SY26-3827-2
File No. S370-30

Systems

OS/VS2 Open/Close/EOV Logic

Release 3.7

Includes the Following Selectable Units:

Supervisor Performance #2	VS2.03.807
Data Management Selectable Unit	VS2.03.808
3800 Printing Subsystem	VS2.03.810
JES3 MSS	5752–818
System Security Support	5752–832
Data Management Support	5752–860

PREFACE

This publication describes the internal logic of the Open, Close, and End-of-volume (O/C/EOV) components of the VS2 operating system. It is intended for people supporting these components who require detailed information about the functions performed, the data areas used, and the methods used to determine the nature and cause of problems encountered when executing the functions.

This book is divided into the following sections:

"Introduction." The introduction describes the relationships between the O/C/EOV components and the rest of the system control program. The introduction also summarizes the functions performed by the O/C/EOV components.

"Method of Operation." Method of operation diagrams (also called HIPO diagrams) describe the functions performed. The diagrams summarize the functions and the extended descriptions that accompany each diagram provide additional details and refer to the object modules executed during the performance of the function.

"Program Organization." Module flowcharts show flow of control among the O/C/EOV object modules by indicating the entries to and exits from each object module of the O/C/EOV components.

"Directory." A list of the O/C/EOV object modules is presented in alphameric sequence, along with cross-references to the diagrams and charts on which each object module is described or referred to. An object-module-to-load-module cross-reference is also provided.

"Data Areas." This section presents lists of the fields of key system control blocks modified by each object module, as well as description of the contents of certain key fields. A schematic diagram showing the relationships among the work areas and control blocks used by the O/C/EOV components is also included in this section.

"Diagnostic Aids." A cross-reference table of the abnormal termination (ABEND) system codes, the internal codes associated with each system code, the message associated with each system code, and informational messages is presented in this section. This section also presents lists of the O/C/EOV object modules executed under a number of typical conditions. How to use the problem determination and diagnostic routines is also described.

"Appendix A: Open and Close Processing of Spooled SYSIN/SYSOUT Data Sets." An appendix is provided to describe the subsystem open and close access method executors.

Prerequisite Reading

To effectively use this book, you should be familiar with the information contained in the reference manuals discussed in the following paragraphs.

OS/VS Data Management Services Guide, GC26-3783, summarizes the O/C/EOV processes, relative to the macros used, in the section "Opening and Closing a Data Set." To become familiar with the user's exit routines exercised during O/C/EOV processing (the standard user label routines, the DCB and EOV exit routines, the DCB ABEND exit routine, and the block-count routine) see the section "Exits to Special Processing Routines."

You should be familiar with the options of the OPEN, CLOSE, and FEOV macro instructions as described in *OS/VS Data Management Macro Instructions*, GC26-3793, and with the EOV, RDJFCB, OPEN TYPE=J, and DEBCHK macros described in *OS/VS2 System Programming Library: Data Management*, GC26-3830.

You should also be familiar with *OS/VS Tape Labels*, GC26-3795 which describes (1) the contents of standard (SL and AL) labels, (2) the purpose and logic of standard and nonstandard label processing routines, as well as the processing of nonlabeled (NL) tapes and those for which you specify that label processing be bypassed (BLP), and (3) the options available in tape processing.

You should understand how the job control language options affect the processing provided by O/C/EOV (concatenating input data sets, overiding label processing, etc.), as described in *OS/VS2 JCL,* GC28-0692.

If you are interested in debugging an O/C/EOV problem, you should be familiar with OS/VS2 *System Programming Library: Debugging Handbook,* Volume 1, GC28-0708, and Volume 2, GC28-0709 (both volumes can be ordered as GBOF-8211), and *OS/VS2 System Programming Library: Service Aids,* GC28-0674. The *Debugging Handbook* includes descriptions of the contents of system work areas and control blocks, such as the DCB, the DEB, and the DSAB. The O/C/EOV main work area and the where-to go table are included in *OS/VS2 Data Areas,* SYB8-0606 (available only on microfiche). The *OS/VS Message Library: VS2 System Messages*, GC38-1002, and *OS/VS Message Library: VS2 System Codes*, GC38-1008, are needed to determine the cause of failures associated with specific messages and codes.

If you want to determine the functions provided by the O/C/EOV-DADSM recovery routines, you should understand the system recovery concept, presented in *OS/VS2 System Programming Library: Supervisor,* GC28-0628.

