFVCBTS	36	24	IEFVCBHT
8	8	IEFVCBIC	28	1C	IEFVCBEX	38	26	IEFVCBTC
12	C	IEFVCBUB	30	1E	IEFVCBLD	39	27	IEFVCBPC
16	10	IEFVCBSC	32	20	IEFVCBHD	40	28	IEFVCBBT

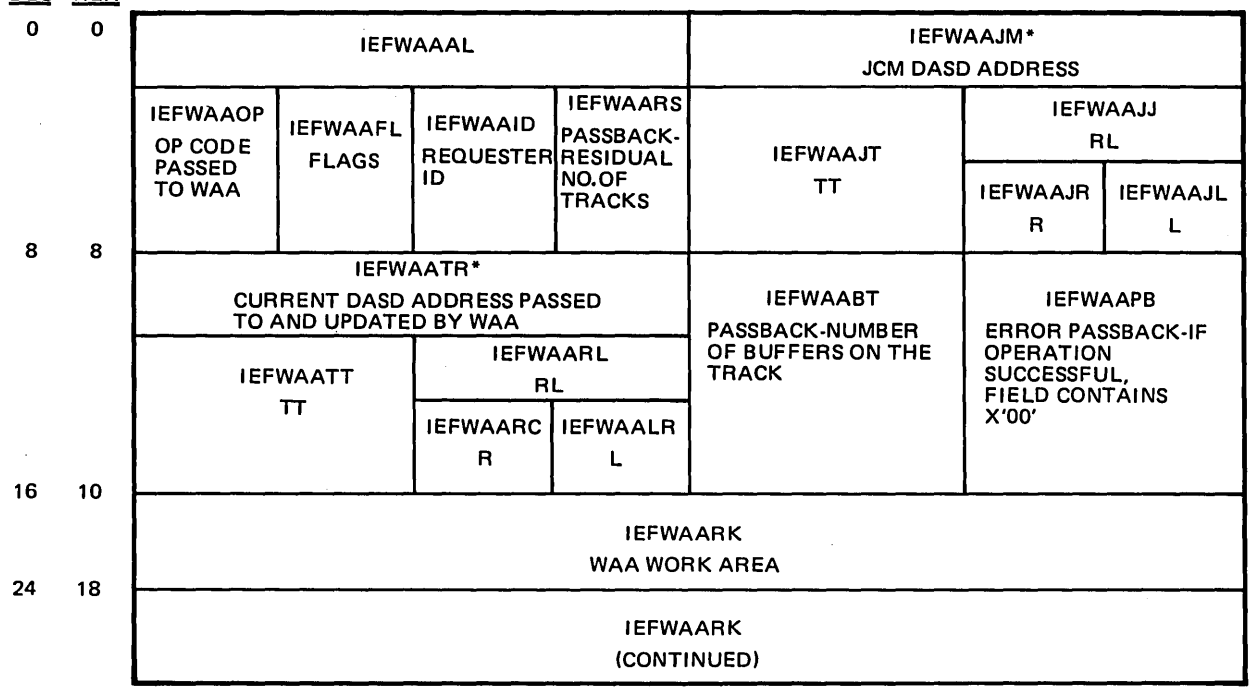
Alphabetical list of fields in VCB

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
IEFVCBBT	40	28	IEFVCBIC	8	8	IEFVCBSC	16	10
IEFVCBCH	0	0	IEFVCBLD	30	1E	IEFVCBTC	38	26
IEFVCBEX	28	1C	IEFVCBLT	34	22	IEFVCBTM	20	14
IEFVCBHD	32	20	IEFVCBPC	39	27	IEFVCBTS	24	18
IEFVCBHT	36	24	IEFVCBRC	4	4	IEFVCBUB	12	C

Work Area Allocation Parameter List (WAAL)

The Work Area Allocation Parameter List (WAAL) for an allocate or a specific allocate request is built by the requester and passed to IEFWAALC to indicate the need for a logical cylinder on the SYS1.SYSPOOL data set. The WAAL requires 32 bytes of storage.

WAAL Storage Map



* The DASD address of a record is in the form of TTRL, where:

- TT is the track address relative to the beginning of the SYS1.SYSPOOL data set.
- R is the block number on track TT.
- L is the logical record number in block R.

Fields in WAAL — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFWAAAL	7	7	IEFWAAJL
0	0	IEFWAAOP	8	8	IEFWAATR
1	1	IEFWAAFL	8	8	IEFWAATT
2	2	IEFWAAID	10	A	IEFWAARL
3	3	IEFWAARS	10	A	IEFWAARC
4	4	IEFWAAJM	11	B	IEFWAALR
4	4	IEFWAAJT	12	C	IEFWAABT
6	6	IEFWAAJJ	14	E	IEFWAAPB
6	6	IEFWAAJR	16	10	IEFWAARK

Alphabetical list of fields in WAAL

FIELD	DEC	HEX	FIELD	DEC	HEX
IEFWAAAL	0	0	IEFWAALR	11	B
IEFWAABT	12	C	IEFWAAOP	0	0
IEFWAAFL	1	1	IEFWAAPB	14	E
IEFWAAID	2	2	IEFWAARC	10	A
IEFWAAJJ	6	6	IEFWAARK	16	10
IEFWAAJL	7	7	IEFWAARL	10	A
IEFWAAJM	4	4	IEFWAARS	3	3
IEFWAAJR	6	6	IEFWAATR	8	8
IEFWAAJT	4	4	IEFWAATT	8	8

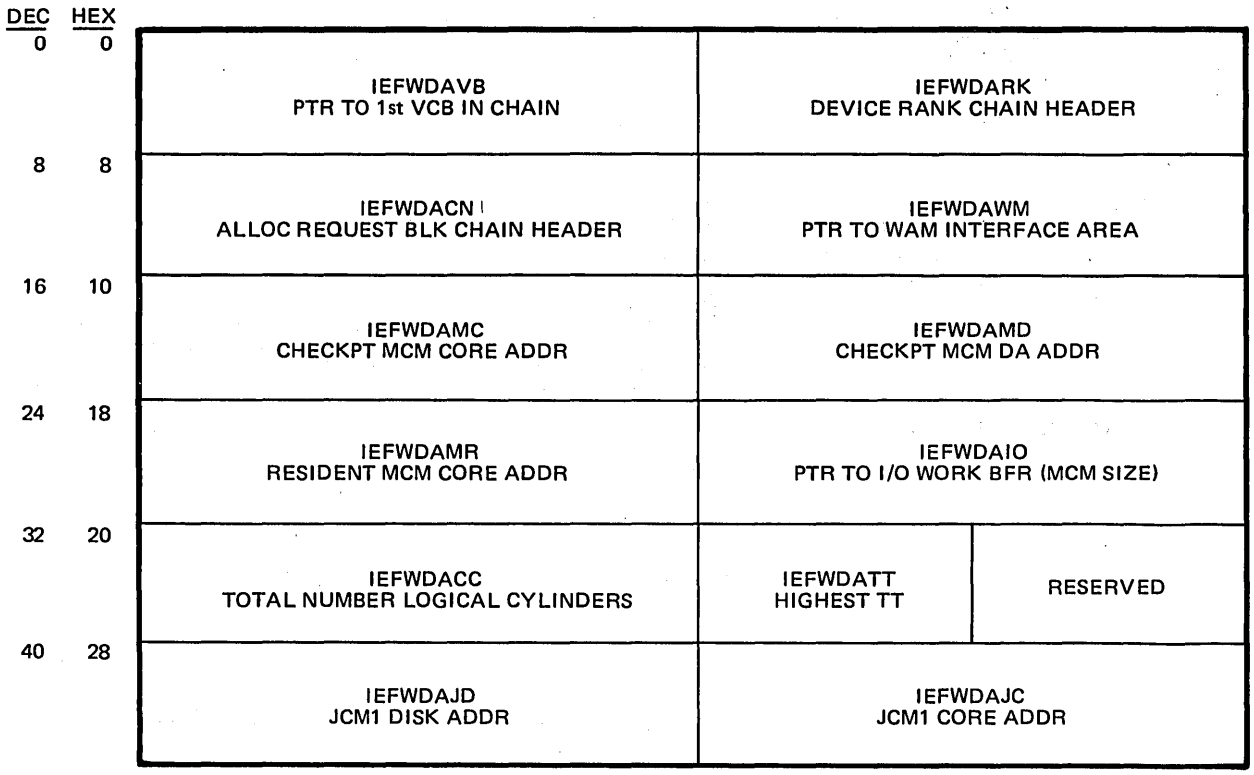
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
IEFWAAOP	OP CODE PASSED TO WAA	IEFWAAA	'01'	CODE TO ALLOCATE SPACE
IEFWAAFL	FLAGS	IEFWAAAC	'40'	SET BY WAA
		IEFWAAF2	'20'	USED INTERNALLY- CONDITIONAL ALLOCATION WHEN SPOOL CRITICAL
		IEFWAASQ	'10'	USE SQS
		IEFWAAAS	'01'	PASSED TO WAA BY REQUESTER OF SPECIFIC ALLOCATE
IEFWAAPB	ERROR PASSBACK	IEFWAABD	'80'	FUNCTION NOT PERFORMED- INVALID TTRL
		IEFWAACR	'40'	FUNCTION NOT PERFORMED- WAA GETMAIN NOT SATISFIED
		IEFWAABW	'20'	FUNCTION NOT PERFORMED- JCM CANNOT BE WRITTEN BECAUSE OF BAD TRACK
		IEFWAABR	'10'	FUNCTION NOT PERFORMED- JCM CANNOT BE READ BECAUSE OF BAD TRACK
		IEFWAANA	'08'	ON SPECIFIC ALLOCATE ONLY- LOGICAL CYLINDER NOT AVAILABLE TO JOB FOR ALLOCATION
		IEFWAABX	'04'	RESERVED
		IEFWAAX1	'02'	FUNCTION NOT PERFORMED- INVALID OP CODE
		IEFWAAX2	'01'	FUNCTION NOT PERFORMED- SPOOL VOLUME INDENTIFICATION NOT POSSIBLE
		IEFWAAX3	'80'	FUNCTION NOT PERFORMED- JCM DICTIONARY OVERFLOW
		IEFWAAX4	'40'	SPOOL CRITICAL - CONTINUE ONLY IF STARTING A WRITER
		IEFWAAX5	'20'	NO MORE SPOOL SPACE AVAILABLE
		IEFWAAX6	'10'	RESERVED
		IEFWAAX7	'08'	RESERVED
		IEFWAAX8	'04'	RESERVED
		IEFWAAX9	'02'	RESERVED
		IEFWAAX0	'01'	RESERVED

Work Area Manager Data Area (WAMDA)

The Work Area Manager Data Area (WAMDA), occupying 50 bytes of storage, contains information of the work area allocation/manager vector table and spool configuration. IEFWAMIN obtains the space at MSI time. The WAMDA is never deleted.

WAMDA Storage Map



Fields in WAMDA — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	IEFWDAUB	16	10	IEFWDAMC	32	20	IEFWDACC
4	4	IEFWDARK	20	14	IEFWDAMD	36	24	IEFWDATT
8	8	IEFWDACN	24	18	IEFWDAMR	40	28	IEFWDAJD
12	C	IEFWDAWM	28	1C	IEFWDAIO	44	2C	IEFWDAJC

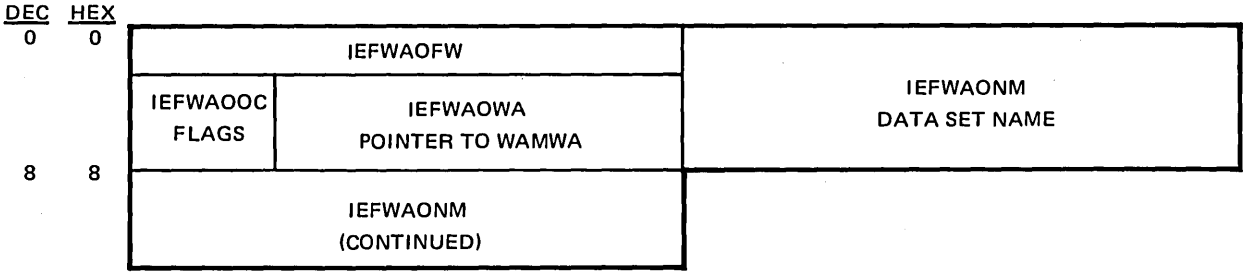
Alphabetical list of fields in WAMDA

FIELD	DEC	HEX	FIELD	DEC	HEX	FIELD	DEC	HEX
IEFWDACC	32	20	IEFWDAJD	40	28	IEFWDARK	4	4
IEFWDACN	8	8	IEFWDAMC	16	10	IEFWDATT	36	0
IEFWDAIO	28	1C	IEFWDAMD	20	14	IEFWDAWM	12	C
IEFWDAJC	44	2C	IEFWDAMR	24	18			

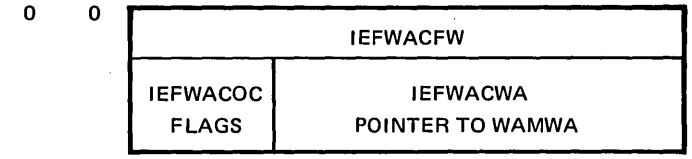
Work Area Manager Parameter List (WAMPL)

The Work Area Manager Parameter List (WAMPL) contains information requesting the work area manager to issue an open (initialize an IOB), close (nullify an IOB), read, or write to the spool data set.

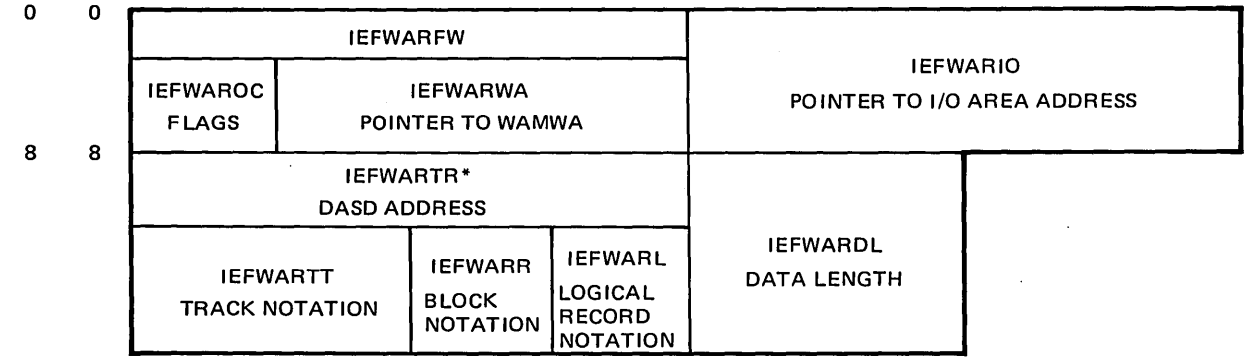
OPEN Storage Map



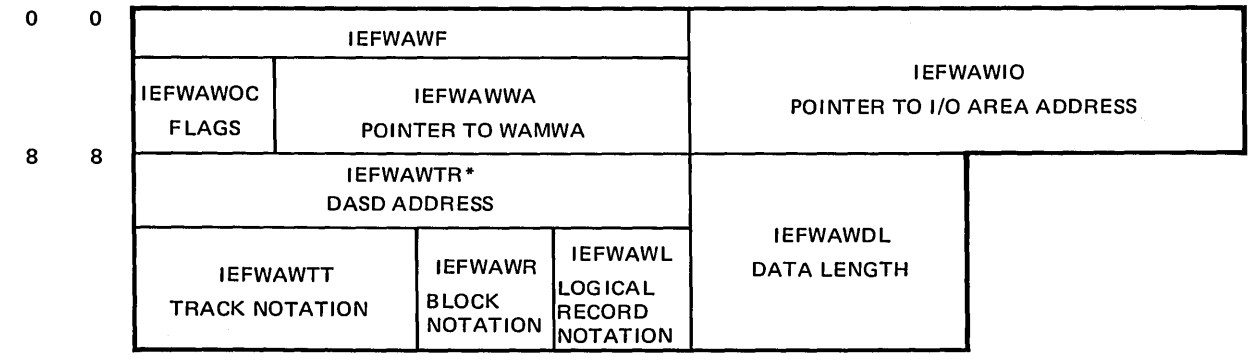
CLOSE Storage Map



READ Storage Map



WRITE Storage Map



* The DASD address of a record is in the form of TTRL where:

- TT is the track address relative to the beginning of SYS1.SYSPOOL data set.
- R is the block number on track TT.
- L is the logical record number in block R.

Fields in WAMPL — by displacement

	DEC	HEX	FIELD
OPEN	0	0	IEFWAOFW
	0	0	IEFWAOOC
	1	1	IEFWAOWA
	4	4	IEFWAONM
CLOSE	0	0	IEFWACFW
	0	0	IEFWACOC
	1	1	IEFWACWA
READ	0	0	IEFWARFW
	0	0	IEFWAROC
	1	1	IEFWARWA
	4	4	IEFWARIO
	8	8	IEFWARTR
	8	8	IEFWARTT
	10	A	IEFWARR
	11	B	IEFWARL
	12	C	IEFWARDL

	DEC	HEX	FIELD
WRITE	0	0	IEFWAWFW
	0	0	IEFWAWOC
	1	1	IEFWAWWA
	4	4	IEFWAWIO
	8	8	IEFWAWTR
	8	8	IEFWAWTT
	10	A	IEFWAWR
	11	B	IEFWAWL
	12	C	IEFWAWDL

Alphabetical list of fields in WAMPL

	FIELD	DEC	HEX
OPEN	IEFWAOFW	0	0
	IEFWAONM	4	4
	IEFWAOOC	0	0
	IEFWAOWA	1	1
CLOSE	IEFWACFW	0	0
	IEFWACOC	0	0
	IEFWACWA	1	1
READ	IEFWARDL	12	C
	IEFWARFW	0	0
	IEFWARIO	4	4
	IEFWAROC	0	0
	IEFWARL	11	B
	IEFWARR	10	A
	IEFWARTR	8	8
	IEFWARTT	8	8
	IEFWARWA	1	1

	FIELD	DEC	HEX
WRITE	IEFWAWDL	12	C
	IEFWAWFW	0	0
	IEFWAWIO	4	4
	IEFWAWOC	0	0
	IEFWAWL	11	B
	IEFWAWR	10	A
	IEFWAWTR	8	8
	IEFWAWTT	8	8
	IEFWAWWA	1	1

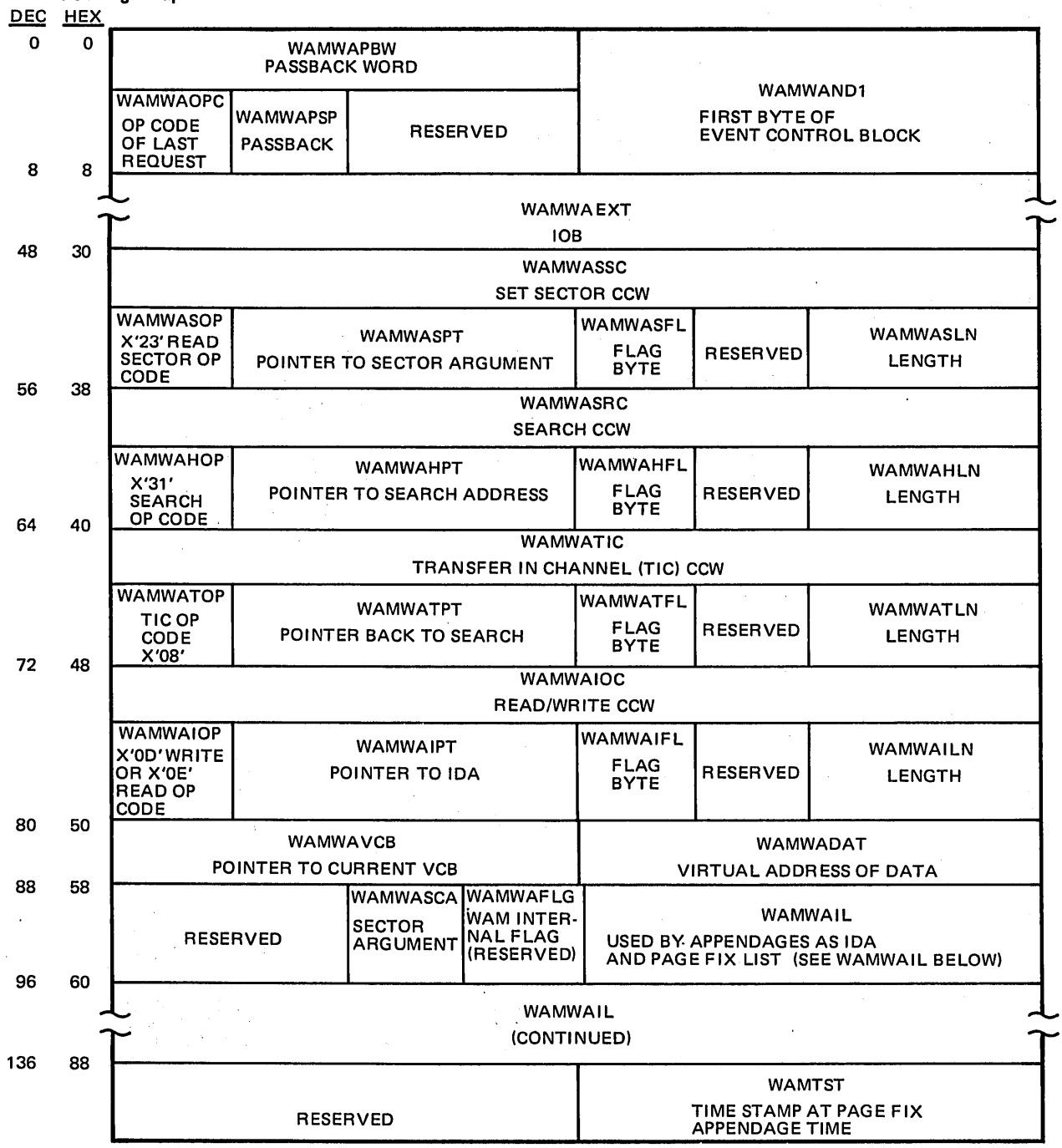
Flags and Masks

	FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
OPEN	IEFWAOOC	FLAGS	IEFWAOC	'80'	OPEN CODE MASK
			IEFWAI1	'01'	RESERVED FOR WAM INITIALIZATION
CLOSE	IEFWACOC	FLAGS	IEFWACC	'40'	CLOSE CODE MASK
			IEFWAI2	'01'	RESERVED FOR WAM INITIALIZATION
READ	IEFWAROC	FLAGS	IEFWARC	'20'	READ CODE MASK
			IEFWAI3	'01'	RESERVED FOR WAM INITIALIZATION
WRITE	IEFWAWOC	FLAGS	IEFWAWC	'10'	WRITE CODE MASK
			IEFWAI4	'01'	RESERVED FOR WAM INITIALIZATION

Work Area Manager Work Area (WAMWA)

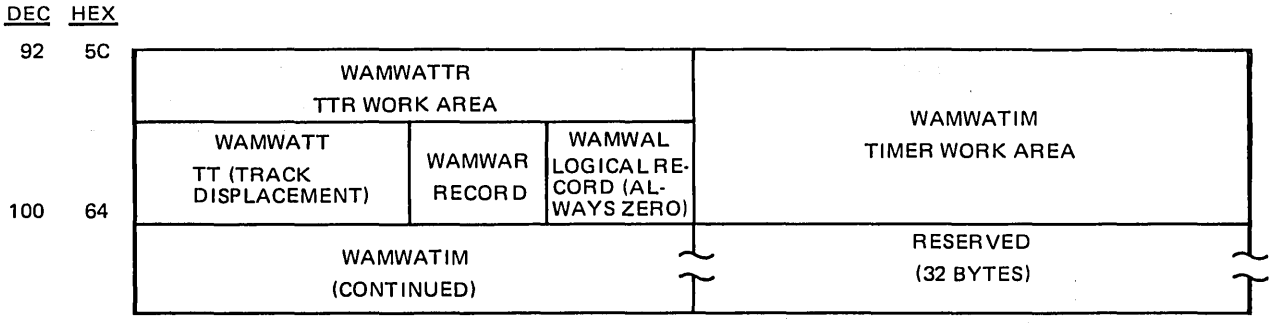
The Work Area Manager Work Area (WAMWA) contains information required by IOS and IOS appendage processing for SYS1.SYSPOOL I/O. WAMWA is initialized during an open request and modified by read/write requests to IEFWAMGR. WAMWA requires 144 bytes of storage, beginning on a doubleword boundary.

WAMWA Storage Map



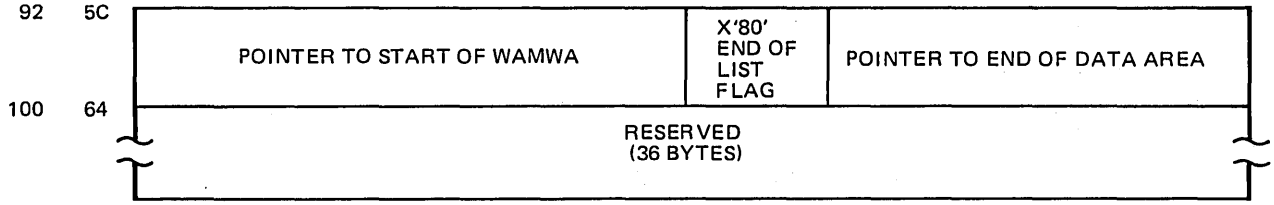
WAMWA Storage Map (Contd.)

WAMWAIL – The WAMWAIL looks like this when work area management has control and at the start of the page fix appendage.

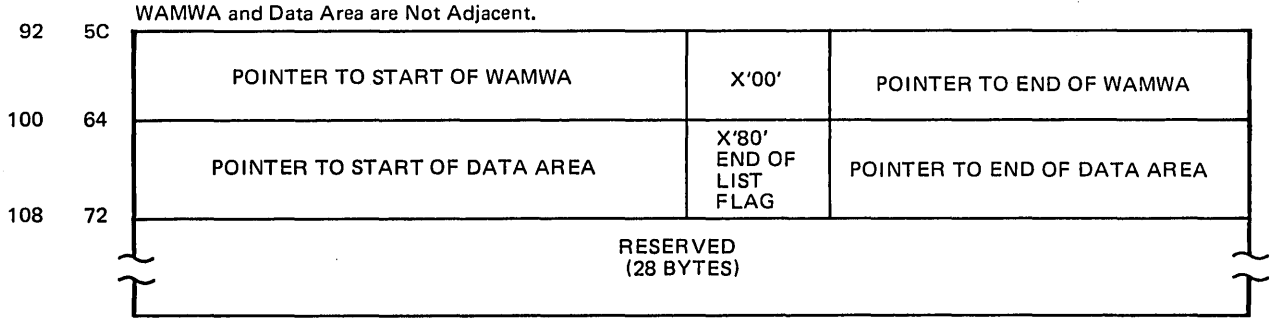


WAMWAIL – The WAMWAIL contains the fix/unfix list when it is used by the page fix/unfix appendage. The WAMWA and data area are fixed or unfixed. The list has two formats, depending on whether the WAMWA and data area are adjacent.

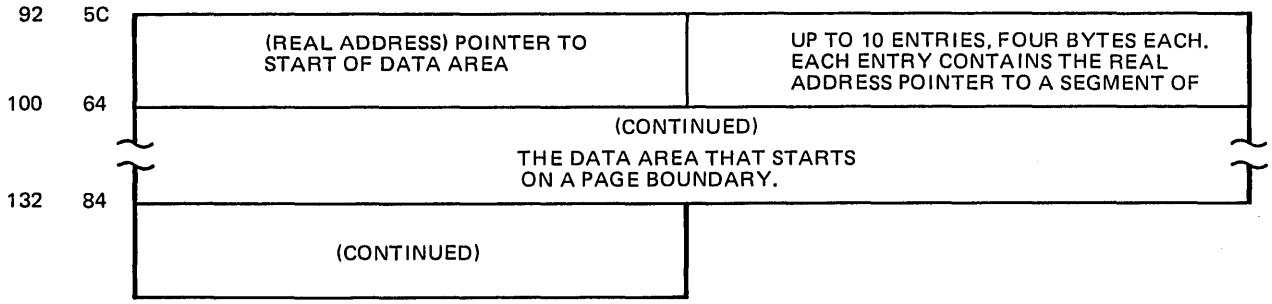
WAMWA and Data Area are Adjacent



WAMWA and Data Area are Not Adjacent.



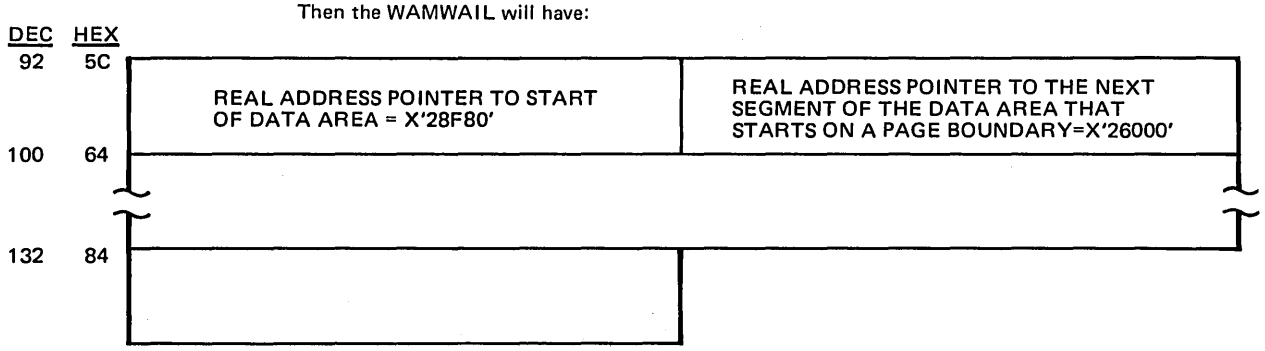
WAMWAIL – The WAMWAIL looks like this when it is used by the start I/O appendage for the indirect address list.



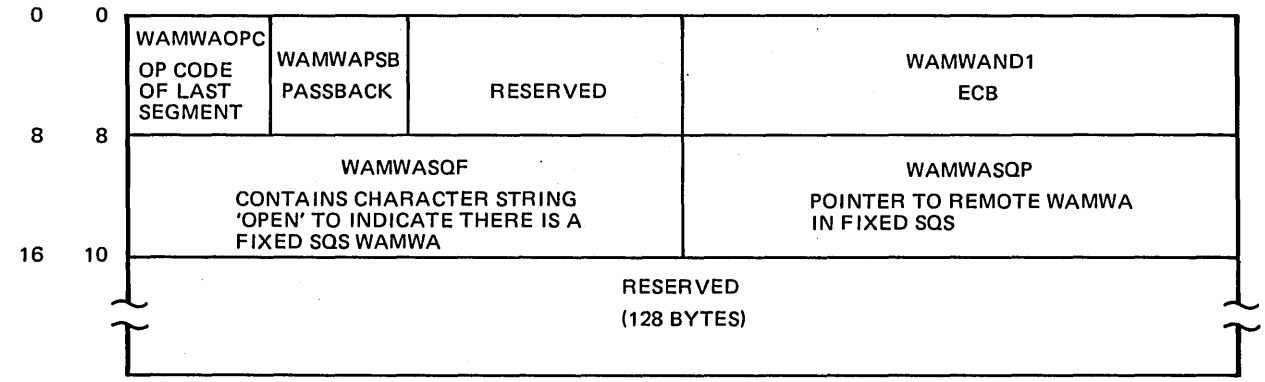
WAMWA Storage Map (Contd.)

Example:
- X'370' byte data area at X'1A2F80' to X'1A32F0'
- X'1A2F80' has a real address of X'28F80'
- X'1A3000' has a real address of X'26000'

Then the WAMWAIL will have:



WAMWA — When the CCW string within the WAMWA crosses a page boundary, a GETMAIN is issued for a remote WAMWA in fixed SQS. The initial WAMWA has this format:



Fields in WAMWA — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	WAMWAPBW	64	40	WAMWATOP
0	0	WAMWAOPC	65	41	WAMWATPT
1	1	WAMWAPSP	68	44	WAMWATFL
4	4	WAMWAND1	70	46	WAMWATLN
8	8	WAMWAEXT	72	48	WAMWAIOC
48	30	WAMWASSC	72	48	WAMWAIOP
48	30	WAMWASOP	73	49	WAMWAIPT
49	31	WAMWASPT	76	4C	WAMWAIFL
52	34	WAMWASFL	78	4E	WAMWAILN
54	36	WAMWASLN	80	50	WAMWAVCB
56	38	WAMWASRC	84	54	WAMWADAT
56	38	WAMWAHOP	90	5A	WAMWASCA
57	39	WAMWAHPT	91	5B	WAMWAFLG
60	3C	WAMWAHFL	92	5C	WAMWAIL
62	3E	WAMWAHLN	140	8C	WAMTST
64	40	WAMWATIC			

Alphabetical list of fields in WAMWA

FIELD	DEC	HEX	FIELD	DEC	HEX
WAMTST	140	8C	WAMWAPBW	0	0
WAMWADAT	84	54	WAMWAPSP	1	1
WAMWAEXT	8	8	WAMWASCA	90	5A
WAMWAFLG	91	5B	WAMWASFL	52	34
WAMWAHFL	60	3C	WAMWASLN	54	36
WAMWAHLN	62	3E	WAMWASOP	48	30
WAMWAHOP	56	38	WAMWASPT	49	31
WAMWAHPT	57	39	WAMWASRC	56	38
WAMWAIFL	76	4C	WAMWASSC	48	30
WAMWAIL	92	5C	WAMWATFL	68	44
WAMWAILN	78	4E	WAMWATIC	64	40
WAMWAIOC	72	48	WAMWATLN	70	46
WAMWAIOP	72	48	WAMWATOP	64	40
WAMWAIPT	73	49	WAMWATPT	65	41
WAMWAND1	4	4	WAMWAVCB	80	50
WAMWAOPC	0	0			

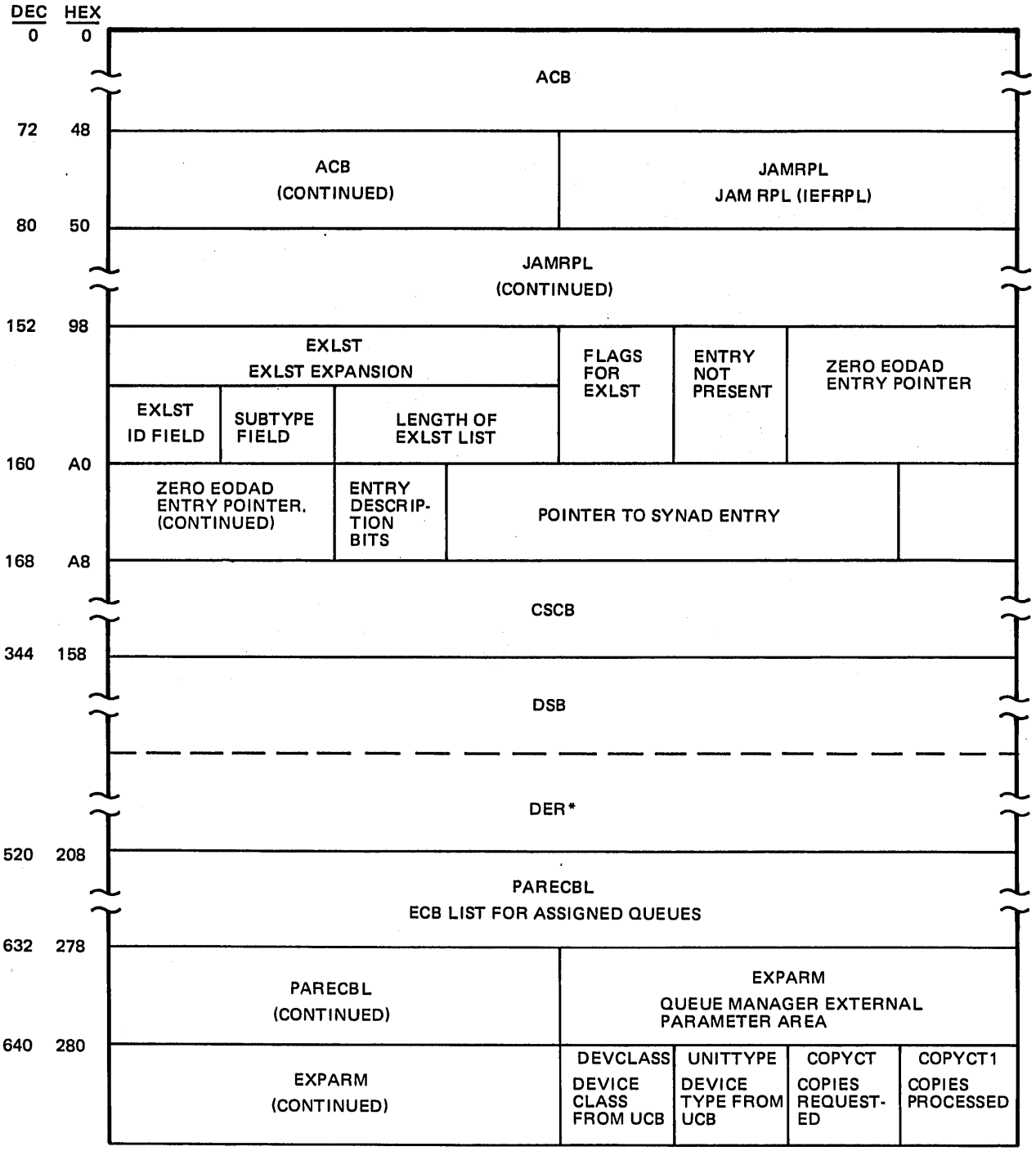
Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
WAMWAPSB	PASSBACK	WAMWASPD	'20'	SPECIFIC DELETE FLAG
WAMWAIFL	FLAG BYTE	WAMWAIDA	'04'	DATA POINTER IS TO INDIRECT ADDRESS LIST

Writer Work Area

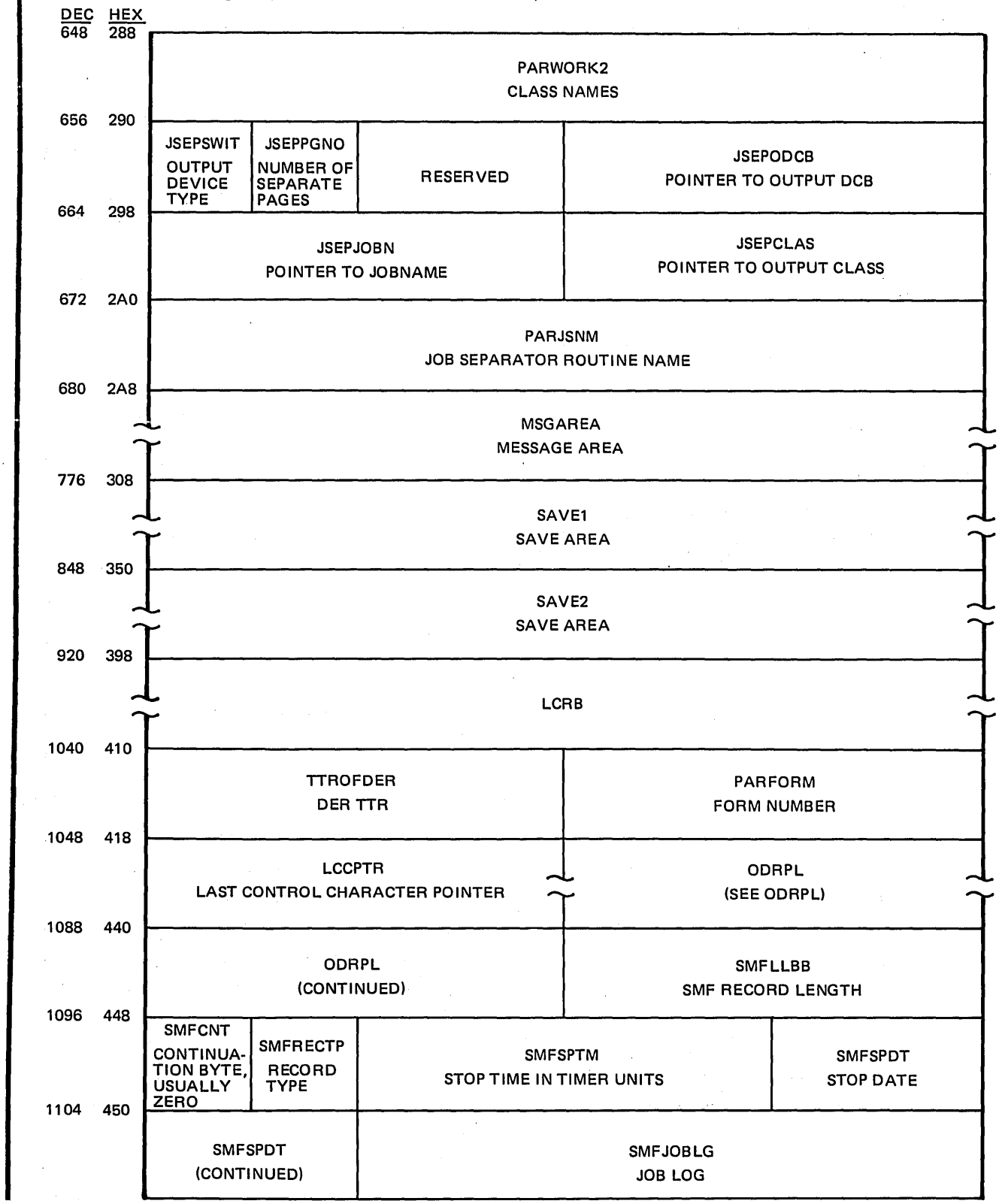
Each system writer uses the writer work area. Storage for the writer work area is reserved at master scheduler initialization time and allocated by the module IEEVMA. The work area is on a free-area list when not in use. Reader and writer work areas may be intermixed in the IEFJES partition. The work area is returned to IEFVMA's free area queue at writer termination time. This work area is always on a 2K boundary.