Related Reading

Information on the components of the operating system that transfer control to and receive control from O/C/EOV appears in the following publications:

- *OS/VS2 BDAM Logic*, SY26-3831
- *OS/VS BTAM Logic*, SY27-7246
- *OS/VS2 DADSM Logic*, SY26-3828
- *OS/VS2 System Logic Library,* Volumes 1 through 7, SY28-0713 through SY28-0719 (all seven volumes can be ordered as SBOF-8210)
- *OS/VS Graphics Access Method Logic*, SY27-7240
- *OS/VS2 ISAM Logic*, SY26-3833
- *OS/VS2 SAM Logic*, SY26-3832
- *OS/VS TCAM Logic*, SY30-2049
- *OS/VS2 Virtual Storage Access Method (VSAM) Logic*, SY26-3825
- *OS/VS2 I/O Supervisor Logic*, SY26-3823
- *OS/VS2 VTAM Logic,* SY28-0621
- *OS/VS2 VIO Logic*, SY26-3834

Selectable Unit Prerequisites

- MVS Supervisor Performance #2, VS2.03.807, SN26-0823 TNL is prerequisite to SN26-0841
- Data Management Selectable Unit VS2.03.808, SN26-0823 TNL is prerequisite to SN26-0843
- 3800 Printing Subsystem, VS2.03.810, TNL SN26-0823 is prerequisite to SN26-0846.
- JES3 MSS, 5752-818, TNL SN26-0823 is prerequisite to SY26-3865.
- System Security Support, 5752-832, TNL SN26-0823 and Selectable Unit Newsletters SN26-0841 and SN26-0843 are prerequisites to SY26-3870.

CONTENTS

ILLUSTRATIONS

Figures

Diagrams

Charts

SUMMARY OF AMENDMENTS

Document Changes

August, 1978

This revision incorporates the information formerly contained in technical newsletter (numbered SN26-0823), Selectable Unit Newsletters (numbered SN26-0841, SN26-0843, and SN26-0846), and Selectable Unit Supplements (numbered SY26-3865, SY26-3870, and SD26-6007).

OS/VS2 MVS Data Management Support Selectable Unit (5752-860)

New Programming Support

A common pass/bypass function is provided in SCRATCH and RENAME. HSM will use this function while copying and restoring data sets to migration storage from public storage and vice versa.

OS/VS2 MVS System Security Support Selectable Unit (5752-832)

New Programming Support

The Resource Access Control Facility (RACF) supports RACF-protected nondirect-access data sets.

OS/VS2 MVS JES3 3850 MSS Selectable Unit (5752-818)

New Programming Support

The IBM 3850 Mass Storage System (MSS) is supported with this selectable unit (SU).

OS/VS2 MVS Data Management Selectable Unit (VS2.03.808)

New Programming Support

Open/Close/Eov supports DASD data sets protected by the Resource Access Control Facility (RACF) by controlling access for input and output to these data sets.

3800 Printing Subsystem Selectable Unit (VS2.03.810)

New Programming Support

The IBM 3800 Printing Subsystem is supported with this selectable unit. Information on the 3800 is provided for planning purposes only, until the product is available.

Supervisor Performance No. 2 Selectable Unit (VS2.03.807)

New Programming Support

Supervisor performance enhancements have been made to some of the Open/Close/EOV modules.

Release 3.7

Technical Changes

Improvements have been made to the description of standard-label tape processing.

Look-ahead mounting is bypassed for MSS volumes.

Release 3

New Programming Support

The IBM 3850 Mass Storage System (MSS) is supported with this release. Information on MSS is provided for planning purposes only until the system is available.

Other Technical Changes

VTAM (Virtual Telecommunications Access Method) is now fully supported by the Open/Close/EOV modules.

A new module, IFG0558X, has been added to take a checkpoint if checkpoint at EOV has been specified.

Release 2

Data Set Security Improvements

In general, the O/C/EOV routines were modified for this release to ensure that these routines process according to their original design and at the same time ensure system integrity and data set security.

Several data set security problems that existed in earlier VS systems have been eliminated in this release:

- Label processing and the attendant data set protection could be inadvertently or maliciously bypassed in certain situations in earlier releases. Label processing can no longer be bypassed in these situations, unless bypass label processing is specified, which should be controlled by each installation's reader procedure.
- During Open processing in earlier releases, a complete direct-access volume might be searched when a format-1 DSCB is being sought. The search has been limited to the extent containing the VTOC for this release.
- In earlier releases, Open, Close, and EOV routines made modifications to DCBs and ACBs in the user's storage. For this release, the DCB is copied into storage with a system storage protection key, thereby averting malicious and inadvertent modifications to the DCB. This copy of the DCB is used only during O/C/EOV processing. Each time an exit is taken to an exit routine provided by the problem program, the copy is written over the DCB in user storage. At the conclusion of O/C/EOV processing, the copy of the DCB in protected storage is freed, before control is returned to the caller. The access method processing routines (READ, WRITE, GET, PUT, etc.) continue to use the DCB in user storage.
- In earlier releases, problem programs using the EXCP access technique could gain unauthorized access to data sets using the EXCP appendages loaded by the Open routines. Open now verifies that every appendage loaded is authorized by the installation.
- Earlier releases inadvertently allowed simultaneous opening and scratching of a data set by different tasks under certain circumstances. It is no longer possible to scratch an open data set.
- In earlier releases, data sets could be read or written by user exit routines during volume switching in EOV and during volume disposition processing in Close and TCLOSE. It was possible to use the DEB associated with the data set for which volumes were being switched or repositioned. In this release, attempts by a user to initiate I/O using this DEB will result in abnormal termination of the user's task (system code of 16E).