Writer Work Area Storage Map

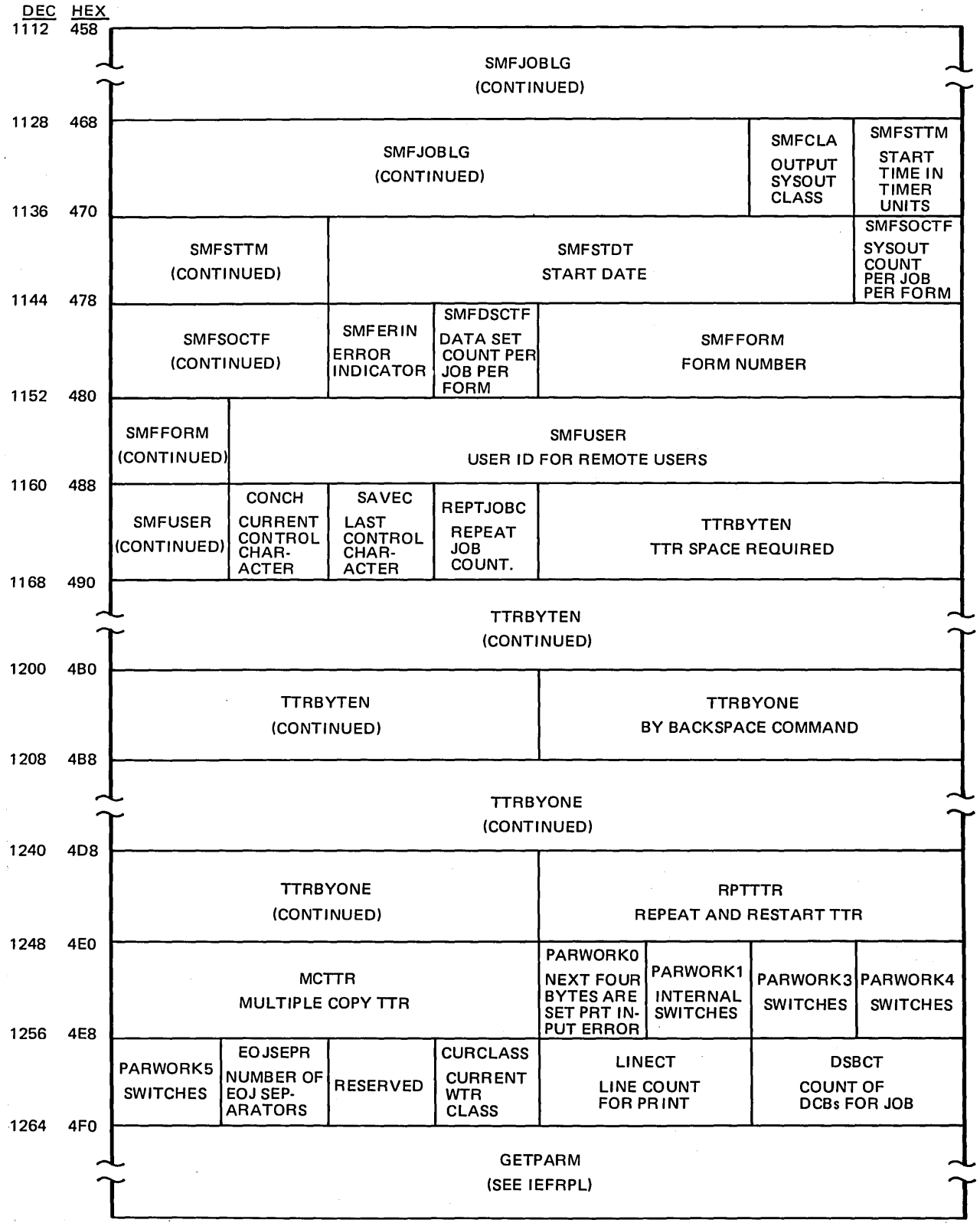


* Overlay of DSB. ORG X'158'

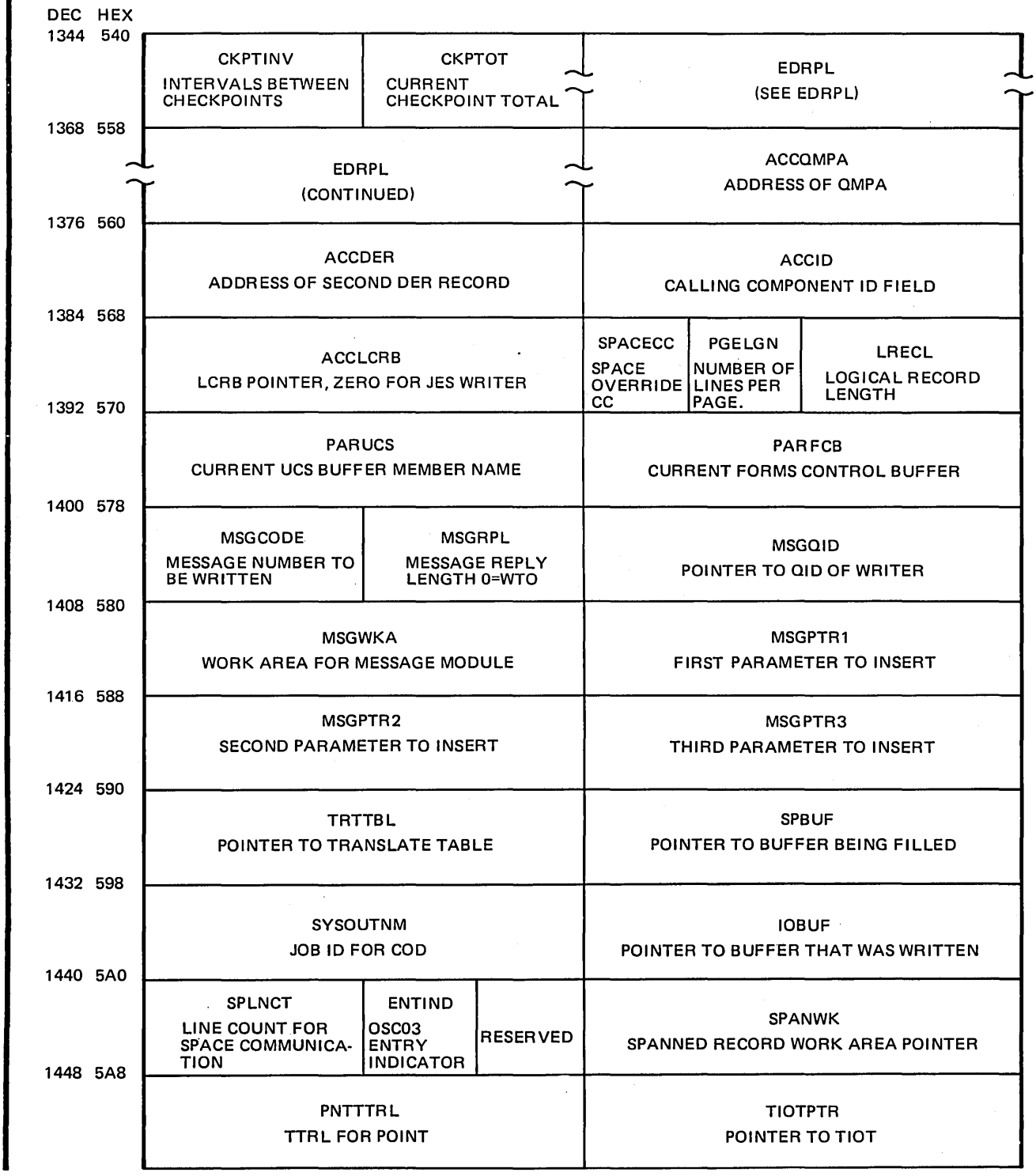
Writer Work Area Storage Map (Contd.)



Writer Work Area Storage Map (Contd.)



Writer Work Area Storage Map (Contd.)



Writer Work Area Storage Map (Contd.)

DEC	HEX		
1456	5B0	PARCOMM COMMUNICATIONS AREA POINTER	COUNT NUMBER OF OUTPUT RECORDS
1464	5B8	OSCTIME TIME WRITER STARTED JOB	WRITECT TOTAL LINES WRITTEN

Note: This work area is followed by 84 bytes of input buffers if the output device is a punch or 308 bytes if the output device is a printer.

Fields in Writer Work Area — by displacement

DEC	HEX	FIELD	DEC	HEX	FIELD
0	0	ACB	1204	4B4	TTRBYONE
76	4C	JAMRPL	1244	4DC	RPTTTR
152	98	EXLST	1248	4E0	MCTTR
168	A8	CSCB	1252	4E4	PARWORK0
344	158	DSB	1253	4E5	PARWORK1
344	158	DER	1254	4E6	PARWORK3
520	208	PARECBL	1255	4E7	PARWORK4
636	27C	EXPARM	1256	4E8	PARWORK5
644	284	DEVCLASS	1257	4E9	EOJSEPR
645	285	UNITTYPE	1259	4EB	CURCLASS
646	286	COPYCT	1260	4EC	LINECT
647	287	COPYCT1	1262	4EE	DSBCT
648	288	PARWORK2	1264	4F0	GETPARM
656	290	JSEPSWIT	1344	540	CKPTINV
657	291	JSEPPGNO	1346	542	CKPTOT
660	294	JSEPODCB	1348	544	EDRPL
664	298	JESPJOBN	1372	55C	ACCQMPA
668	29C	JSEPCLAS	1376	560	ACCDER
672	2A0	PARJSNM	1380	564	ACCID
680	2A8	MSGAREA	1384	568	ACCLCRB
776	308	SAVE1	1388	56C	SPACECC
848	350	SAVE2	1389	56D	PGELGN
920	398	LCRB	1390	56E	LRECL
1040	410	TTROFDER	1392	570	PARUCS
1044	414	PARFORM	1396	574	PARFCB
1048	418	LCCPTR	1400	578	MSGCODE
1052	41C	ODRPL	1402	57A	MSGRPL
1092	444	SMFLLBB	1404	57C	MSGQID
1096	448	SMFCNT	1408	580	MSGWKA
1097	449	SMFRECTP	1412	584	MSGPTR1
1098	44A	SMFSPTM	1416	588	MSGPTR2
1102	44E	SMFSPDT	1420	58C	MSGPTR3
1106	452	SMFJOBLG	1424	590	TRTTBL
1134	46E	SMFCLA	1428	594	SPBUF
1135	46F	SMFSTTM	1432	598	SYSOUTNM
1139	473	SMFSTDT	1436	59C	IOBUF
1143	477	SMFSOCTF	1440	5A0	SPLNCT
1147	47B	SMFERIN	1442	5A2	ENTIND
1148	47C	SMFDSCTF	1444	5A4	SPANWK
1149	47D	SMFFORM	1448	5A8	PNTTTRL
1153	481	SMFUSER	1452	5AC	TIOTPTR
1161	489	CONCH	1456	5B0	PARCOMM
1162	48A	SAVEC	1460	5B4	COUNT
1163	48B	REPTJOBC	1464	5B8	OSCTIME
1164	48C	TTRBYTEN	1468	5BC	WRITECT

Alphabetical list of fields in Writer Work Area

FIELD	DEC	HEX	FIELD	DEC	HEX
ACB	0	0	PARCOMM	1456	5B0
ACCDER	1376	560	PARECBL	520	208
ACCID	1380	564	PARFCB	1396	574
ACCLCRB	1384	568	PARFORM	1044	414
ACCQMPA	1372	55C	PARJSNM	672	2A0
CKPTINV	1344	540	PARUCS	1392	570
CKPTOT	1346	542	PARWORK0	1252	4E4
CONCH	1161	489	PARWORK1	1253	4E5
COPYCT	646	286	PARWORK2	648	288
COPYCT1	647	287	PARWORK3	1254	4E6
COUNT	1460	5B4	PARWORK4	1255	4E7
CSCB	168	A8	PARWORK5	1256	4E8
CURCLASS	1259	4EB	PGELGN	1389	56D
DER	344	158	PNTTTRL	1448	5A8
DEVCLASS	644	284	REPTJOBC	1163	48B
DSB	344	158	RPTTTR	1244	4DC
DSBCT	1262	4EE	SAVEC	1162	48A
EDRPL	1348	544	SAVE1	776	308
ENTIND	1442	5A2	SAVE2	848	350
EOJSEPR	1257	4E9	SMFCLA	1134	46E
EXLST	152	98	SMFCNT	1096	448
EXPARM	636	27C	SMFDSCTF	1148	47C
GETPARM	1264	4F0	SMFERIN	1147	47B
IOBUF	1436	59C	SMFFORM	1149	47D
JAMRPL	76	4C	SMFJOBLG	1106	452
JSEPCLAS	668	29C	SMFLLBB	1092	444
JSEPJOBN	664	298	SMFRECTP	1097	449
JSEPODCB	660	294	SMFSOCTF	1143	477
JSEPPGNO	657	291	SMFSPDT	1102	44E
JSEPSWIT	656	290	SMFSPTM	1098	44A
LCCPTR	1048	418	SMFSTDT	1139	473
LINECT	1260	4EC	SMFSTTM	1135	46F
LRCB	920	398	SMFUSER	1153	481
LRECL	1390	56E	SPACECC	1388	56C
MCTTR	1248	4E0	SPANWK	1444	5A4
MSGAREA	680	2A8	SPBUF	1428	594
MSGCODE	1400	578	SPLNCT	1440	5A0
MSGPTR1	1412	584	SYSOUTNM	1432	598
MSGPTR2	1416	588	TIOTPTR	1452	5AC
MSGPTR3	1420	58C	TRTTBL	1424	590
MSGQID	1404	57C	TTRBYONE	1204	4B4
MSGRPL	1402	57A	TTRBYTEN	1164	48C
MSGWKA	1408	580	TTROFDER	1040	410
ODRPL	1052	41C	UNITTYPE	645	285
OSCTIME	1464	5B8	WRITECT	1468	5BC

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
ACBMACR1	MACRF FIRST BYTE	ACBKEY	'80'	ACCESS DATA VIA IX
		ACBADR	'40'	ACCESS WITHOUT IX
		ACBADD	'40'	SAME AS ABOVE
		ACBCNV	'20'	CONTROL INTERVAL PROC.
		ACBBLK	'20'	SAME AS ABOVE
		ACBSEQ	'10'	SEQUENTIAL PROC.
		ACBDIR	'08'	DIRECT PROCESSING
		ACBIN	'04'	GET
		ACBOUT	'02'	PUT
		ACBUPD	'01'	UPDATE
ACBMACR2	MACRF SECOND BYTE	ACBCBOPN	'80'	COL BIN ALLOWED
		ACBCBRD	'40'	COL BIN READ
		ACBCHN	'20'	CHAINED SCHEDULING
		ACBSKP	'10'	SKIP SEQ ACCESSING
		ACBHLD	'08'	TRACK HOLD OPTION
ACBRECFM	RECORD FORMAT	ACBRECAF	'80'	JES FORMAT
ACBCCTYP	CONTROL CHARAC-TER	ACBCCASA	'04'	ASA CHARACTERS
		ACBCCMCH	'02'	MACHINE CHARACTERS
ACBOPT	NON-USER OPTIONS	ACBCRNCK	'80'	NO CHECK FOR MODIFY
		ACBCRNRE	'40'	NO DATA ERASE/REPOS
		ACBDSORGA	'08'	JES ACB IN DSORG+1
ACBOFLGS	OPEN/CLOSE FLAGS	ACBEOV	'20'	EOV DETECTS COMPLETED
		ACBOPEN	'10'	ACB IS OPEN
		ACBDSERR	'08'	NO FURTHER REQUESTS
		ACBEXFG	'02'	USER EXIT FLAG
		ACBIOSFG	'01'	PROCESS BY IOS
ACBINFLG	INDICATOR FLAGS	ACBJEPS	'40'	JEPS PROCESSING
		ACBIJRQE	'20'	RQE OBTAINED
CHSTS	STATUS FLAGS	CHAP	'80'	ASSIGNMENT PENDING
		CHSYS	'40'	SYSTEM TASK CSCB
		CHSOUT	'20'	CANCEL ALL SYSOUT
		CHQSPC	'10'	INSUFFICIENT QSPACE CAUSING ABEND 422
		CHAD	'08'	ADD THIS CSCB TO CHAIN
		CHDL	'04'	DELETE THIS CSCB FROM CHAIN
		CHFC	'02'	FREE THIS CSCBS CORE
		CHABTERM	'01'	EXECUTE BRANCH ENTRY TO ABTERM
CHACT	FLAGS INDICATING ACTIVITY INVOLVED	CHSWAP	'80'	SWAPPLE JOB
		CHTERM	'40'	TERMINAL JOB
		CHDISC	'20'	CANCEL IMPLIES DISCONNECT
		CHDSI	'10'	ON MEANS NO DATA SET INTEGRITY
		CHCL	'08'	CANCELABLE JOB STEP
		CHCLD	'04'	CANCEL COMMUNICATION SWITCH
		CHAIFX	'02'	CANCELABLE - MFTII ONLY
		CHIFY	'01'	SYSTEM ASSIGNED PROCEDURE MFTII
CHACT1	FLAG BYTE	CHRDWTR	'80'	COMMAND WAS START RDR OR WTR
CHSWT	COMMUNICATION SWITCHES	CHJCT	'40'	READER RETURN WITH IN-CORE JCT
		CHPSD	'20'	WRITER PAUSE DATA SET
		CHPSF	'10'	WRITER PAUSE FORMS
		CHAC	'08'	ID SPECIFIED ON S COMMAND
CHTYPE	FLAGS	CHDSTAT	'80'	STATUS DISPLAY (SVC 104) CMD
		CHHIAR	'02'	ON MEANS H1 SPECIFIED ON COMMAND (ICB337)
		CHDEF	'01'	ON MEANS DEFAULT TO H0 (ICB337)
CHCSYSO	EXPRESS CANCEL SYSOUT	CHALL	'80'	ALL SPECIFIED
		CHINN	'40'	IN SPECIFIED
		CHOUT	'20'	OUT SPECIFIED
		CHHOLD	'10'	HOLD Q SPECIFIED
		CHQUE	'08'	SPECIFIC QUEUE
		CHDUMP	'04'	DUMP SPECIFIED
		CHJB	'02'	END SCAN SWITCH
		CHUSERID	'01'	INDICATES 'USER=' SPECIFIED ON CANCEL COMMAND
DSBFLAG	STATUS BYTE	DSBSYSO	'80'	SYSOUT DATA SET
		DSBEMPTY	'40'	EMPTY SYSOUT DATA SET
		DSBHOLDQ	'20'	TERM HOLD Q IND

Flags and Masks

FLAG FIELD	CONTAINS	MASK NAME	VALUE	MEANS
DSBPAGE	DSB INDICATORS	DSBPROC	'01'	WTR HAS PROCESSED DSB
		DSBCKPT	'02'	WTR HAS CKPOINTED DATA SET
		DSBHDIND	'04'	HOLD COMMAND ISSUED FOR DS
DERLOG	LOG STATUS BYTE	DERLOGID	'80'	DER FOR LOG DATA SETS
		DERLOGXO	'40'	LOG X OPENED
		DERLOGXE	'20'	LOG X ENQUEUED
		DERLOGYO	'10'	LOG Y OPENED
		DERLOGYE	'08'	LOG Y ENQUEUED
DERFLAGS	DER FLAGS - BYTE 1	DERJOBRS	'80'	JOB TO BE RESTARTED
		DERWMST	'40'	WARM START CONDITION
		DERJOBF	'20'	JOB FAIL BIT
		DERINTRP	'10'	INTERPRETATION COMPLETE
		DERTERM	'08'	TERMINATION COMPLETE
		DERGSTRT	'04'	GENERALIZED START IDENT.
		DERPROCS	'02'	PROCEDURE SPOOLED INDICATOR
		DERTMRT	'01'	TERMINATION HAS PERFORMED ROUTE
DERFLAG1	DER FLAGS- BYTE 2 BITS 2-7 UNUSED	DERDSO	'80'	JOB SELECTED BY DSO WRITER
		DERSYSTK	'40'	JOB IS A SYSTEM TASK
DEVCLASS	DEVICE CLASS FROM UCB	UNREC	'08'	UNIT RECORD
		MAGTAPE	'80'	MAGNETIC TAPE
UNITTYPE	DEVICE TYPE FROM UCB	MT2400	'01'	2400 MAG TAPE.
		CR2540	'02'	2540 CARD PUNCH
		CR1442	'03'	1442 CARD READ PUNCH
		CR2520	'05'	2520 CARD READ PUNCH
		PR1403	'08'	1403 & 1404 PRINTER
		PR3211	'09'	3211 PRINTER
		PR1443	'0A'	1443 PRINTER
ODRFLAG	FLAGS	ODRSOUT	'40'	SYSOUT DATA SET
		ODROUT	'20'	OUTPUT DATA SET
		ODRINPT	'10'	INPUT DATA SET
		ODRSQA	'08'	GET DSD FROM SQA
		ODRWNEW	'04'	WRITE TO NEW DATA SET
		ODRWOLD	'02'	WRITE TO EXISTING DATA SET
		ODRREAD	'01'	READ FROM EXISTING DATA SET
PARWORK0		SETPIERR	'80'	SETPRT INPUT ERROR
		SETPOERR	'40'	SETPRT ERROR CLOSE WRITER
		LOADDEF	'20'	LOAD DEFAULT IMAGES
		FCBTST	'10'	FCB TEST IN PROGRESS
		OUTERROR	'08'	OUTPUT ERROR OCCURRED FOR JOB
		INERROR	'04'	INPUT ERROR OCCURRED FOR JOB
		BUFSW	'02'	FIRST TIME BUFFER SWITCH
		NOODS	'01'	ODS WAS ISSUED IN OSC03
PARWORK4		SEPCOMP	'01'	INDICATES SEPARATORS HAVE BEEN WRITTEN
		SMFREQD	'02'	INDICATES SMF RECORDS REQUIRED
		SMFINIT	'04'	INDICATES SMF REC. HAS BEEN INIT
		DSETPAUS	'08'	INDICATES DATASET PAUSE
		FIRSTIO	'10'	INDICATES FIRST I/O THIS DSB
		SPACE	'20'	SPACE COMMAND
		EBUF	'40'	INDICATES PUT RTN SHOULD WRITE SHORT BUFFER
		MCOPMSG	'80'	INDICATES MULTIPLE COPY MSG
PARWORK5	SWITCHES	RMTDEV	'80'	BIT INDICATING A REMOTE DEVICE
		CKPTREQ	'40'	USER-REQUESTED CHECKPOINT
		EDSNOTDN	'20'	AN ODS HAS BEEN ISSUED
		ANYODS	'04'	AT LEAST ONE ODS
		BLKNONZO	'01'	BLK SPEC IN PROC
EDRFLG	FLAGS	EDRFPUT	'80'	FORCED PUT
PARCOMM	COMMUNICATIONS AREA PTR	COMMECB	'00'	DISP OF COMMAND ECB
		COMMCIB	'04'	DISP OF COMMAND INPUT BUFFER

Section 6: Diagnostic Aids

CONTENTS

This section contains aids for debugging job management-related problems. Additional aids can be found in the *OS/VS1 Debugging Guide* listed in the *Preface.*

The pages on *Diagnostic Techniques* give a brief description of possible trouble spots in the job management elements and indicate where patch areas can be located.

The charts titled *Register Usage* list the job management modules alphabetically. All registers appear next to each module except work registers, which are registers used for more than one purpose in a module.

The *Message/Module Relationship* charts cross-reference the operator messages with the module where they originated, not the module that prints them out, and with the program organization chart(s) on which they appear.

Return Codes charts list modules alphabetically with the return codes and their meanings. The return codes are located in register 15 unless otherwise indicated. The codes do not represent system codes (see the *System Codes* manual listed in the *Preface*) or codes relating to message modules. Numbers enclosed within apostrophes are hexadecimal numbers; numbers without apostrophes are decimal numbers.

Control block pointer charts give the user a quick reference to the pointer fields of some of the job management data areas.

Diagnostic Procedures

The modifications discussed in this section are for diagnostic purposes only. You *must* remove them as soon the desired information is obtained. These modifications constitute alterations to modules with System Control Programming (SCP) support, and, if not promptly removed, the altered code may result in error conditions for which IBM is not responsible.

Patch Areas

The JES Reader and JES Writer have one patch area in every CSECT. Each area is a DC of the CSECT name repeated such that the reserved area comprises about 5% of the CSECT.

System Restart has a 100-byte patch area in module IEFSD305. The DC defining this area is labeled *ZAP.*

Throughout the rest of the scheduler, if a module does contain a patch area, the DC defining area is labeled *ZAPAREA*. Each byte in the area is initialized to X'FF'.

All patch areas in the Communications Task have the label PATCH1, are DCs, and are 100 bytes long. The patch areas available in the Communications Task are:

Module Name	SVCLIB Name
IEECLCTX	IGCXLO7B
IEECMCTR	IGC0007B
IEECMCTX	IGCXMO7B
IEECMOCP	IGC2I07B
IEECMPMC	IGC1107B
IEECMPMP	IGC2107B
IEECMPMX	IGC0107B
IEECMPM1	IGC0207B
IEECMWTL	IGC0907B
IEECNCTX	IGCXN07B
IEECOCTX	IGCXD07B
IEECVML3	IGC0603E
IEECVML5	IGC0703E
IEECVML6	IGC0803E
IEECVML7	IGC0903E
IEECVOCC	IGC1IQ7B
IEECVOCX	IGC0I07B
IEECXDOM	IGC0008G
IEEMFWTO	IGC0003E
IEEVROUT	IGC0503E
IEEVWTOR	IGC0103E
IEE1A03D	IGC1203D
IEE1B03D	IGC1303D

All other transients are sufficiently below their maximum size design point to allow for patches. However, DCs or DSs have not been coded into the modules to reserve areas where patches could be made.

Module Entry Trap

A one-instruction loop at the entry point to a particular module or at the entry point to a specific routine allows the user to stop when that module or routine receives control. This loop is especially helpful when the module operates in a transient area or when the entry point cannot be determined easily. An example of how this may be done is:

```
NAME    IEFDSDRP  IEFDSDRP
VERIFY  0004      05C0,47F0,C010
REP     0004      05C0,47F0,C002
```

To apply this trap, you must use the microfiche listing to determine the addresses, displacements,

and instructions to be used. If it is not necessary to obtain a system printout at the trap, the original instruction can replace the trap via the console, and processing can continue.

Note: Several of these traps can be applied throughout a module if the user is not certain which direction a given process may take.

Spool I/O Error Trap

You can set the spool I/O error traps by altering the work area manager abnormal end appendage to obtain a program check. You can do this by using the HMASPZAP (superzap) utility program or through the console.

The superzap program provides a permanent alter as in this example:

```
NAME   IEFJECS  IEFWAMGR
VERIFY X900F
REP    X0000
```

where X is the byte displacement of the label IEFWMAE in the module IEFWAMGR.

Using alter virtual through the console provides a temporary alter as in this example:

You can complete alter virtual after IEFJECS is loaded by master scheduler initialization.

AV X

where X is found as follows:

JESCT+X'24'	(pointer to WAMDA)
WAMDA+X'00'	(pointer to a VCB)
VCB+X'08'	(pointer to a DCB)
DCB+X'2C'	(pointer to a DEB)
DEB+X'1C'	(pointer to a AVT)
AVT+X'10'	(pointer to the abnormal end appendage

Work Area Reference Trap

This trap enables the user to verify data in one of the Scheduler's work areas.

With the module entry trap, the system can stop processing at a key point within a specific module to obtain the address of the component's work area (see *Register Usage*). The address of the specific location to be referenced (virtual address) is set on the MAIN storage address dial on the CPU. Set the address compare switch to any logical address; set the address compare control to normal stop. Processing is suspended when the specific address is referenced. By displaying the PSW (DP), the address of the instruction referencing the work area address can be determined.

To stop processing only when a specific value is stored at a particular address, follow this example after you determine the real address of the data area: If the work area starts at real address X'10000' and you wish to stop when the first byte at that address becomes a X'80', set the data dials to X'80'. The address dials should reflect the real address of X'10000' and the address compare control should be set to STOP.

Diagnostic Techniques by Area

Communications Task

Communications task failures can be localized by observing these guides:

1. When an ABEND code or failure message is printed, refer to the OS/VS System Codes or OS/VS1 System Messages publication listed in the *Preface*. These manuals generally define the problem and the action required to correct that problem.
2. For ABENDs it is necessary to refer to a dump to get further information. The communications task has its own TCB and PRB. These two blocks should be checked to determine if the communications task is active or waiting. If the PRB shows that a WAITM was issued from IEECMAWR, the failure is probably not in the communications task; other tasks in the system should then be checked. (This condition usually indicates that the console request key was pressed after the failure occurred.)
3. The communications task contains a data area (UCM) and uses several queues (WQE, RQE, DQE, and UCME). A check of the information contained in them can isolate the problem to a specific function of the communications task.

JES Reader

If a recursive ABEND occurs, take a dump at the entry to the STAE exit for the first occurrence. Then check status indicators and passback in work areas.

JES Writer

A basic diagnostic technique is to first find the writer work area and then start checking the dump for switches, passback, buffer pointers, job name, job class, and information relating to the type and status of the data set being processed.

System Restart

A basic diagnostic technique is to check the return codes from the queue manager, termination, and JECS. System restart interfaces extensively with these components; their passback, along with system status at failure time, determine which restart routines are executed.

Allocation

A basic diagnostic technique is to check the TIOT after completion of allocation. To do this, stop at XCTL at EXIT from allocation, where allocation exits to IEFALRET. Then register 1 points to a parameter list of six words. The fourth word in the list points to the TIOT list, and the first word of the TIOT list points to the TIOT.

Another technique is to stop on the return codes passed by JECS setting an address stop at the beginning of module IEFYSVMS in load modules IEFW21SD, IEFSD526, IEFMCVOL, and IEFVM6LS. Then register 14 contains the address of the calling routine. By checking the calling routine, you can locate the RPL and determine the type of JECS function to be performed and if the parameters in the RPL are correct.

Queue Manager

When an I/O error occurs, a useful diagnostic technique is to check the I/O control blocks, especially the IOB, to determine the type of error involved.

A branch zap can be done in module IEFQMJ01 at label QMRAW35 (see the microfiche listing for displacement) to trap I/O errors on a read/write to the queue or SWADS. When QMRAW35 receives control, an I/O error has occurred and register 2 points to the ECB/IOB. Register 6 points to the caller's save area.

Initiator

If the failure does not take down the initiator and partition recovery occurs, information about the failure environment can be obtained from the DAR dump data set (SYS1.DUMP) if it was allocated.

If partition recovery does not occur, locate the TRCB/CSCB chain for the partition (partition PIB field SD33TRCN). Each system task active at the time of failure is issued an internal stop command and field TRCBSW1 in its TRCB is turned on with an X'08' to indicate an internal stop. Determining which tasks were or were not stopped helps to determine where clean-up failed.

It may not be possible to recreate the failure with other jobs. If so, the peculiarities of the failing job may aid in determining the problem area.

To repair the initiator, use the HMASPZAP (superzap) service aid program to install a temporary fix. For example:

- If there is a FREEMAIN problem for space in a problem program area use a NO-OP, because all this storage is freed by abend. (All programs return to the scheduler through abend, so this storage is automatically freed when the next task is finished.)
- If the problem is an SMF exit routine which determines if SMF is in the system, and SMF is either not in the system or not a necessity, superzap out the SMF exit routine.

Knowing the system option interfaces can be helpful.

To resolve a specific problem in the initiator, you can set stops at various points, enabling you to gather information. These stops can be set in the initiator by overlaying at the CIM hooks with a branch to asterisk or to the zaparea. The CIM hooks can be found by checking the microfiche listing for the following modules:

Module	*HCIM Hook Location*
IEFSD161	Return from the interpreter.
IEFSD162	Return from allocation.
IEFMF263	Prior to problem program (CSECT IEFSD263)
IEFSD515	After problem program.
IEFSD510	When initiator has no work.

Termination

You can use two types of traps to diagnose termination modules. These traps enable the user to stop on return codes passed by JECS and to obtain a storage dump at job or job step termination when other attempts at stops fail.

Stopping on return codes from JECS requires the user to set an address stop at the beginning of module IEFYSVMS in load module IEFSD161. Then register 14 contains the address of the calling routine. Register 4+X'2E4' is the RPL passed to JECS. The user can determine from the RPL the type of JECS function that was to be performed and if the parameters in the RPL are correct.

If SMF=FULL, then obtaining a storage dump at job or job step termination requires the user to set a data compare trap for a X'53' at location X'8B'. Register 1 then points to the SMF record. Register 1+X'5' points to the one-byte record number. Record 4 is step termination, and record type 5 is the job termination.

The type 4 record contains a step number; therefore a dump can be taken of the step in question or at the end of the job (record type 5). This dump, together with a SWADS and job queue dump, should provide sufficient information to determine if an error occurred in the termination modules.

Note: You must use OPT=2 or DSV=2 or 3 to create a type 4 record.

Command Processors

The following diagnostic procedure is recommended:

1. Check the TCB for system error codes.
2. Check the RB for the module name.
3. Check the log for the command issued and the message returned.
4. Get a listing of the command processor module and determine which interface was not complete.

Interpreter

Generally data type problems or I/O type problems cause interpreter problems.

Data type problems generally result from incorrectly built control blocks, causing invalid messages or allocation problems. Therefore, it is necessary to trap the problem while the control block is being built. By using the data compare trap, the user can stop on SVC 60 (X'3C') when the interpreter issues its STAE macro instruction in module IEFVH1. Register 12 points to the interpreter work area (IWA). By setting a reference trap on a specific field within the work area, the user can verify the integrity of the tables as they are being built.

All major interpreter I/O requests pass through the following modules:

IEFVSPL (spool manager interface): All I/O requests for these spooled data sets are issued here: JCL (JL), proclib (PL), system messages (SM), and SYSOUT (SO). A trap can be set at label SPLERR to catch a spool I/O error condition.

IEFVHQ (queue manager interface): All I/O requests for the job queue and SWADS are issued here. A trap can be set at label H2 to catch the job queue or SWADS I/O error condition.

These two are utility modules for other interpreter modules, which actually initiate the I/O requests. The following are the major I/O requests by module:

IEFVH1	Open (ODS) for JL and PL data sets.
IEFVHA	GET for JL and PL data sets; READ from
IEFVHE	Initial (ODS) for SM data set.
IEFVHEB	Initial assign/start for SWADS space; read JMR from the job queue.
IEFVHH	End ENDS SM data set; open (ODS) for all SM data sets after the first data set; writes VOLT, DSNT, step ACT, SCT, DSENQ, and JCT to SWADS; writes JMR to the job queue.
IEFVGM	PUTS to the SM data set.
IEFVSD13	Assigns DSBs on the job queue.
IEFVSCDQ	Writes SCDs to the job queue.
IEFVINA	(In-stream procedure only) GETS from the JL data set.
IEFVJA	Writes job ACT to SWADS
IEFVEA	POINT to the PL data set and JL data set (for in-stream procedures); FIND for SYS1.PROCLIB; writes step ACT and SCTX to SWADS.
IEFVDA	ODS and ENDS for SO data set; writes JFCB, SIOT, JFCBX, VOLT, and DSNT to SWADS.

IEFVDBSD	Writes DSENQ to the SWADS.
IEFVHN	ENDS for JL, PL, and SM data sets; destructive close CLOD for I/O error conditions; nondestructive CLOD.

WTP

Refer to the *Return Codes* portion of *Diagnostic Aids* of this manual to interpret the return codes from the spool manager.

SMF

The following technique can determine what part of the system is in control when an SMF record is being written:

1. Put a data compare trap on location X'8B' (the location of the last SVC code) for an SMF writer SVC (X'53').
2. When the data compare trap occurs, register 1 points to the SMF record being written. The sixth byte of that record indicates the type of record being written. The record types are as follows:

Record Types	Module That Writes It	Record Description	Necessary Keywords (MAN=ALL)
0	IEESMFI2	IPL	OPT = 1 or 2
1	IEFSMFLK IEESMFI2	Wait Time	OPT = 1 or 2
2	IFASMFDP	Dump Header	OPT = 1 or 2
3	IFASMFDP	Dump Trailer	OPT = 1 or 2
4	IEFSMFLK	Step Termination	OPT = 2
5	IEFSMFLK	Job Termination	OPT = 1 or 2
6	IEFOSC01 IEFOSC02 IEFOSC05	Output Writer	OPT = 1 or 2
7	IEESMFWT	Data Lost	OPT = 1 or 2
8	IEESMFI2	I/O Configuration	OPT = 1 or 2
9	IEE2303D	VARY ONLINE	OPT = 1 or 2
10	IEFXJIMP	Allocation Recovery	OPT = 1 or 2
11	IEFXCSSS IEFZHMSG	VARY OFFLINE	OPT = 1 or 2
12	IEE1403D	End of Day	OPT = 1 or 2
13	IEEDFINA	Dynamic Storage Configuration	OPT = 1 or 2
14	Data Management	Input or RDBACK DS Activity	DSV = 2 or 3
15	Data Management	Output, UPDAT, INOUT, OUTIN DS Activity	DSV = 2 or 3
17	Data Management	Scratch Data Set Status	DSV = 2 or 3, OPT = 2, REC = 2
18	Data Management	RENAME data set status	DSV = 2 or 3, OPT = 2
19	Data Management	Direct Access Volume	DSV = 1 or 3, OPT = 2
20	IEESMFIE	Job Commencement	DSV = 2 or 3, OPT = 2

Register Usage

Module	Register Contents
IEECIR50	12-base register 15-linkage address
IEECLCTX	1- input parameter list pointer 4- MCS UCM pointer 9- UCME pointer 10-UCM pointer 11-base register 14-return address
IEECMAWR	2- UCM pointer 4- UCM prefix pointer 9- base register
IEECMCTR	1- XCTL address pointer
IEECMCTX	1- input parameter list pointer 3- CVT pointer 4- MCS UCM pointer 9- UCME pointer 10-UCM pointer 11-base register 14-return address
IEECMDCM	2- UCM pointer 4- UCM prefix pointer 6- WQE pointer 10-base register 11-DOM element pointer 14-return address
IEECMDSV	1- subroutine to be used 2- UCM pointer 7- MCS UCM prefix pointer 10-base register 14-return address
IEECMOCP	1- input parameter list pointer 2- extended SVRB pointer 11-UCME pointer 12-base register 13-register save area pointer 14-return address 15-XCTL parameter list pointer
IEECMPMC	1- input parameter list pointer 8- WQE pointer 9- DCB pointer 10-UCM pointer 11-CXSA pointer 12-base register 13-register save area pointer 14-return address
IEECMPMP	1- input parameter list pointer 8- WQE pointer 9- DCB pointer 10-UCME pointer 11-CXSA pointer 12-base register 13-register save area pointer 14-return address
IEECMPMX	1- input parameter list pointer 8- WQE pointer 9- DCB pointer 10-UCME pointer 11-CXSA pointer 12-base register 13-register save area pointer 14-return address
IEECMPM1	1- pointer to XSA 2- 0 MLWTO line just written 4 MLWTO line to be written 7- pointer to UCM 8- pointer to WQE 9- pointer to DCB 10-pointer to UCME 11-pointer to XSA
IEECMWSV	2- UCM pointer 4- MCS UCM prefix 6- WQE pointer 15-base register
IEECMWTL	4- MCS UCM prefix pointer 12-base register 14-return address
IEECNCTX	1- input parameter list pointer 4- MCS UCM pointer 9- UCME pointer 10-UCM pointer 11-base register 14-return address
IEECOCTX	1- input parameter list pointer 4- MCS UCM pointer 9- UCME pointer

Module	Register Contents
	10-UCM pointer 11-base register 14-return address
IEECVCRA	2- base register 10-POST completion code 11-ECB pointer 12-COMTASK TCB pointer 14-branch and link
IEECVCRX	1- UCM pointer 2- return address 8- base register 11-UCMXECB pointer
IEECVCTE	14-return address pointer
IEECVDOM	6- CVT pointer 14-return address
IEECVML3	0- message number and UCM entry ID 1- pointer to WPL 4- pointer to TCB 5- pointer to SVRB 10-pointer to UCM 14-return address
IEECVML5	2- 0 fill new minor buffer 4 chain a major or a minor 8 fill a new buffer on connection 3- message number 4- UCM ID/flag bytes 1,2,3 5- pointer to major 6- return address 7- pointer to minor 10-pointer to UCM 12-pointer to XSA
IEECVML6	2- 0 NOP (go to IEECVML7) 4 force end to MLWTO, exit 8 get two buffers C get one buffer 3- message number 4- UCM ID/flag bytes 1,2,3 5- pointer to major 6- return address 7- pointer to minor 10-pointer to UCM 12-pointer to XSA
IEECVML7	3- message number 4- UCM ID/flag bytes 1,2,3 5- pointer to major 6- return address 7- pointer to minor 10-pointer to UCM 12-pointer to XSA
IEECVOCC	1- work area pointer 2- extended SVRB pointer 11-UCME pointer 12-base register 13-register save area pointer 14-return address 15-XCTL parameter list pointer
IEECVOCX	2- extended SVRB pointer 11-UCME pointer 12-base register 13-register save area pointer 14-return address 15-XCTL parameter list pointer
IEECXDOM	0- possible UCM ID 0- describes register 1 1- message number of MSGLIST address 4- UCM pointer 6- UCM prefix pointer 11-base register
IEEDFINA	5- CVT pointer 6- SMF record pointer 9- MSRDA pointer
IEEDFIN1	3- TSCB pointer 6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer 10-TCB pointer 11-partition work area 14-branch and link
IEEDFIN2	6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer 14-branch and link
IEEDFIN3	6- internal data area pointer 7- base register 8- MSRDA pointer

Module	Register Contents
	9- CVT pointer 10-TCB pointer 14-branch and link
IEEDFIN4	3- TSCB pointer 6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer 10-TCB pointer
IEEDFIN5	1- PIB pointer 6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer 10-TCB pointer 12-boundary box pointer 14-PQA pointer
IEEDFIN6	3- base for displacement into messages 6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer
IEEDFIN7	6- internal data area pointer 7- base register 8- MSRDA pointer 9- CVT pointer 14-branch and link
IEEDFIN8	6- DEFINE work area
IEELOGWR	8- base register 12-LCA pointer 13-register save area pointer 15-branch and link
IEEMB800	5- EXCPNTS table entry pointer 6- UCB pointer 7- UCB LUT pointer 11-base register 12-data area pointer 13-register save area pointer
IEEMFWTO	0- possible UCM pointer 1- user parameter list pointer 4- TCB of task issuing SVC pointer 5- input list address pointer (SVRB address) 7- WTOR message length

Module	Register Contents
IEEPSN	12-base register 13-register save area pointer
IEESD561	2- XSA pointer 6- QSWORK pointer 7- CSCB pointer 9- base register
IEESD562	2- XSA pointer 6- QSWORK pointer 7- CSCB pointer 8- work area pointer 9- base register
IEESD563	2- XSA pointer 5- QMPA pointer 6- QSWORK pointer 7- CSCB pointer 8- queue manager work area pointer 9- base register
IEESD564	2- XSA pointer 6- QSWORK pointer 7- CSCB pointer 8- work area pointer 9- base register
IEESD565	2- XSA pointer 5- QMPA pointer 6- QSWORK pointer 7- CSCB pointer 8- work area pointer 9- base register
IEESD571	1- CSCB pointer 2- XSA pointer 6- MSRDA pointer 8- base register 9- UCB pointer 15-CVT pointer
IEESD575	9- base register 13-register save area pointer 14-return address
IEESMFAL	5- pointer to volume serial or unit 8- UCB pointer 11-base register 12-SMCA pointer

Module	Register Contents
IEESMFIT	6- work area pointer 10-base register 13-register save area pointer
IEESMFI2	10-base register 12-SMCA pointer 14-return address
IEESMFI3	6- pointer to save area and work area 10-base register 12-SMCA pointer
IEESMFOI	2- base register 12-SMCA pointer 13-register save area pointer 14-return address 15-branch and link
IEESMFOP	4- JFCB pointer 5- SMCA device area pointer 9- UCB entry pointer 10-TIOT entry pointer 11-base register 12-SMCA pointer 13-work area pointer
IEESMFWT	5- DCB pointer 7- SMF buffer pointer 8- base register 13-work area pointer
IEESMF8C	9- base register 12-SMCA pointer 13-work area pointer 14-branch and link
IEEVCTI	9- base register 10-UCME pointer 11-base register 13-register save area pointer 14-return address
IEEVJCL	6- TRCB pointer 10-SDT pointer 11-CSCB pointer 12-base register
IEEVLIN	0- queue manager interface pointer 2- base register 3- CVT pointer

Module	Register Contents
	4- WTO work area pointer 5- queue manager work area pointer 12-LCA pointer 13-register save area pointer
IEEVMNT1	11-CSCB pointer 12-base register 13-TRCB pointer
IEEVMNT2	9- TIOT pointer 12-base register 13-register save area pointer
IEEVRCTL	5- TRCB pointer 10-CSCB pointer 12-base register
IEEVRFRX	1- input parameter list pointer 3- function control 9- base register 15-return code
IEEVRJCL	1- RPL pointer 2- JCL record pointer 12-ACB pointer 13-register save area pointer 15-base register
IEEVROUT	0- request for BPL space 1- BPL pointer 3- base register 4- TCB pointer 5- RB pointer 6- WPL pointer 8- QID pointer 12-CVT pointer 13-register save area pointer (XSA) 15-return code for BPL request
IEEVSMSG	4- parameter list pointer 12-base register
IEEVSTAR	10-SDT pointer 12-base register
IEEVWTOR	0- UCM ID 1- user parameter list pointer 4- pointer to TCB of task issuing SVC 10-UCM pointer

Module	Register Contents
	11-base register 12-CVT pointer
IEEXEDNA	2- XSA pointer 5- UCB pointer 12-UCME 14-return address
IEE00110	1- first time indicator 3- XSA address 11-base register 14-return address
IEE0303D	0- UCM indicator 1- CIB pointer 2- XSA pointer 3- base register
IEE0303F	1- JECS interface 2- enable/disable state indicator 3- work area pointer 5- RPL pointer 8- text buffer pointer 9- base register 12-LCA pointer 13-register save area pointer 14-exit address 15-XCTL address
IEF0403D	0- UCM indicator 1- CIB pointer 2- XSA pointer 3- base register 12-UCM pointer
IEE0403F	2- work area pointer 5- RPL pointer 6- LRCB pointer 8- spool and queue manager linkage 10-branch and link 11-base register 12-LCA pointer
IEE0503D	2- XSA pointer 4- base register
IEE0603D	2- XSA pointer 3- parameter pointer 4- KEYTAB pointer 6- data 7- time
	9- parameter pointer 12-base register 15-XCTL address
IEE0903D	3- CVT pointer 5- base register 9- pseudo clocks pointer 14-return address
IEE1A03D	1- message code 3- input buffer pointer 8- base register
IEE1B03D	8- base register 13-UCM pointer
IEE1103D	2- XSA pointer 3- CVT pointer 8- UCB pointer 9- unit name pointer 12-base register
IEE1403D	3- CVT pointer 7- SMCA pointer 11-EOD operand pointer 12-base register 15-LCA pointer
IEE1603D	2- XSA pointer 3- CVT pointer 4- delimiter pointer 6- end of parameter pointer 8- UCB pointer 9- unit name pointer 12-base register 14-return address 15-XCTL address
IEE1903D	2- XSA pointer 3- base register 4- MSRDA pointer 8- TCB pointer 10-CVT pointer
IEE2303D	1- SMF record pointer 2- XSA pointer 3- SMCA pointer 4- XSA pointer 6- bin number 8- UCB pointer 9- CIB pointer 10-branch and link

Module	Register Contents
	12-base register 13-number of left parentheses 15-UCM pointer
IEE2903D	2- XSA pointer 9- base register 10-message area pointer 12-branch and link
IEE3203D	2- XSA pointer 4- keyword characters counter 6- Pointer to first character of keyword 7- end of buffer pointer 9- Pointer to current character of operand 11-keyword table pointer 12-base register
IEE3303D	2- XSA pointer 3- base register 4- UCME pointer 6- CVT pointer 7- end of buffer pointer 8- STAE 9- units pointer 10-UCM pointer 14-return address
IEE3503D	2- XSA pointer 3- base register 7- CVT pointer 10-UCM pointer 14-return address 15-branch and link
IEE3703D	2- XSA pointer 12-base register
IEE4303D	2- XSA pointer 3- UCM pointer 4- UCME pointer 5- UCB pointer 13-base register
IEE4403D	2- XSA pointer 4- UCM pointer 6- bin number 8- UCB pointer 9- CIB pointer 10-branch and link

Module	Register Contents
	12-base register 13-number of left parentheses
IEE4503D	2- XSA pointer 4- UCM pointer 5- UCME pointer 9- CIB pointer 10-MSRDA pointer 11-CSCB pointer 12-base register 14-return address
IEE4603D	0- WTO ID 2- XSA pointer 3- base register 8- STAE 9- unit pointer 10-UCM pointer 12-UCB pointer 14-return address
IEE4703D	2- XSA pointer 3- base register 11-UCM pointer 14-return address 15-CVT pointer
IEE4903D	2- XSA pointer 3- base register 8- STAE 9- unit pointer 10-UCM pointer 12-UCB pointer 14-return address
IEE5603D	2- XSA pointer 7- UCM entry pointer 8- RCT pointer 9- UCM pointer 10-CVT pointer 12-base register 14-return address
IEE5903D	2- XSA pointer 7- UCM entry pointer 8- RCT pointer 9- UCM pointer 10-CVT pointer 12-base register 14-return address
IEE60110	2- CSCB address 3- XSA address

Module	Register Contents
	11-base register 14-return address
IEE6503D	2- XSA pointer 11-time 12-base register 13-date
IEE6703D	2- XSA pointer 4- resident DCM pointer 5- transient DCM pointer 9- UCM entry pointer 10-UCM pointer 12-screen area control block pointer 13-base register 14-return address
IEE6803D	2- XSA pointer 7- screen area control block pointer 8- UCM entry pointer 9- transient DCM pointer 10-resident DCM pointer 12-UCM pointer 13-base register 14-return address
IEE6903D	2- XSA pointer 7- screen area control block pointer 8- UCM entry pointer 9- transient DCM pointer 10-resident DCM pointer 13-base register
IEE7203D	2- XSA pointer 3- base register 5- UCME pointer
IEE7303D	2- XSA pointer 3- base register 5- UCME pointer
IEE7503D	2- XSA pointer 3- UCM pointer 10-resident DCM pointer 11-routing control table pointer screen area control block pointer 12-UCM entry pointer 13-base register 14-return address

Module	Register Contents
IEE7603D	2- XSA pointer 3- UCM pointer 10-resident DCM pointer 11-UCM entry pointer 12-screen area control block pointer 13-base register 14-return address
IEE7703D	2- XSA pointer 7- screen area control block pointer 8- UCM entry pointer 9- transient DCM pointer 10-resident DCM pointer 12-UCM pointer 13-base register
IEE7803D	1- major WQE pointer 2- XSA pointer 4- minor WQE pointer 8- UCM entry pointer 11-base register 12-UCM pointer
IEE7903D	2- XSA address 11-base register 13-save area address 14-return address 15-return code register
IEE6603D	12-base register
IEE9903D	2- XSA pointer 5- parameter pointer 6- branch and link
IEFAB400	6- input parameter pointer 8- DDTBL pointer 11-base register 13-register save area 14-return address
IEFAB401	3- LBWA pointer 4- device entry pointer 5- logical channel table pointer 6- candidate list pointer 7- DDTBL pointer 11-base register 13-register save area 14-return address
IEFAB403	0- TCB pointer 1- RQE pointer

Module	Register Contents
	2- IOB pointer 3- DEB pointer 4- DCB pointer 7- UCB pointer 11-EXCP count 12-EXCPCNTS table entry pointer 14-return address 15-base register
IEFAB405	15-return code
IEFAB406	15-return code
IEFAB407	15-return code
IEFAB408	15-return code
IEFAB416	15-return code
IEFACTFK	14-return address
IEFACTLK	3- work area pointer 4- parameter list pointer 5- JMR pointer 7- return address 9- base register 10-SCT pointer 11-JCT pointer 12-LCT pointer 13-register save area 14-return address
IEFACTRT	1- return code 14-return address 15-return code
If SMF = FULL	
IEFBMGET	1- RPL pointer 3- length of BUTE information 6- GETBUF RPL pointer 7- BUTBL pointer 8- JESCT pointer 11-base register 13-register save area pointer 14-return address
IEFBMPUR	6- DEB pointer 7- BUTBL pointer 11-base register 12-BUTBL entry pointer
IEFBMPUT	1- RPL pointer 6- RPL pointer 7- BUTBL pointer 8- JESCT pointer 11-base register 14-return address
IEFDSDRP	2- branch and link 12-base register 13-register save area pointer 14-return address
IEFDSLST	1- parameter list pointer
IEFDSOAL	1- parameter list pointer 11-base register 13-register save area pointer 14-return address 15-linkage
IEFDSOCP	6- DSOCB pointer 8- base register 11-base register
IEFDSOFB	8- DSOCB pointer 9- base register 10-JCB pointer
IEFDSOSM	6- DSOCB pointer 11-base register
IEFDSOWR	7- DSOCB pointer 11-base register
IEFDSTBL	1- parameter list pointer
IEFDSTRT	14-return address
IEFIDUMP	4- work area pointer 9- base register 11-TCB pointer 12-LCT pointer 14-return address 15-entry point
IEFIIC	2- TIOT pointer 4- LOT pointer 6- PIB pointer 7- CVT pointer 12-base register 15-linkage

Module	Register Contents
IEFINTQA	10-work area pointer 11-SWADS control block pointer 12-base register 13-register save area pointer
IEFMCVOL	2- JCT pointer 3- SCT pointer 4- JFCB pointer 5- work area pointer 7- VCB pointer 9- base register 10-SIOT pointer 11-register save area pointer 12-LCT pointer 14-return address 15-linkage
IEFMF102	2- JCT pointer 4- CSCB pointer 6- ENQ list pointer 10-LOT pointer 12-base register 13-register save area pointer
IEFMF105	1- QMRES pointer 2- LOT pointer 8- CVT pointer 9- TCB pointer 12-base register
IEFMF106	2- QMPA pointer 8- LOT pointer 12-base register 13-register save area pointer
IEFMF263	2- TCB pointer 8- parameter pointer 12-base register 13-register save area pointer 14-SVC 3 pointer
IEFMSGJP	1- parameter list pointer 11-base register 13-register save area pointer 14-return address
IEFORMAT	3- base register 4- parameter list pointer 5- work area pointer
IEFOSC01	9- base register 12-work area pointer
IEFOSC02	9- base register 12-work area pointer
IEFOSC03	9- base register 12-work area pointer
IEFOSC04	9- base register 12-work area pointer
IEFOSC05	9- base register 12-work area pointer
IEFOSC06	1- parameter list pointer 9- base register 12-work area pointer
IEFOSC07	11-base register 12-work area pointer
IEFOSC08	7- spanned record pointer 9- base register 12-work area pointer
IEFPRES	3- LUT pointer 4- UCB pointer 5- preslist pointer. 7- messages pointer 9- base register 13-register save area pointer 14-return address
IEFPRTXX	6- JFCB pointer 8- DCB pointer 13-base register
IEFQMIFC	12-base register 13-register save area pointer
IEFQMJ01	8- base register
IEFQMJ02	4- QMPA pointer 8- base register 13-register save area pointer
IEFQMJ03	8- base register 9- base register
IEFQMMAC	3- work area pointer 4- parameter list pointer 11-base register 13-register save area pointer 14-return address
IEFRPREP	9- base register 12-LCT pointer
IEFSCAN	9- base register 12-LUT pointer 14-return address

Module	Register Contents
IEFSD055	1- parameter area pointer 14-return address
IEFSD097	9- base register 12-LCT pointer 15-return code
IEFSD101	1- LOT pointer 12-base register
IEFSD160	8- IEL pointer 12-base register
IEFSD161	10-LOT pointer 11-DER pointer 12-base register 13-register save area pointer
IEFSD162	5- CSCB pointer 6- SCT pointer 7- JCT pointer 8- LCT pointer 12-base register
IEFSD164	5- SCT pointer 9- DCB pointer 11-LOT pointer 12-base register
IEFSD165	1- JCT pointer 5- SCT pointer 11-LOT pointer 12-base register
IEFSD166	9- CSCB pointer 11-LOT pointer 12-base register
IEFSD168	8- CSCB pointer 9- JCT pointer 11-LOT pointer 12-base register
IEFSD180	2- JCT pointer 3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer 13-register save area pointer 14-return address
IEFSD195	9- base register 14-return address 15-return code
IEFSD21Q	1- parameter list pointer 9- base register

Module	Register Contents
	14-branch and link 15-branch
IEFSD22Q	2- SCT pointer 3- JCT pointer 4- work area pointer 8- JSCB pointer 9- base register 12-LCT pointer 13-register save area pointer 15-return code
IEFSD300	1- parameter area pointer 8- base register 11-base register
IEFSD301	8- base register 14-return address
IEFSD302	8- base register 14-return address
IEFSD303	8- base register 14-return address
IEFSD304	8- base register 14-return address 15-branch and link
IEFSD305	8- base register 11-base register 13-register save area pointer
IEFSD309	11-base register 14-return address
IEFSD31Q	2- SCT pointer 3- JCT pointer 4- termination work area pointer 9- base register 12-LCT pointer 13-register save area pointer 15-return code
IEFSD41Q	1- parameter list pointer 9- base register 12-LCT pointer 15-return code
IEFSD42Q	1- parameter list pointer 2- SCT pointer 3- JCT pointer 4- termination work area pointer 5- TCB pointer 6- JSCB pointer 9- base register 12-LCT pointer 13-save area in LCT pointer

Module	Register Contents
IEFSD510	6- TCB pointer 7- CVT pointer 8- CIB pointer 9- CSCB pointer 10-TRCB pointer 11-PIB pointer 12-base register 13-register save area pointer
IEFSD515	9- PIB pointer 11-LOT pointer 12-base register
IEFSD519	5- QMPA pointer 6- CSCB pointer 7- QMRES pointer 10-PIB pointer 11-base register
IEFSD533	1- input parameter list pointer 11-base register
IEFSD536	5- message code 8- condition code 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFSD598	2- base register 4- TCB pointer 5- RB pointer
IEFSMCLD	1- RPL pointer 3- SUTBL pointer 4- DSD pointer 11-base register 13-register save area pointer 14-return address
IEFSMEND	2- RPL pointer 3- SUTBL pointer 4- DSD pointer 8- LRCB pointer 11-base register 13-register save area pointer 14-return address
IEFSMFAT	1- TCB pointer 4- base register 5- TIOT pointer 8- UCB pointer 9- SMCA pointer 10-TCT pointer 13-register save area pointer

Module	Register Contents
IEFSMFIE	3- pointer to accounting fields 4- register 13 contents 7- JCT pointer 8- ACT pointer 9- SCT pointer 10-LCT pointer 12-base register 13-register save area pointer
IEFSMFLK	9- base register 10-JCT pointer 11-SCT pointer 12-LCT pointer 13-register save area pointer
IEFSMFSO	8- RPL pointer 11-base register 13-register save area pointer 14-return address 15-output limit extension value
IEFSMFWI	2- base register 9- parameter area pointer 12-LCT pointer 13-register save area pointer
IEFSMGET	5- RPL pointer 6- LRCB pointer 7- buffer pointer 8- user data area pointer 11-base register 13-register save area pointer
IEFSMIFC	0- LRCB pointer 1- RPL pointer 7- DEB pointer 8- RPL pointer 11-base register 13-register save area pointer 14-return address 15-spool manager routine entry point
IEFSMODS	2- RPL pointer 3- SUTBL pointer 4- DSD pointer 8- LRCB pointer 11-base register 13-register save area pointer 14-return address
IEFSMPUT	5- RPL pointer 6- LRCB pointer 7- buffer pointer 8- user data area pointer 11-base register

Module	Register Contents
	13-register save area pointer 14-return address
IEFSMREP	2- buffer pointer 3- RPL pointer 6- LRCB pointer 8- JESCT pointer 11-base register 13-register save area pointer 14-return address
IEFSTDSC	8- DER pointer 11-base register
IEFUIV	1- input parameter list 14-return address 15-return code
IEFUJI	1- input parameter list 14-return address 15-return code
IEFUJP	1- input parameter list 14-return address 15-return code
IEFUJV	1- input parameter list 14-return address
IEFUSI	1- input parameter list 14-return address 15-return code
IEFUSO	1- input parameter list output extension (output) 14-return address 15-return code
IEFUTL	1- input parameter list output extension (output) 14-return address 15-return code
IEFVDA	1- pointer to base register 6- IEFVHQ pointer 8- IEFVGT pointer 9- IEFVGK pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVDBSD	11-base register 12-IWA pointer 13-register save area pointer

Module	Register Contents
IEFVEA	8- SCT pointer 9- IEFVGT pointer 10-EWA pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVFA	11-base register 12-IWA pointer 13-register save area pointer
IEFVFB	11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFVGI	1- dictionary entry pointer 10-auxiliary work area pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return addresss
IEFVGK	10-work area pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFVGM	2- message number or 0 9- JCL statement pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFVGS	1- output pointer 3- input pointer 10-auxiliary work area pointer 12-IWA pointer 13-register save area pointer 14-return address
IEFVGT	2- length of parameter 3- pointer to parameter length byte 4- PDT pointer 10-work area pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address

Module	Register Contents
IEFVHA	11-base register 12-IWA pointer 13-register save area pointer
IEFVHC	8- JCT pointer 9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHCB	9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHE	9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHEB	7- SCT pointer 8- JCT pointer 9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHEC	7- SCT pointer 8- JCT pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHF	3- CWA pointer 9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHH	6- queue parameter list pointer 7- SCT pointer 8- JCT pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHM	5- verb pointer 9- JCL statement pointer 10-statement parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVHN	1- exit parameter list pointer 2- return code 11-base register 13-register save area pointer
IEFVHQ	1- QMPA pointer 8- JCT pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFVH1	1- parameter list pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVINA	7- work area pointer 8- JCT pointer 10-EWA pointer 11-base register 12-IWA pointer 13-register save area pointer
IEFVINB	1- parameter list pointer 7- work buffer pointer 11-base register 12-IWA pointer 15-return code
IEFVINC	1- parameter list pointer 2- work area pointer 3- count pointer 6- return code 11-base register
IEFVINE	1- input buffer pointer 15-return code
IEFVJA	8- JCT pointer 9- IEFVGT pointer 10-local work area pointer 11-base register 12-IWA pointer

Module	Register Contents
	13-register save area pointer 15-IEFVGK pointer
IEFVJIMP	2- SCT pointer 3- JCT pointer 4- termination work area pointer 5- save area in LCT pointer 9- base register 12-LCT pointer
IEFVKIMP	9- base register 12-LCT pointer
IEFVMA	11-base register 12-reader or writer work area pointer 13-register save area pointer
IEFVMB	2- pointer to JCL being processed 11-base register 12-work area pointer 13-register save area pointer
IEFVMC	2- pointer to JCL record being examined 3- return address 4- base register 11-IEFVMB pointer 12-work area pointer 13-register save area pointer
IEFVMD	9- base register 12-work area pointer
IEFVMLS1	2- JCT pointer 3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer 13-register save area pointer 14-return address
IEFVMLS6	8- JCT pointer 9- base register 12-LCT pointer
IEFVM2LS	3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer
	12-LCT pointer 14-return address
IEFVM3LS	3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer 13-register save area pointer 14-return address
IEFVM4LS	3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer
IEFVM5LS	3- SCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer
IEFVM76	2- JCT pointer 4- JFCB pointer 9- base register 10-SIOT pointer 12-LCT pointer
IEFVRRC	10-base register 11-base register 12-work area pointer 13-register save area pointer 14-return address
IEFVRR1	11-base register 12-work area pointer 13-register save area pointer 14-return address
IEFVRR2	11-base register 12-work area pointer 13-register save area pointer 14-return address
IEEVRR3	10-branch and link 11-base register 12-work area pointer 13-register save area pointer 14-return address

Module	Register Contents
IEFVSCDQ	2- QMPA pointer 10-JCT pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFVSDRD	12-base register 14-return address
IEFVSD13	2- QMPA pointer 4- base register 7- SCT pointer 8- SCD pointer 9- SIOT pointer 10-DWA pointer 11-JCT pointer 12-IWA pointer 13-register save area pointer 14-return address
IEFVSPL	1- RPL pointer 11-base register 12-IWA pointer 13-register save area pointer 14-return address
IEFWAALC	2- VCB pointer 3- WAA parameter list pointer 9- WAMDA pointer 11-base register 13-register save area pointer 14-return address
IEFWAA01	3- WAA parameter list pointer 11-base register 13-register save area pointer 14-return address
IEFWAA02	3- WAA parameter list pointer 11-base register 13-register save area pointer 14-return address
IEFWAA03	1- code for introductory branch table 3- WAA parameter list pointer 11-base register 13-register save area pointer 14-return address

Module	Register Contents
IEFWAD	7- UCB pointer 8- base register 10-CVT pointer 11-work area pointer 13-register save area pointer 14-return address
IEFWAMAP	1- end of VCB.DCB.DEB list pointer 12-base register 13-register save area pointer 14-return address 15-passback register
IEFWAMGR	1- parameter list pointer 3- base register 13-register save area pointer 14-return address
IEFWA000	9- base register 12-LCT pointer
IEFWAMIN	1- V=R or V=V indicators 9- base register 11-base register 12-base register 13-register save area pointer 14-return address 15-passback register
IEFWARIN	11-base register 13-register save area pointer 14-return address 15-passback register
IEFWCIMP	8- base register 9- base register 12-LCT pointer
IEFWD000	9- base register
IEFWEXTA	9- base register
IEFWSWIN	2- base register 3- JCT pointer 12-LCT pointer
IEFWTP00	4- TCB pointer 5- SVRB pointer 6- input parameter area pointer 10-current console UCB pointer 11-base register

Module	Register Contents
	12-CVT pointer
	14-return address
IEFXCSSS	9- base register
	12-LCT pointer
IEFXDPTH	1- parameter list pointer
	4- CLT pointer
	6- ACB pointer
	7- LCT pointer
	14-return address
	15-base register
IEFXH000	7- base register
	8- ACB pointer
	11-AWT pointer
	12-LCT pointer
	6- UCB pointer
IEFXJIMP	7- SCT pointer
	8- ACB pointer
	9- base register
	11-AWT pointer
	12-LCT pointer
	6- UCB pointer
IEFXKIMP	7- SCT pointer
	9- base register
	12-LCT pointer
IEFXTDMY	9- base register
	12-LCT pointer
IEFXT00D	1- parameter list pointer
	6- SRT pointer
	7- work area pointer
	8- PDQ pointer
	9- base register
	10-JFCB pointer
	11-TIOT pointer
	12-LCT pointer
IEFXT002	6- SRT pointer
	7- work area pointer
	9- base register
	10-JFCB pointer
	11-TIOT pointer
	12-LCT pointer
IEFXT003	1- parameter list pointer
	6- SRT pointer
	7- work area pointer
	9- base register
	10-JFCB pointer
	11-TIOT pointer
	12-LCT pointer
IEFXVNSL	14-return address
	15-return code
IEFXV001	3- test pattern
	9- base register
IEFXV002	1- test pattern
	2- UCB pointer
	9- base register
	14-return address
	15-return code
IEFXV003	1- LUT pointer
	9- base register
	11-work area pointer
IEFX300A	3- AVT pointer
	6- UCBOB pointer
	9- base register
	12-LCT pointer
IEFX5000	6- SRT pointer
	9- base register
	11-AWT pointer
	12-LCT pointer
	14-branch
	15-linkage
IEFYNIMP	2- SCT pointer
	3- JCT pointer
	4- termination work area pointer
	6- TCB pointer
	9- base register
	12-LCT pointer
	13-save area in LCT pointer
IEFYPJB3	2- SCT pointer
	3- JCT pointer
	4- termination work area pointer
	5- SIOT pointer
	6- TCB pointer
	9- base register
	10-UCB pointer
	11-TIOT pointer
	12-LCT pointer
	13-save area in LCT pointer

Module	Register Contents
IEFYSVMS	1- parameter list pointer 2- JCT pointer 3- SCT pointer 9- base register 12-LCT pointer
IEFYTVMS	2- SCT pointer 3- JCT pointer 4- termination work area pointer 5- DSB pointer 6- CSCB pointer 9- base register 10-JSCB pointer 12-LCT pointer
IEFZAJB3	2- SCT pointer 3- TCB pointer 4- termination work area pointer 5- queue entry pointer 6- JCT pointer 9- base register 12-LCT pointer 13-register save area pointer 14-return address 15-entry point
IEFZGJB1	2- SCT pointer 3- JCT pointer 4- termination work area pointer 6- SRT pointer 8- PDQ pointer 9- base register 12-LCT pointer 13-register save area pointer 14-return address
IEFZGST1	2- SCT pointer 3- JCT pointer 4- termination work area pointer 5- TIOT index 6- SRT pointer 7- JFCB pointer 9- base register 12-LCT pointer 13-register save area pointer 14-return address
IEFZGST2	2- SCT pointer 3- JCT pointer 4- termination work area pointer 9- base register 12-LCT pointer

Module	Register Contents
	13-save area in LCT pointer 14-return address
IEFZHMSG	0- TIOT length 1- current SIOT entry pointer 2- current TIOT entry pointer 3- JCT pointer 4- termination work area pointer 5- TIOT index 6- SRT pointer 9- base register 12-LCT pointer 13-register save area pointer 14-return address
IEF065FK	1- LOT pointer 12-base register
IEF160DM	12-base register
IEF160FK	12-base register
IEF161DM	1- LOT pointer 12-base register
IEF161FK	1- LOT pointer 12-base register
IEF263FK	1- initiator parameter list pointer
IGCU103D	9- UCB pointer 10-base register 11-CSCB pointer 12-work area pointer 14-branch and link 15-CVT pointer
IGCU203D	7- UCB pointer 10-work area pointer 11-CSCB pointer 12-base register
IGCU303D	7- UCB pointer 10-work area pointer 11-CSCB pointer 12-base register
IGCU403D	10-work area pointer 11-XSA pointer 12-base register

Module	Register Contents
IGC5A07B	1- CXSA parameter list pointer 2- DCME pointer 3- operand pointer 7- UCM pointer 8- UCM prefix pointer 9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer
IGC5C07B	1- CXSA parameter list pointer 8- UCM pointer 9- DCM pointer 10-UCME pointer 11-CXSA pointer 12-base register
IGC5D07B	1- CXSA parameter list pointer 7- UCM pointer 9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer
IGC5E07B	1- CXSA parameter list pointer 7- UCM pointer 9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer
IGC5F07B	1- CXSA parameter list pointer 10-UCM pointer 11-base register 12-CXSA pointer 13-DCM pointer
IGC5G07B	1- CXSA parameter list pointer 3- extended SVRB pointer 8- OPEN control block pointer 10-DCM pointer 11-UCM pointer 12-base register
IGC5J07B	1- CXSA parameter list pointer 7- DCM pointer 8- UCME pointer 9- base register 10-CXSA pointer 11-UCM pointer 12-WQE pointer

Module	Register Contents
IGC5K07B	1- CXSA parameter list pointer 7- UCM pointer 8- UCME pointer 9- DCM pointer 10-CXSA pointer 11-base register
IGC5L07B	1- CXSA parameter list pointer 10-DCM pointer 11-CXSA pointer 12-base register 13-UCME pointer
IGC5M07B	1- CXSA parameter list pointer 10-DCM pointer 11-CXSA pointer 12-base register 13-UCME pointer
IGC5N07B	1- CXSA parameter list pointer 9- UCME pointer 10-DCM pointer 11-CXSA pointer 12-base register
IGC5O07B	1- CXSA parameter list pointer 8- UCME pointer 9- base register 10-CXSA pointer 11-DCM pointer
IGC5P07B	1- CXSA parameter list pointer 7- entry area pointer 8- channel program area pointer 9- UCM pointer 10-DCM pointer 11-base register 12-CXSA pointer 13-UCME pointer
IGC5Q07B	1- CXSA parameter list pointer 8- channel program area pointer 9- UCM pointer 10-DCM pointer 11-base register 12-CXSA pointer 13-UCME pointer
IGC5R07B	1- CXSA parameter list pointer 9- UCM pointer 10-DCM pointer 11-base register

Module	Register Contents
	12-CXSA pointer 13-UCME pointer
IGC5S07B	1- CXSA parameter list pointer 8- DCM pointer 9- CXSA pointer 10-base register 11-UCME pointer
IGC5T07B	1- CXSA parameter list pointer 10-DCM pointer 11-CXSA pointer 12-base register 13-UCME pointer
IGC5107B	1- CXSA parameter list pointer 3- mask 7- UCM pointer 8- WQE pointer 9- DCM pointer 10-UCME pointer 11-CXSA pointer 12-base register 14-register save area pointer
IGC5207B	1- CXSA parameter list pointer 3- UCME pointer 4- message in DCM pointer 10-IOB pointer 12-DCM pointer 13-CXSA pointer 14-register save area pointer 15-UCB pointer
IGC5307B	1- CXSA parameter list pointer 8- UCM pointer 9- DCB pointer 10-UCME pointer 11-CXSA pointer 12-base register
IGC5607B	1- CXSA parameter list pointer 7- UCM pointer 9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer 13-IOB pointer 15-mask
IGC5707B	1- CXSA parameter list pointer 7- UCM pointer

Module	Register Contents
	9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer 13-IOB pointer 14-branch and link
IGC5807B	1- CXSA parameter list pointer 7- UCM pointer 9- DCM pointer 10-base register 11-CXSA pointer 12-UCME pointer 13-IOB pointer 15-mask
IGC5907B	1- CXSA parameter list pointer 7- UCM pointer 9- DCM pointer 10-base register 11-CXSA pointer 13-IOB pointer 15-mask
IGF2403D	1-parameter list pointer 2 XSA pointer 3- CVT pointer 4- UCB pointer 5- pointer to first operand 6- exit pointer 8- STAE 12-base register 15-return code
IGF2603D	2- XSA pointer 9- RVT pointer 10-CVT pointer 11-MSB pointer 12-base register 13-register save area pointer 14-return address 15-branch register
IGG019DF	2- work area pointer 5- RPL pointer 11-base register 14-return address
IGG019DG	2- ACB pointer 4- RPL pointer 7- pointer to DCB for SETPRT 11-base register 12-work area pointer 13-register save area

Module	Register Contents
	pointer 14-return address
IGG019DH	2- DCB pointer 5- RPL pointer 6- ACB pointer 11-base register 13-register save area pointer 14-return address
IGG019DL	1- RQE pointer 2- IOB pointer 3- DEB pointer 4- ACB pointer 7- UCB pointer 10-completion code 11-ECB pointer 12-TCB pointer 14-return address 15-base register
IGG019DM	2- RQE pointer 3- DEB pointer 4- work area pointer 5- ACB pointer 6- RPL pointer 7- UCB pointer 15-base register
IGG019FM	2- DCB pointer 4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer
IGG019MA	1- DECB pointer 2- IOB pointer 3- read/write area pointer 4- DCB pointer 5- CCWs pointer 6- model channel program pointer 7- count of CCWs completed 11-number of CCWs in channel program 12-base register 13-register save area pointer 14-return address 15-return code
IGG019MB	2- IOB pointer 3- DEB pointer

Module	Register Contents
	4- DCB pointer 5- CCW pointer 6- CCW pointer 7- UCB pointer 10-ECB pointer 12-base register 13-register save area pointer 14-return address 15-entry point
IGG019MR	1- terminal test parameter list pointer 3- DECB pointer 4- base register 5- IOB pointer 6- DCB pointer 14-return address
IGG0196U	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0196V	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0196W	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0196X	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0196Y	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer

Module	Register Contents
	7- current entry pointer 8- current load pointer 11-base register pointer 15-control program list pointer
IGG0196Z	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0201M	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer
IGG0201N	4- OPEN work area pointer 5- parameter list pointer 6- WTG table pointer 7- current entry pointer 8- current load pointer 11-base register 15-control program list pointer

Message/Module Relationship

IEE007I	IEE1603D	71, 102
IEE008I	IEE3703D	58
	IGF2603D	74
IEE009I	IEE0403F	102
IEE010I	IEE0403F	102, 103
IEE011I	IEE1603D	71, 102
IEE012I	IEEVLIN	
IEE014I	IEE0303F	71, 102, 103
IEE015W	IEFPARMS	56
IEE018I	IEE0403F	102
IEE019I	IEE1603D	71, 102, 103
IEE023I	IEE1603D	102
IEE026I	IEE1603D	71, 102, 103
	IGF2503D	92, 93
	IEE5703D	
	IEE7203D	86
IEE030I	IEFDFIN8	56
IEE031I	IEFDFIN8	56
IEE032I	IEE1603D	102, 103
IEE033I	IEE1603D	71, 102, 103
IEE034I	IEE1603D	102, 103
	IEFOSC03	
IEE035I	IEFDFIN8	56
IEE037I	IEELOGWR	102, 103
IEE043I	IEE0403F	102, 103
	IEEVLIN	
IEE044I	IEELOGWR	102, 103
IEE045I	IEE0403F	102, 103
IEF047I	IEFOSC03	
IEE048I	IEFDFIN5	56, 61
IEE052I	IEFSD569	56
IEE054I	IEE6603D	
IEE055A	IEE6603D	
IEE058A	IEE6603D	
IEE079I	IEEDFIN5	61
IEE080I	IEE9803D	56
IEE090I	IEE7903D	67
IEE091I	IEE7903D	67
IEE092I	IEE7903D	67
IEE093I	IEE7903D	67
IEE094I	IEE60110	67
IEE095I	IEE7903D	67
IEE096I	IEE7903D	67
IEE101A	IEFSD569	56
IEE110I	IEE2903D	62
IEE111I	IEE2903D	62
IEE112I	IEE6503D	88
IEE113I	IEE6503D	88
IEE116A	IEE6603D	
IEE117A	IEE6503D	
IEE118I	IEE6503D	88
IEE119I	IEE6503D	88
IEE120I	IEESD563	59
	IEESD565	
IEE124I	IEEVRCTL	
	IEESD575	
IEE132I	IEEVRCTL	
IEE134I	IEEVRCTL	
IEE135I	IEEVMNT2	
IEE136I	IEE3503D	62
IEE137I	IEE6303D	78
IEE138I	IEESD571	
IEE139I	IEFSD569	56
IEF140I	IEEVCTI	56
IEE141A	IEECLCTX	114
IEE142I	IEECOCTX	114
IEE143I	IEECMCTX	114
IEE147I	IEE0303F	71
IEE150I	IGC5A07B	
IEE151E	IGC5N07B	128
	IGC5807B	129
	IGC5407B	
	IGC5607B	
	IGC5L07B	
IEE151I	IEE6703D	60
IEE153E	IGC5107B	125
IEE154I	IGC5F07B	
IEE155E	IGC5107B	
	IGC5A07B	
IEE156E	IGC5407B	128
	IGC5A07B	
IEE156I	IEE6303D	78
	IEE6403D	78
	IEE6703D	60
	IEE6803D	60
	IEE6903D	60
	IEE7503D	60, 76, 91
	IEE7603D	76, 91
	IEE7703D	60
	IEE7803D	60
IEE157E	IGC5D07B	129
	IGC5807B	129
	IGC5607B	
IEE158E	IGC5407B	
	IGC5607B	
	IGC5807B	
IEE158I	IEE6703D	60
IEE159E	IGC5207B	126
	IGC5907B	123
	IGC5K07B	123
	IGC5J07B	123
IEE159I	IGC5907B	129
IEE160I	IGC5O07B	128
	IGC5207B	126
	IGC5907B	123, 129
	IGC5T07B	
	IGC5307B	
IEE161I	IGC5A07B	
IEE162I	IGC5N07B	

Message	Generating Module	Fig. Number Section 3-
	IGC5S07B	
IEE163E	IGC5A07B	
IEE164I	IGC5F07B	
IEE170E	IGC5C07B	124
IEE171E	IGC5C07B	124
IEE191I	IEFVMA	
IEE192I	IEFVMA38	
	IEFVMD	
IEE229I	IEE4703D	101
IEE250I	IEEXEDNA	
IEE251I	IEERTE1	82
IEE252I	IEE8803D	63
IEE253I	IEE8803D	63
IEE254I	IEERTE	82
	IEERTE2	82
IEE255I	IEERTE	82
	IEELGON1	73
IEE298I	IEE9903D	104
IEE299I	IEE4403D	96
	IEE5703D	101
IEE300I	IEE4403D	100
IEE301I	IEE3703D	58
IEE302I	IEE1103D	95, 96
	IEE4603D	95
	IGF2403D	99
IEE303I	IEE1103D	96, 97
	IEE4603D	99
	IGF2403D	99
IEE304I	IEESD565	
IEE305I	IEE4503D	75, 90, 91
	IEE7503D	60, 76, 91
	IEESD571	59
	IEE9903D	104
	IEE0403D	
IEE306I	IEE9903D	104
	IEE4403D	100
	IEE4703D	101
	IEE3703D	58
	IEE5703D	101
IEE307I	IEE3303D	96, 100
	IEE3203D	96, 98, 100
	IEE1103D	95, 97
	IGF2503D	92
	IEE4303D	98
	IEE9903D	104
	IEEVSTAR	2, 3, 8, 11
	IEFSD160	2
	IEEVRCTL	11
	IEEVMNT1	12
	IEFOSC03	
	IEFVMD	
	IEE3703D	
	IGF2403D	99

Message	Generating Module	Fig. Number Section 3-
IEE308I	IEEVMNT1	12
	IEEVRCTL	11
	IEFSD160	2
	IEEVSTAR	2, 8, 11
	IEE0403D	91
	IEE3803D	65
	IEE9903D	104
	IEE1103D	95
	IEE3303D	96, 100
	IEE1903D	77, 89
	IEE4503D	75, 90, 91
	IEFOSC03	43
	IEE3703D	
	IEFAB405	
	IEFVMD	
IEE309I	IEE3303D	96
	IEE3203D	96, 98, 100
	IEE3803D	65
	IEE7203D	101
	IEE1403D	68
	IEE9903D	104
	IEE1103D	97
	IEE4403D	100
	IEE4703D	101
	IEFSD160	2
	IEEVSTAR	2, 3, 8, 12
	IEEVMNT1	12
	IEEVRCTL	11
	IEESD571	
	IGF2403D	99
IEE310I	IEE9903D	104
	IEE1403D	68
	IEERTE1	82
IEE311I	IEE3203D	96
	IEE1103D	95, 97
	IEE0403D	91
	IEE4503D	75, 91
	IEE1903D	77, 89
	IGF2503D	92, 93
	IEFOSC03	43
	IEE3703D	58
	IEFSD160	2
	IEEVSTAR	2, 3, 8, 11
	IEEVRCTL	11
	IEEVMNT1	12
	IEE9903D	
IEE312I	IEE9903D	104
	IEEVSND1	85, 86, 87
	IEESD562	59
	IEFOSC03	43
	IEE4403D	
IEE313I	IGF2503D	92, 93
	IEE4403D	96, 100

Message	Generating Module	Fig. Number Section 3-
	IEFSD569	56
	IEE4303D	98
	IEE4703D	101
	IEE1103D	
	IGF2403D	99
IEE314I	IEE1103D	
IEE317I	IEE3703D	58
IEE319I	IEE1603D	71, 102
IEE321I	IEESD575	
IEE322I	IEESD575	
IEE323I	IEESD575	59
	IEFOSC03	43
	IEFVMD	
IEE324I	IEE8703D	72
IEE328I	IEE8703D	72
IEE334I	IEE1403D	68
IEE335I	IEEVMNT1	
IEE336I	IEE8703D	72
IEE337I	IEELGON1	73
IEE338I	IEE4703D	101
IEE339I	IEE4303D	98
	IEE4403D	96, 100
	IEE5703D	101
	IEE7203D	101
IEE341I	IEE4503D	75, 90
	IEE9903D	104
IEE342I	IEE9903D	104
	IEE4503D	75, 90
	IEE3703D	58
	IGC5L07B	
IEE343A	IEFSD569	56
IEE344A	IEFSD569	56
IEE345I	IGF2503D	92, 93
	IEE0403D	
	IEE4403D	96
	IEE4303D	98
	IEE3303D	96, 100
	IEE6303D	78
	IEE7203D	101
	IEE7503D	60, 76, 91
	IEE7703D	60, 76
	IEE4703D	101
	IGF2403D	99
IEE346I	IEFVMA	
IEE347I	IEELGON1	73
	IEERTE1	82
	IEE3803D	65
IEE348I	IEEVCTI	
IEE349I	IEE4903D	100
	IEE7203D	101
IEE351I	IEESMFOI	56
	IEESMFIT	56
IEE352A	IEESMFI3	56
IEE353A	IEESMFI3	56
IEE354I	IEESMFI3	56
IEE355I	IEESMFI3	56
IEE356A	IEESMFI3	56
IEE357A	IEESMFI3	56
IEE358I	IEESMFOP	56
IEE360I	IEESMFOP	56, 134, 135
IEE361I	IEFSMFWT	56
IEE362A	IEESMFOP	56, 134, 135
IEE363I	IEFSMFAL	56
IEE364I	IEESMFWT	56
IEE372I	IEELGON1	73
IEE373I	IEELGON1	73
IEE374I	IEE8803D	63
IEE375I	IEE8903D	63
IEE376I	IGF2403D	99
IEE377I	IGF2403D	99
IEE378I	IEE1903D	77, 89
	IGF2403D	99
IEE379I	IEE1903D	77, 89
	IGF2403D	99
IEE380I	IGF2503D	92, 93
IEE381I	IGF2503D	92, 93
IEE382I	IGF2503D	92, 93
IEE392I	IEERTE	82
IEE393I	IEELIST	70
IEE394I	IEELIST	70
IEE396I	IEERTE1	82
IEE398I	IEERTE	82
IEE399I	IEERTE2	82
IEE402I	IEELIST	70
	IEE3803D	65
	IEE8703D	72
IEE411I	IEEVSND1	85-87
IEE412I	IEEVSND1	85-87
IEE413I	IEEVSND1	85-87
IEE414I	IEERTE2	82
IEE416I	IEERTE3	82
IEE417I	IEELGON1	73
IEE420I	IEELGON1	73
IEE428I	IEELGON1	73
IEE429I	IEELGON1	73
IEE430I	IEELGON1	73
IEE431I	IEELGON1	73
IEE432I	IEELGON1	73
IEE433I	IEELGON1	73
IEE434I	IEELGON1	73
IEE435I	IEELGON1	73
IEE436I	IEELGON1	73
IEE437I	IEE8703D	72
IEE438I	IEE8703D	72
IEE439I	IEELIST	70
IEE440I	IEELIST	70

Message	Generating Module	Fig. Number Section 3-
IEE441I	IEELIST	70
IEE442I	IEELIST	70
IEE443I	IEELGON1	73
IEE444I	IEE8803D	63
IEE445I	IEE8803D	63
IEE446I	IEE8803D	63
IEE447I	IEE8803D	63
IEE448I	IEE8803D	63
IEE449I	IEEVSND5	83-86
IEE464I	IEEVSND9	83-86
IEE465I	IEEVSND5	83-86
IEE466I	IEEVSND5	83-86
	IEEVSND8	83,85
IEE467I	IEEVSND1	85-87
IEE468I	IEEVSND5	83-86
IEE471I	IEERTE1	82
IEE472I	IEEVSND1	85-87
	IEEVSND8	83, 85
IEE473I	IEEVSEND	83-86
	IEEVSND1	85-87
	IEEVSND9	83-86
IEE474I	IEERTE	82
IEE450I	IGCU403D	64
IEE452I	IGCU103D	64
IEE453I	IGCU103D	64
IEE454I	IGCU103D	64
IEE455I	IGCU103D	64
IEE542I	IEEDFIN4	56, 61
IEE543I	IEEDFIN7	61
IEE544I	IEEDFIN6	56
	IEEDFIN7	61
IEE545A	IEE9803D	56
IEE546A	IEE9803D	56
IEE547A	IEE9803D	56
IEE548A	IEE9803D	56
IEE549A	IEE9803D	56
IEE550I	IEE9803D	56
IEE551I	IEE9803D	56
IEE552I	IEE9803D	56
IEE553I	IEE9803D	56
IEE555I	IEESD562	
IEE600I	IEE1A03D	80
IEE696I	IEE0403F	
IEE701I	IEE1A03D	80
IEE702I	IEE1A03D	80
IEE703I	IEE1A03D	80
IEE704I	IEE1A03D	80
IEE706I	IEE1403D	68, 94
IEE800I	IEE3503D	62
IEE801D	IEFSD569	56
IEE802A	IEEDFIN2	56, 61
	IEEDFIN7	61
IEE803A	IEEDFIN7	61
	IEEDFIN2	56, 61
IEE804I	IEEDFIN7	61
IEE805I	IEEDFIN6	61
	IEEDFIN4	61
IEE806I	IEEDFIN3	56, 61
IEE807A	IEFSD569	56
IEE807I	IEEDFIN2	56, 61
	IEEDFIN3	61
IEE808A	IEEDFIN3	56, 61
	IEEDFIN2	56, 61
IEE809I	IEEDFIN3	61
IEE810I	IEEDFIN3	56, 61
IEE811I	IEEDFIN3	61
IEE812I	IEEDFIN3	56, 61
IEE813I	IEE1903D	89
IEE814I	IEEDFIN2	56, 61
	IEEDFIN8	56
IEE815A	IEEDFIN2	56, 61
IEE816I	IEEDFIN3	56
	IEEDFIN7	61
IEE817I	IEEDFIN3	56
	IEEDFIN7	61
IEE818A	IEEDFIN2	56, 61
	IEEDFIN3	
IEE819I	IEEDFIN7	61
IEE820I	IEEDFIN3	61
IEE821I	IEE3503D	62
IEE822A	IEEDFIN3	61
IEE822I	IEEDFIN2	56
IEE823A	IEEDFIN2	61
IEE866I	IEEDFIN7	61
	IEEDFIN8	56
IEE869I	IEESD564	
IEE870I	IEESD564	
IEE921I	IEE7503D	76
	IEE7603D	76, 91
	IEE7703D	60
IEE924I	IEE6803D	60
	IEE6903D	60
IEE925I	IEE6303D	78
	IEE6803D	60
	IEE6903D	60
IEE926I	IEE6303D	78
	IEE6703D	60
	IEE6803D	60
	IEE7503D	60
	IEE7603D	91
	IEE7703D	60
IEE927I	IEE6703D	60
	IEE7803D	60
IEE928I	IEE6803D	60
	IEE6903D	60
IEE929I	IEE6903D	60

Message	Generating Module	Fig. Number Section 3-
IEE930I	IEE6403D	78
IEE931I	IEE6303D	78
	IEE6803D	60
	IEE6903D	60
	IEE7203D	101
	IEE7303D	100
IEF000I	IEEVRCTL	
IEF001I	IEEVRCTL	
IEF002I	IEEVRCTL	
IEF003I	IEEVRCTL	
IEF004I	IEFIIC	
IEF005E	IEFSD510	2-5, 7, 8
IEF006I	IEFSTDSC	
IEF007I	IEFSTDSC	
IEF008I	IEFDSOWR	7
IEF009I	IEFDSOWR	7
IEF010I	IEFDSOWR	7
IEF011I	IEFDSOCP	8
	IEFDSOSM	
IEF012I	IEFDSOCP	8
IEF013I	IEFMF263	
IEF014I	IEFMF263	
IEF018I	IEFPARMS	56
IEF019I	IEFPARMS	56
IEF020I	IEFPARMS	56
IEF021I	IEFPARMS	56
IEF022I	IEFPARMS	56
IEF023I	IEFPARMS	56
IEF024I	IEFPARMS	56
IEF025D	IEFMF263	
IEF025I	IEFPARMS	56
IEF026I	IEFPARMS	56
IEF027I	IEFPARMS	56
IEF028I	IEFPARMS	56
IEF029I	IEFPARMS	56
IEF030I	IEFPARMS	56
IEF031I	IEFPARMS	56
IEF032I	IEFPARMS	56
IEF033I	IEEVRCTL	
IEF034D	IEFOSC03	
IEF038I	IEFVSPL	
IEF039I	IEFVMC	
IEF041I	IEFVINA	20
IEF042I	IEFVDA	17, 19
IEF043I	IEFVMB	
	IEFVMC	
IEF044I	IEFVMA	39
IEF045I	IEFVMA	
	IEFVMC	
IEF046I	IEFVMA	
	IEFVMC	
IEF047I	IEFOSC03	
IEF048I	IEFOSC03	43
IEF049I	IEFOSC02	43
	IEFOSC01	43
IEF050I	IEFVEA	19
IEF051I	IEFPARMS	56
IEF052A	IEFQMJ01	
IEF053A	IEFINTQA	
IEF054A	IEFINTQA	
IEF055A	IEFWAMIN	56
IEF056A	IEFWAMIN	56
IEF057A	IEFWAMIN	56
IEF058N	IEFWAMIN	56
IEF059A	IEFWAMIN	56
IEF060A	IEFWAMIN	56
IEF061A	IEFWAMIN	56
IEF062W	IEFWAMIN	56
IEF063W	IEFWAMIN	56
IEF064A	IEFWAMIN	56
IEF065A	IEFWAMIN	56
IEF066A	IEFWAMIN	56
IEF067A	IEFWAMIN	56
IEF068A	IEFWAMIN	56
IEF069W	IEFWAMIN	56
IEF070I	IEFWAMIN	56
IEF071E	IEFWAA02	51
IEF072I	IEFWAA02	51
	IEFWAA03	
IEF073A	IEFWAA01	48, 50
	IEFWAA03	
IEF074A	IEFWAA01	48, 50
	IEFWAA03	
IEF075A	IEFWAA01	48, 50
	IEFWAA03	
IEF076A	IEFWAA01	48, 50
IEF077A	IEFWAA02	51
IEF078I	IEFVMC	41
	IEFVMA	
IEF081I	IEFSMIFC	41
	IEFVMB	
	IEFVMC	
IEF088I	IEEMB800	56
IEF094I	IEFWTP00	109
IEF095I	IEFWTP00	109
IEF127I	IEFXKIMP	28
IEF128I	IEFXKIMP	28
IEF129I	IEFXKIMP	28
IEF130I	IEFXKIMP	28
IEF131I	IEFXKIMP	28
IEF132I	IEFXKIMP	28
IEF133I	IEFXKIMP	28
IEF134I	IEFXKIMP	28
IEF135I	IEFXKIMP	28
IEF136I	IEFXKIMP	28
IEF137I	IEFXKIMP	28

Message	Generating Module	Fig. Number Section 3-
IEF138I	IEFXKIMP	28
IEF139I	IEFXKIMP	28
IEF140I	IEFXKIMP	28
IEF141I	IEFXKIMP	28
IEF143I	IEFXKIMP	28
IEF144I	IEFXKIMP	28
IEF145I	IEFXKIMP	28
IEF146I	IEFXKIMP	28
IEF161I	IEFVMA	
IEF165I	IEFVHM	14
	IEFVMC	41
IEF166I	IEFVMC	41
IEF183I	IEFSD518	
IEF184I	IEFSD518	
IEF201I	IEFVJIMP	32
IEF202I	IEFVKIMP	23
IEF203I	IEFSD160	2, 3, 8, 11, 12
IEF204I	IEFSD160	2, 2, 8, 11, 12
IEF205I	IEFSD160	2, 3, 8, 11, 12
IEF206I	IEFSD160	2, 3, 8, 12
IEF209I	IEFSD21Q	
	IEFSD518	
IEF210I	IEFVMLS1	23
	IEFVMLS6	24
IEF211I	IEFVMLS6	24
IEF212I	IEFVMLS6	24
IEF213I	IEFVMLS6	24
IEF214I	IEFVMLS6	24
IEF215I	IEFVMLS6	24
IEF216I	IEFVMLS6	24
IEF217I	IEFVMLS6	24
IEF218I	IEFVMLS6	24
IEF219I	IEFVMLS6	24
IEF220I	IEFVMLS6	24
IEF221I	IEFVMLS6	24
IEF225D	IEFRPREP	32
IEF228I	IEFRPREP	32
IEF230I	IEFXCSSS	23
IEF231I	IEFVMLS1	23
	IEFVMLS6	24
IEF232I	IEFVMLS6	24
	IEFVMLS1	23
IEF233A	IEFWD000	23, 26
IEF234E	IEFWD000	23, 26
IEF235I	IEFSD195	28
	IEFXJIMP	27
IEF236I	IEFXT002	23
IEF237I	IEFXT002	23
	IEFYPJB3	32
IEF238A	IEFXJIMP	27
	IEFSD195	28
	IEFXT002	
IEF238I	IEFSD195	27
IEF239I	IEFSD195	27
IEF240I	IEFXKIMP	28
	IEFSD195	28
	IEFXJIMP	27
	IEFXCSSS	23
IEF241I	IEFXJIMP	27
	IEFSD195	28
IEF242I	IEFYPJB3	32
IEF243E	IEFWEXTA	23
	IEFXCSSS	
IEF243I	IEFXCSSS	23
IEF244I	IEFSD195	28
	IEFXJIMP	27
IEF245I	IEFXJIMP	27
	IEFSD195	28
IEF246I	IEFXKIMP	28
IEF247I	IEFSD195	28
	IEFXJIMP	27
IEF248I	IEFXT002	23
IEF249I	IEFPRES	56
IEF250D	IEFPRES	56
IEF250I	IEFPRES	56
IEF251I	IEFXJIMP	27
	IEFSD195	28
	IEFXKIMP	28
IEF252I	IEFXKIMP	28
	IEFXT002	23
IEF253I	IEFXKIMP	28
IEF254I	IEFXKIMP	28
IEF255I	IEFXKIMP	28
IEF256I	IEFXKIMP	28
IEF257I	IEFXKIMP	28
IEF258I	IEFXKIMP	28
IEF260I	IEFXKIMP	28
IEF261I	IEFXKIMP	28
IEF262I	IEFXKIMP	28
IEF263I	IEFXKIMP	28
IEF264I	IEFXKIMP	28
IEF265I	IEFXKIMP	28
	IEFSD195	28
	IEFXJIMP	27
IEF266I	IEFXKIMP	
IEF267I	IEFXKIMP	28
IEF268I	IEFXKIMP	28
	IEFPRES	56
IEF269A	IEFPRES	56
IEF270D	IEFPRES	56
IEF271I	IEFPRES	56
IEF272I	IEFYNIMP	32
IEF273I	IEFXKIMP	28
IEF277I	IEFVMLS6	24
IEF278I	IEFXCSSS	23
IEF279I	IEFXCSSS	23

Message	Generating Module	Fig. Number Section 3-
IEF280E	IEFZGST2	
	IEFZGJB1	
IEF280I	IEFZGST2	
	IEFZGJB1	
IEF281I	IEFXCSSS	23
	IEFZGJB1	
	IEFZGST2	
IEF282I	IEFXCSSS	23
	IEFZGST2	
	IEFZGJB1	
IEF283I	IEFZGST1	
	IEFZGJB1	
IEF285I	IEFZGST1	
	IEFZGJB1	
IEF286I	IEFVMLS6	24
IEF287I	IEFZGST1	
	IEFZGJS1	
IEF288I	IEFPRTXX	
IEF292I	IEFZGJB1	
IEF294I	IEFVMLS6	24
IEF296I	IEFSD195	28
	IEFXJIMP	27
IEF298I	IEFSD31Q	33
IEF300I	IEFOSC03	
IEF301I	IEFOSC03	43
IEF307I	IEFOSC03	
IEF308I	IEFVMLS6	24
	IEFVMLS1	23
IEF309I	IEFVMLS1	23
	IEFVMLS6	24
IEF310I	IEFVMLS6	24
	IEFVMLS1	23
IEF311I	IEFOSC05	
IEF315I	IEFVMLS1	23
	IEFVMLS6	24
IEF317I	IEFSD162	
IEF318I	IEFVMLS6	24
IEF319I	IEFPRTXX	
IEF320I	IEFPRTXX	
IEF322I	IEFSD31Q	
IEF325I	IEFOSC05	
IEF326I	IEFOSC05	
IEF327I	IEFOSC05	
IEF328I	IEFOSC05	
IEF329I	IEFOSC05	
IEF331I	IEFOSC05	
IEF334I	IEFVMB	
IEF335I	IEEFVMC	
IEF340I	IEFOSC03	
IEF346I	IEFOSC01	
IEF349I	IEFOSC07	43
IEF352D	IEFOSC03	
IEF354I	IEFOSC03	43

Message	Generating Module	Fig. Number Section 3-
IEF357I	IEFOSC03	
IEF359D	IEFOSC03	
IEF370I	IEFXKIMP	28
IEF372I	IEFVMLS6	24
IEF373I	IEFSMFWI	
IEF374I	IEFSMFWI	
IEF375I	IEFSMFWI	
IEF376I	IEFSMFWI	
IEF382A	IEFOSC01	
IEF383A	IEFOSC01	
IEF385I	IEFSD263	7
	IEFDSOWR	7
IEF386I	IEFSD263	7
	IEFDSOWR	7
IEF388I	IEFXJIMP	27
	IEFSD195	28
IEF389I	IEFSD195	28
	IEFXJIMP	27
IEF390I	IEFSD518	
IEF391I	IEFDSOSM	10
IEF392I	IEFDSOSM	
IEF393I	IEFDSOCP	8
	IEFDSOSM	
IEF394I	IEFDSOCP	8
IEF395I	IEFSD263	7
	IEFDSOWR	7
IEF397I	IEFDSOSM	10
IEF398I	IEFDSOCP	8
IEF401W	IEFSD055	
IEF403I	IEFSD21Q	23
IEF404I	IEFSD31Q	
IEF406I	IEFVMA	
	IEFOSC03	
	IEFVMC	
IEF407I	IEFVMA	
	IEFOSC03	
	IEFVMC	
IEF408I	IEFVMB	
	IEFVMA	
	IEFOSC03	
IEF409I	IEESD563	
IEF413I	IEFVHQ	
IEF414I	IEFVMB	
IEF417I	IEFVHA	14, 16-21
	IEFVMB	41
	IEFVMC	
IEF419I	IEFVMB	41
	IEFVMC	
IEF420I	IEFSD305	35
IEF421I	IEFSD305	35
IEF425I	IEFQMJ01	54
IEF427I	IEFSD160	2, 3, 11, 12
IEF430I	IEFVHEC	18

Message	Generating Module	Fig. Number Section 3-
IEF431W	IEFSD300	35
IEF438I	IEFOSC07	43
IEF440I	IEFQMJ02	54
	IEFQMJ01	54
IEF441I	IEFSD305	35
	IEFSD300	
IEF442A	IEFSD304	
IEF443A	IEFSD304	
IEF450I	IEFYNIMP	32
IEF451I	IEFVJIMP	32
IEF452I	IEFSD21Q	23
IEF453I	IEFSD31Q	
IEF454I	IEFXKIMP	28
	IEFXT003	28
IEF491I	IEFAB407	24
IEF492I	IEFVMLS1	23
IEF493I	IEFVMLS1	23
IEF500I	IEFXV001	26
IEF501I	IEFXV001	26
IEF502I	IEFXV001	26
IEF503I	IEFXV001	26
IEF504A	IEFXV002	26
IEF505I	IEFXV001	26
IEF506I	IEFSD195	28
	IEFXT002	23
	IEFXJIMP	27
IEF507D	IEFWAD	
IEF507I	IEFWAD	
IEF508E	IEFQMJ01	
IEF508I	IEFXV001	26
IEF512I	IEFSD304	
IEF513I	IEFSD304	
IEF514I	IEFSD304	
IEF515I	IEFSD304	
IEF516I	IEFMF263	
IEF517I	IEFVDA	17, 19
IEF533A	IEFWEXTA	23
	IEFWD000	23, 26
IEF601I	IEFVHA	14, 16-21
	IEFVHCB	14-20
	IEFVHC	22
	IEFVFA	15-17, 19
IEF605I	IEFVHC	21
	IEFVHCB	14-20
IEF606I	IEFVDA	17, 18
IEF607I	IEFVHEC	18
IEF610I	IEFVHC	21
	IEFVHCB	14-21
	IEFVHA	14-21
IEF611I	IEFVHEB	15-17, 19
	IEFVHH	11
IEF612I	IEFVEA	16, 19
IEF613I	IEFVEA	16, 19

Message	Generating Module	Fig. Number Section 3-
IEF614I	IEFVEA	16, 19
IEF615I	IEFVEA	16, 19
IEF616I	IEFVFA	15-17, 19
IEF617I	IEFVDA	
IEF618I	IEFVFA	15-17, 19
	IEFVFB	19
IEF621I	IEFVHC	14-21
	IEFVHA	14
IEF622I	IERVFA	15-17, 19
IEF623I	IEFVFA	15-17, 19
	IEFVFB	19
IEF624I	IEFVFB	19
	IEFVFA	15-17, 19
	IEFVDA	17, 19
IEF625I	IEFVFA	15-17, 19
IEF626I	IEFVFA	15-17, 19
IEF627I	IEFVFA	15-17, 19
IEF628I	IEFVFA	15-17, 19
IEF629I	IEFVFA	15-17, 19
	IEFVHH	11
IEF630I	IEFVDA	17, 19
	IEFVFA	15-17, 19
	IEFVFB	19
IEF631I	IEFVDA	17, 19
IEF632I	IEFVDA	17, 19
	IEFVFA	15-17, 19
	IEFVEA	16, 19
	IEFVJA	15
IEF633I	IEFVJA	15
IEF634I	IEFVJA	15
IEF635I	IEFVFA	15-17, 19
IEF636I	IEFVDA	17, 19
IEF637I	IEFVEA	16, 19
	IEFVJA	15
IEF638I	IEFVEA	16, 19
	IEFVGT	
IEF639I	IEFVEA	16, 19
	IEFVJA	15
	IEFVGT	
IEF640I	IEFVFA	15-17, 19
	IEFVDA	17, 19
	IEFVGT	
	IEFVGK	
	IEFVGS	
IEF641I	IEFVGK	
IEF642I	IEFVDA	17, 19
	IEFVEA	16, 19
	IEFVJA	15
	IEFVFB	19
	IEFVGT	
IEF643I	IEFVDA	17, 19
	IEFVGT	
IEF644I	IEFVDA	17, 19

Message	Generating Module	Fig. Number Section 3-
	IEFVGT	
IEF645I	IEFVDA	17, 19
	IEFVEA	16, 19
	IEFVGS	
IEF646I	IEFVEA	16, 19
	IEFVDA	17, 19
	IEFVJA	15
IEF647I	IEFVDA	17, 19
	IEFVFB	19
	IEFVGT	
IEF648I	IEFVDA	17, 19
IEF649I	IEFVDA	17, 19
IEF650I	IEFVFA	15-17, 19
IEF651I	IEFVFA	15-17, 19
IEF652I	IEFVFA	15-17, 19
IEF653I	IEFVFB	19
IEF654I	IEFVDA	17, 19
IEF655I	IEFVDA	17
IEF657I	IEFVHH	16, 19
IEF658I	IEFVDA	17
	IEFVHCB	14-20
	IEFVHC	21
IEF659I	IEFVHEB	15-17, 19
IEF660I	IEFVHC	21
	IEFVHCB	14-20

Message	Generating Module	Fig. Number Section 3-
IEF661I	IEFVHEC	18
IEF662I	IEFVINA	20
IEF663I	IEFVINA	20
IEF665I	IEFVINA	20
IEF668I	IEFVHCB	14-20
IEF669I	IEFVDA	17, 19
IEF670I	IEFVDA	17, 19
	IEFVEA	19
IEF671I	IEFVDA	17, 19
	IEFVEA	19
IEF672I	IEFVDA	17, 19
IEF690I	IEFVDA	17, 19
IEF691I	IEFVDA	17, 19
IEF692I	IEFVDA	17, 19
IEF861I	IEFVMC	41
	IEFMF102	
IEF863I	IEFMF102	
IEF864D	IEFMF102	
IEF866E	IEFSD515	
IEF868I	IEFOSC04	43
IGF953I	IGF2603D	74
IGF955I	IGF2603D	74
IHJ007I	IEFVRR3	74
	IEFVRR2	74
	IEFVRR1	74
	IEFVRRC	74
	IEFDSDRP	

Return Codes

Module	Return Code	Meaning
IEECVML7		
	Register 15	
	4	The number of lines in the parameter list is equal to zero or greater than 10. The request is ignored or the message is terminated at the tenth line.
	8	The message ID passed cannot be matched with any existing unended MLWTO chain. The request is ignored.
	12	An error line type, a second C line, or an error label line was found. The message is terminated at that point.
	16	The request is made with routing code 11 (WTP). The request is ignored.
	20	An MLWTO request is made and the queue to "hardcopy only" bit is on in the MCSFLAGS field. The request is ignored.
IEEPSN	4	System task step included in multistep procedure
	IEEPSN Work Area	
	'80'	System task
	'40'	DSI
	'01'	DSO starting
	'02'	Generalized START
	'04'	Writer starting
	'08'	Reader starting
IEESMFAL	0	OK
	4	error
IEEVRFRX	0	Successful processing
	4	No match to key
IEEVROUT	'00'	No errors
	'80'	Unauthorized issuer of remote WTOR- Not supervisor of key 0 user
	'84'	QID invalid - too large
	'88'	Receiver not logged on
	'8C'	RTAM unable to find space for BPL
	'90'	RTAM unable to queue BPL
IEE0303D	8	Nonzero protect key
	8	Attempt to add CIB is rejected
	8	Both registers 0 and 1 are zero on entry
	4	Delete bit on and CSCB not on chain
	4	Invalid origin pointer (CIB only)
IEE0303F	0	Request successful
	'000278'	Output limit has been reached
	'0Cnnnn'	I/O error
	Register 1 SMF=FULL	

Module	Return Code	Meaning
IEFAB405	0	successful
	4	locate error
	8	queue error#
	12	mount CVOL
	16	GDG dataset
	20	GETMAIN error
	24	SIOT creation caused total SIOT's in step to exceed 255
IEFAB406	0	successful
	4	#
	8	queue error
	24	SIOT creation caused total SIOT's in step to exceed 255
IEFAB407	0	successful
	4	#
	8	queue error
	24	SIOT creation caused total SIOT's in step to exceed 255
IEFAB408	0	successful
	8	SWADS full or invalid request to IEFAB408
	12	I/O error

RETCODE field in parameter list for IEFAB405 contains one of the following return codes from Catalog Management:

8	qualified name not found
16	not lowest index
20	syntax error in name
24	permanent I/O error
28	TTR out of extent
32	invalid work area pointer
36	data set not found
44	Volume List too small, correct size not returned

#RETCODE field in parameter list for module contains one of the following return codes from IEFAB408:

8	SWADS full or invalid request to IEFAB408
12	I/O error

IEFAB416	0	successful
	4	unsuccessful due to no entry in TIOT for the Private Catalog
IEFAB417	0	successful
	4	hooking of control blocks unsuccessful
	8	unsuccessful OPEN of Private Catalog
IEFAB418	0	successful
	4	hooking of control blocks unsuccessful
	8	unsuccessful CLOSE of Private Catalog
IEFAB420	0	successful
	4	unsuccessful hooking of control blocks

Module	Return Code	Meaning
IEFACTRT	4	Do not write SMF record
	other	Write SMF record
	Register 15	
	4	Cancel remainder of job
	other	Continue processing
IEFDSDRP	0	Normal
IEFDSOAL	'00'	No errors
	'04'	No space
	'0C'	I/O error
IEFDSOWR	'00'	No errors
	'04'	I/O error on DSO device
	'0C'	I/O error on job queue or spool manager
IEFINTQA	'00'	Normal
	'04'	Invalid device type
	'08'	I/O error during format
	'12'	Unable to open data set or obtain device characteristics
IEFMCVOL	'04'	Required volume not mounted
IEFMF105	nonzero value in register 15	Queue management I/O error system code = X'0B0'
IEFMF106	'0B0'	Queue management I/O error
IEFMF263	'00'	Required 1:1 space was gotten
	'04'	Required 1:1 space is unavailable
IEFORMAT	'00'	Normal
	'12'	I/O error
IEFQMJ01	'00'	Normal
	'04'	Queue full
	'08'	SWADS full
	'12'	I/O error
	'16'	Allow a writer to start
	'20'	Allow an initiator to start
	'24'	Allow a writer and an initiator to start
IEFQMJ02	'00'	Normal return, request has been completed
	'04'	Queue full (table break-up request), or SWADS full (table break-up request)
	'0C'	I/O error
	'00'	Normal return, request has been completed.
	'04'	Queue full (table break-up request), or SWADS full (table break-up request)
	'06'	I/O error
IEFQMMAC	'00'	Normal
	'12'	Invalid function code in parameter list
	LCT	

Module	Return Code	Meaning
IEFRPREP	'00'	No restart
	'08'	Restart is to occur
IEFSD097	'00'	Wait posted normally
	'04'	Wait posted by CANCEL
	'08'	Cannot wait
IEFSD161	'0B0'	I/O error
IEFSD162	R13/0	Job cannot run in this partition
	R4/0	Allocation cannot be done
	R15/0	Queue manager I/O error. System code = X'0B0'
IEFSD164	'0B0'	I/O error
IEFSD165	'0B0'	I/O error
IEFSD166	'0B0'	I/O error
IEFSD168	'0B0'	I/O error
IEFSD195	'00'	Operator replied WAIT
	'04'	Operator replied CANCEL
	'08'	Operator replied NOSEP
IEFSD22Q	0	Step termination
IEFSD309	0	No error
	1	Error
IEFSD31Q	4	Job termination
	0	System restart only
IEFSD41Q	'00'	Successful allocation
	'04'	Unsuccessful allocation
IEFSD518	'0B0'	I/O error
	0	No additional DSO required
	4	Additional DSO is required
IEFSD519	0	Read-in of DER successful
	12	Queue management I/O error
	RPL	
IEFSMCLD	'0C025C'	Write I/O error
	'080264'	GETMAIN failure
	'08026C'	Job specified not active
	'0802A0'	Invalid JCM TTR
	'0802B0'	WAA RB, invalid op code
	'0802B4'	Spool volume identication not possible, WAA
	RPL	
IEFSMEND	'0C025C'	Write I/O error
	'080260'	Data set not found
	'0802A8'	ENDS issued for unopened file
	'0802AC'	Invalid LRCBPTBF address
	'040000'	Invalid control table pointer

Module	Return Code	Meaning
IEFSMFAT	0	No SMF
	address	Pointer to TCT
	RPL - byte 1	
IEFSMGET	'00'	Request executed
	'04'	Invalid RPL
	'08'	Logical error or end of file
	'0C'	Uncorrectable I/O error
	RPL - bytes 2 and 3	
	'0000'	No error
	'0258'	I/O error on read
	'0270'	Invalid request
	'028C'	End of file
	'0290'	User buffer too short
	'0294'	Invalid point argument
	RPL	
IEFSMIFC	'040000'	Invalid RPL
	RPL	
IEFSMODS	'040000'	Invalid control block pointer
	'0C0258'	Read I/O error
	'0C025C'	Writer I/O error
	'080260'	Data set not found
	'080264'	GETMAIN failure
	'080268'	Internal table overflow
	'080274'	No spool space available second ODS on a spooled data set was issued before ENDS for that data set was issued
	'0802A0'	Invalid JCM pointer
	'0802B0'	WAARB, invalid op code
	'0802B4'	Spool volume identification not possible, WAA
	'002B8'	Spool space critical
	'0002BC'	No data in data set
	RPL - byte 1	
IEFSMPUT	'00'	Request executed
	'04'	Invalid RPL
	'08'	Logical error or end of file
	'0C'	Uncorrectable I/O error
	RPL - bytes 2 & 3	
	'0000'	No error
	'0258'	Read I/O error
	'025C'	Write I/O error
	'0270'	Invalid request
	'0274'	No more spool space
	'0278'	Output limit reached
	'027C'	Output limit exceeded

Module	Return Code	Meaning
	'029C'	Work area area allocation error
	'02B8'	Spool space critical
	RPL	
IEFSMREP	'0C0258'	Read I/O error
	'0C025C'	Write I/O error
	'080260'	Data set not found
	'080264'	GETMAIN failure
	'080268'	Internal table overflow
	'08026C'	Job specified not active
	'080270'	Invalid request
	'080274'	No more spool space
	'080288'	Data set not opened
	'08029C'	Work area allocation error
	'0802A0'	Invalid JCM pointer
	'0802B0'	Invalid WAAXB op code
	'0802B4'	Spool volume identification impossible
	'0802B8'	Cylinder from spool reserve user only to start initiator
IEFSTDSC	'00'	Normal (R1 points to JCT)
	'04'	Job on hold queue
	'08'	Job to be canceled
IEFUIV	4	Cancel job
IEFUJI	4	Cancel job
IEFUJP	4	Cancel job
IEEUJV	4	Cancel job
IEFUSI	4	Cancel job
IEFUSO	0	Cancel job
	4	Continue the step for the number of accesses shown in register 1
IEFUTL	0	Cancel job
	4	Continue the step or job for the number of timer units shown in register 1
	Register 2	
IEFVHN	0	Successful operation
	4	JCL error
	8	Spool or queue manager I/O error
IEFVINB	0	Procedure not found in directory
IEFVINC	0	Successful return
	4	Already 15 in-line procedures
IEFVINE	0	No label
	1-8	Length of label
	12	Syntax error in label field

Module	Return Code	Meaning
IEFVMD	R15 = 0	No ATTACH to be done
	R15/0	Address of ATTACH module
IEFVRRC	0	Normal
	code/0	Uncorrectable I/O error. Message IHJ007I is issued.
IEFVRR1	'08'	I/O error
	'04'	Job not found
	'00'	Normal
IEFVRR2	'00'	Normal
	'04'	SCT not found
	'08'	I/O error
	'0C'	JCL error
IEFVRR3	'00'	Normal
	'04'	Go to interface with JES reader and interpreter
	'08'	I/O error
IEFVSDRD	'10'	Abend occurred before allocation was complete.
	'04'	No restart. Job termination complete.
	'0C'	No restart. TIOT I/O error.
	'14'	Unable to establish STAE exit.
	'00'	No restart
	'08'	Restart
	WAA parameter list - byte 14	
IEFWAALC	'08'	JCM TTRL invalid. TTRL is not within spool extents or the record number is greater than the number of records that can fit on track TT.
	'02'	Invalid op code
	'01'	Spool volume identification not possible--internal tables destroyed.
	'00'	Operation requested is successful
	WAA parameter list - bytes 14 & 15	
IEFWAA01	'4000'	GETMAIN not satisfied
	'0100'	Spool volume identification not possible--internal tables destroyed
	'0080'	JCM dictionary overflow. The total number of readers, writers, and initiators exceeds the sysgen specifications, or a system component is not closing the JCM
	'0040'	Spool space critical--continue only if starting a writer
	'0020'	No more spool space available
	'0000'	Operation requested is successful
IEFWAA02	'8000'	JCM TTRL is invalid. The TTRL is not within spool extents or the record on spool is not a valid JCM.
	'4000'	GETMAIN not satisfied

Module	Return Code	Meaning
	'2000'	JCM cannot be written because of bad track
	'1000'	JCM cannot be read because of bad track
	'0100'	Spool volume identification not possible--internal tables destroyed
	'0000'	Operation requested is successful
IEFWAA03	'8000'	JCM TTRL is invalid. The TTRL is not within spool extents or the record on spool is not a valid JCM
	'4000'	GETMAIN not satisfied
	'2000'	JCM cannot be written because of bad track
	'1000'	JCM cannot be read because of bad track
	'0800'	Specific cylinder requested not available for allocation
	'0100'	Spool volume identification not possible--internal tables destroyed.
	'0080'	JCM dictionary overflow. The total number of readers, writers, and initiators exceeds the sysgen specifications, or a system component is not closing the JCM
	'0000'	Operation requested is successful
IEFWAD	0	Normal exit
	4	Record too long for a track
	8	No space left in SYS1.ACCT
	12	SYS1.ACCT not found
	16	Permanent I/O error
	20	End of file not found
	24	Unit not found
	28	Device not resident
	ECB	
IEFWAMGR	'42000000'	Extent violation. Input TTR not within VCB TTR extents.
	'0F000000'	Load real address of set sector, search, TIC, or I/O CCW failed, or indirect address list has overflowed
	WAA work area - byte 2	
	'20'	Permanent I/O error
	WAA parameter list - bytes 14 & 15	
	Register 15	
IEFWAMAP	'00'	GETMAIN error
	nonzero	pointer to WAM appendages and I/O related tables
IEFWAMIN	'00'	successful operation
	nonzero	Error found. Return to ready.

Module	Return Code	Meaning
IEFWARIN	'00'	succesful operation
	nonzero	Error found. Return to ready.
IEFXDPTH	variable #	Number of data paths to device specified in input parameter list
IEFXVNSL	'00'	Serial number found
	nonzero	Serial number not found
IEFXV002	'00'	Successful read
	'04'	Read error
	'08'	Device not ready
IGF2403D	0	Successful
	4	TP device; cannot vary.
	8	Last path; cannot vary.
	12	Path does not exist.
	16	Invalid base address
	20	Received device; cannot vary.
	24	Only one path; cannot vary.
	RPLFDBK	
IGG019DF	'0600'	I/O error on GET request
	RPLFDBK	
IGG019DG	'0604'	I/O error on PUT request
	Register 15	
	'00'	Successful completion

Module	Return Code	Meaning
	'04'	Operation cancelled by operator. The character set image was not in the library, or the print chain/train was not available.
	'08'	Operation terminated by uncorrectable error in searching the directory of the system library.
	'0C'	Operation terminated by an uncorrectable I/O error while loading the UCS buffer.
	'10'	Operation terminated by an uncorrectable I/O error while attempting to display the character set for visual verification.
	'14'	Operation cancelled by operator. The wrong character set was displayed for visual verification.
	'18'	No operation was performed because (a) the data control block was not open, (b) the data control block was not valid for a sequential data set, (c) the SETPRT parameter list was not valid, or (d) the output device was not a UCS printer
	'1C'	No operation was performed due to an uncorrectable error in a previously initiated output operation. The error analysis (SYNAD) routine is entered when the next PUT or CHECK macro instruction is issued.
	RPLFDBK	
IGG019DH	'0600'	I/O error on GET request
	'0604'	I/O error on PUT request
IGG019MA	'00'	Normal return
	'04'	Channel busy
	'08'	Invalid relative line number
	'0C'	Invalid command
	'10'	All terminals skip bit on
	'14'	Line error
	'18'	No buffer available
	'1C'	No buffer pool
	'20'	No buffer routine

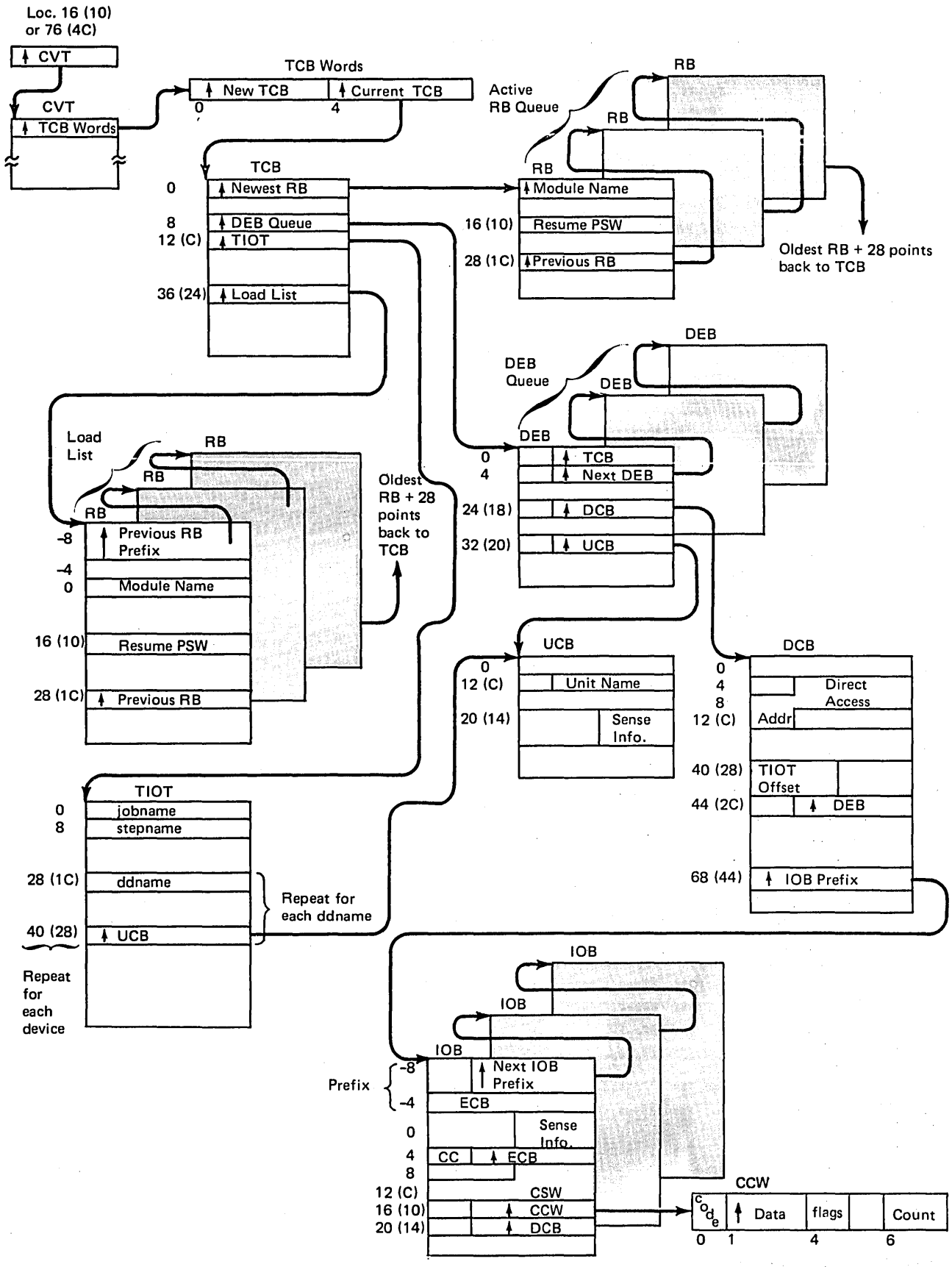


Figure 6-1. Control Block Flow and Relationship

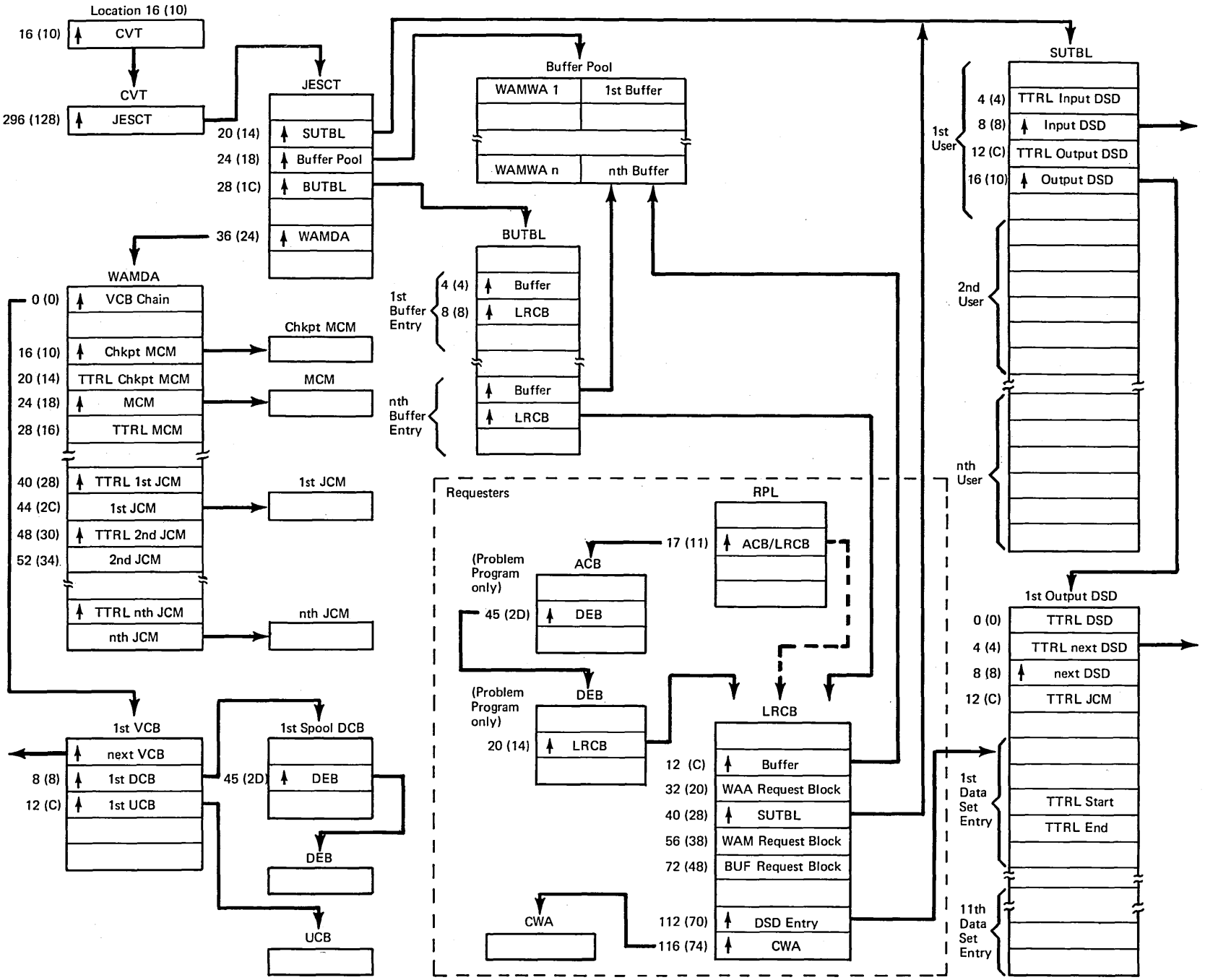


Figure 6-2. JECS Control Blocks

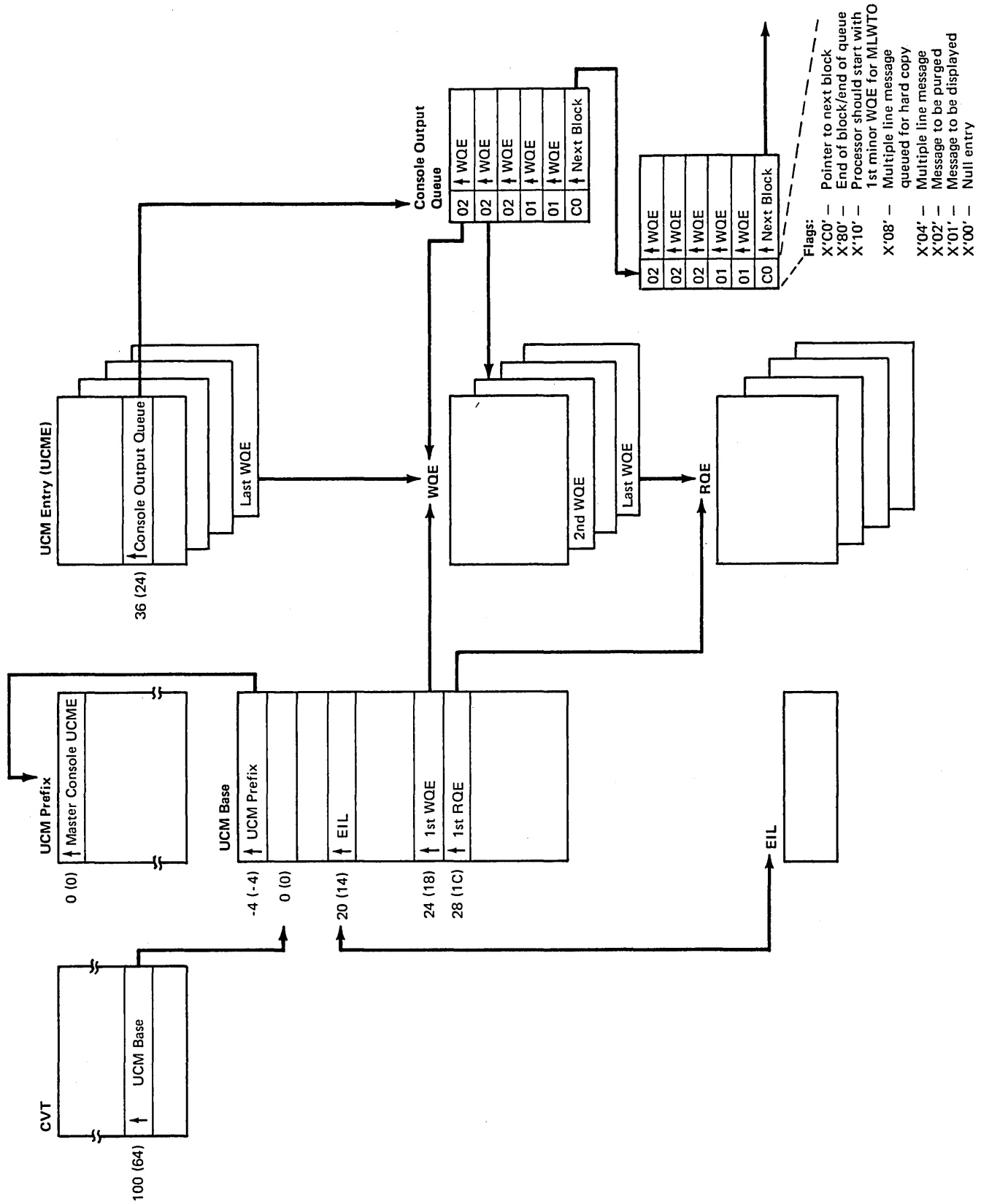


Figure 6-3. Communications Task Control Blocks

Appendix A. Dictionary of Abbreviations

ACB	access method control block
ACT	account control table
AT	alternate TIOT
AVR	automatic volume recognition
AVT	allocate volume table
AWT	allocate work table
BAQ	job queue UCB address
BPL	buffer parameter list
BSAM	basic sequential access method
BTAM	basic telecommunications access method
BUTBL	buffer usage table
CCW	channel command word
CDRPL	close data set request parameter list
CIB	command input buffer
CLT	channel load table
CMCM	checkpoint master cylinder map
COMTASK	communications task
CSCB	command scheduling control block
CVT	communications vector table
CWA	common work area
DADSM	direct access device space management
DASD	direct access storage device
DCB	data control block
DEB	data extent block
DER	disk entry record
DIDOCS	device independent display operator console support
DMT	device mask table
DNT	device name table
DQE	DOM queue element
DSB	data set block
DSD	data set dictionary
DSENQ	data set enqueue table
DSI	data set integrity
DSNT	data set name table
DSO	direct system output
DSOCB	direct system output control block
ECB	event control block
EDRPL	end data set request parameter list
EIL	event identification list
GMT	greenwich mean time
GTF	generalized trace facility
ICB	interrupt control block
IDA	indirect address list
IEFRPL	request parameter list
IEL	initiator entrance list
IOB	input/output block
IPL	initial program load
IQE	interruption queue element
IRB	interruption request block
IWA	interpreter work area
JAM	job entry subsystem access method
JCL	job control language
JCLS	job control language set
JCM	job cylinder map
JCT	job control table
JECS	job entry central services
JEPS	job entry peripheral services
JES	job entry subsystem
JESCT	job entry subsystem communication table
JFCB	job file control block
JFCBX	job file control block extension
JMR	job management record
JSCB	job step control block
LCA	log control area
LCT	linkage control table
LOT	life-of-task block
LRCB	logical record control block
LTH	logical track header
LTPC	logical time pseudo clock
LWA	local work area
MCM	master cylinder map
MCS	multiple console support
MSB	maintenance status block
MSI	master scheduler initialization
MSRDA	master scheduler resident data area
NIP	nucleus intialization program
NWE	no work element
ODRPL	open data set request parameter list
OLT	online test
ORE	operator reply element
PCCB	private catalog control block
PDQ	passed data set queue
PDT	parameter descriptor table
PIB	partition information block
PQA	protected queue area
QCR	queue control record
QMPA	queue manager parameter area
QMRES	queue manager resident area
QSAM	queued sequential access method
RCI	reversed compatibility interface
RCT	record control table

RES	remote entry subsystem
RMCM	resident master cylinder map
RMS	recovery management support
RPL	request parameter list
RPS	rotational position sensing
RSRPL	reposition data set request parameter list
RTAM	remote terminal access method
RVT	recovery vector table
SCCR	spool configuration control record
SCD	SYSOUT class directory
SCT	step control table
SDT	start descriptor table
SETLT	set parameter list
SIOT	step input/output table
SMCA	system management control table
SMF	system management facilities
SQS	system queue space
SUTBL	spool user table
SWADS	scheduler work area data set
TCB	task control block
TCE	TIOT chain element
TCT	timing control table
TCTIOT	timing control task input/output table
TIOT	task input/output table
TQE	timer queue element
TRCB	transient routine control block
TSCB	time slice control block
UCB	unit control block
UCM	unit control module
VCB	volume control block
VOLT	volume table
VTOC	volume table of contents
WAA	work area allocation
WAACH	checkpoint work area allocation param
WAADL	delete work area allocation parameter list
WAAL	work area allocation parameter list
WAAIO	I/O work area allocation parameter list
WAM	work area manager
WAMDA	work area manager data area
WAMPL	work area manager parameter list
WAMVT	work area manager vector table
WAMWA	work area manager work area
WQE	write queue element
WTL	write-to-log
WTO	write-to-programmer
WTOR	write-to-operator with reply
WTP	write-to-programmer
XSA	extended save area

Index

Indexes to OS/VS publications are consolidated in the *OS/VS Master Index*, GC28-0602, and the *OS/VS Master Index of Logic*, GY28-0603. These master indexes reference other publications that contain additional information about the subjects listed here.

OS/VS1 Job Management Logic (File No. S370-36) Printed in U.S.A. SY24-5161-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

OS/VS1
Job Management Logic
SY24-5161-1

READER'S COMMENT FORM

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. *For more direct handling of such request, please contact your IBM representative or the IBM Branch Office serving your locality.*

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

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

Cut or Fold Along Line

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

First Class
Permit 170
Endicott
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____

Department _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Cut or Fold Along Line

OS/VS1 Job Management Logic (File No. S370-36) Printed in U.S.A. SY24-5161-1