Storage Protection Key Implementation

The VS2 release 2 system control program uses storage protection keys to protect sensitive data from the user. The storage protection key designation is keys 0 through 7 for system use and keys 8 through 15 for programs executing in problem program state.

The O/C/EOV routines execute using a PSW that has a storage protection key of 5, called *data management key*. O/C/EOV routines obtain virtual storage for work areas and some control blocks (for example, the DEB and the DEB extension) in key 5 storage.

System Recovery Routines

Recovery routines have been added to the O/C/EOV components to ensure that the system control program does not fail when errors occur in the O/C/EOV and DADSM components. These routines are entered under all abnormal termination conditions encountered in the O/C/EOV components. Their function is to release all storage obtained and free resources acquired during O/C/EOV processing and to attempt to restore the DCBs and ACBs to closed status.

Resource Manager Routine (Task Close)

A data management resource manager called *Task Close* has been added to the O/C/EOV components to close open data sets and to release all resources held in connection with these data sets. This resource manager is called by the termination manager during task and address space termination, normal and abnormal.

Opening and Closing VIO Data Sets

Changes have been made to the Open and Close routines to process virtual input/output (VIO) data sets. These changes involve using the equivalent of a format-1 DSCB that resides in the scheduler work area (SWA) in operations formerly using a format-1 DSCB residing in a volume table of contents.

Dynamic Deallocation of Data Sets

Changes were made to the Close component to allow problem programs to release data sets and devices dynamically at the time the data set is closed, rather than releasing them normally at task termination. This requires the use of a new system control block, the DSAB, throughout O/C/EOV processing, and further requires that the data extent block (DEB) be extended. The DEB extension is acquired and initialized during Open processing.

VTAM Data Sets

Data sets using the virtual telecommunications access method (VTAM) can be created and processed under VS2. O/C/EOV components recognize the VTAM data set organization and transfer control to the VTAM Open and Close routines. VTAM is not included in the initial release of OS/VS2. The VTAM information is provided here for planning purposes.

Job Entry Subsystem Support

Open and Close executors have been added in this release to establish an interface with the job entry subsystems, JES2 and JES3.

In OS MVT releases and in VS2 release 1, SYSIN and SYSOUT data sets were created by the reader-interpreter. When opened by the user, these data sets were accessed like any other sequential data set on a direct-access device. Open and Close invoked SAM executors and EOV extended SYSOUT data sets as if they were any other direct-access data set.

In VS1 and in this release of VS2, the job entry subsystem uses the ACB-RPL interface, which requires opening and closing both the user's DCB and an ACB for the spooled (SYSIN/SYSOUT) data set. The DCB is used as a compatibility interface between the job entry subsystem and the user's program, and the ACB is used by the job entry subsystem in reading and writing the spooled data sets.

In addition, in VS2 release 2, ACBs for SYSIN and SYSOUT data sets are opened and closed using subsystem executors to establish the interface the job entry subsystems uses to read and write the spooled data sets. These executors are documented in the appendix of this manual.

Performance Improvements: Packaging and Related Changes

The O/C/EOV object modules have been packaged in large load modules. This was done so the XCTL macro instructions (SVC 7) used by the O/C/EOV routines could be replaced by branch instructions, thereby improving performance. The GETMAIN and FREEMAIN macro instructions (SVC 10) issued by the O/C/EOV routines were replaced by linkage for branch entry to the system routines that obtain and free virtual storage. Note that SVC 7 and 10 will no longer appear in the SVC trace table entries resulting from execution of the O/C/EOV routines.

Note also that the informational messages (mount and demount messages, etc.) are now issued using only one multiple-line WTO macro instruction, instead of two single-line WTO macro instructions.

INTRODUCTION

The Open, Close, and End-of-volume (O/C/EOV) routines are type 4 supervisor call (SVC) routines that reside in SYS1.LPALIB. All members on SYS1.LPALIB are made resident in virtual storage in the pageable portion of the link pack area (PLPA) when the system is initialized.

A problem or system program executes the Open, Close, and EOV routines by issuing a macro instruction (for example, OPEN, CLOSE, EOV, or FEOV) whose expansion includes an SVC instruction. Execution of the SVC instruction causes control to be passed to the SVC second-level interruption handler (SLIH), which passes control to the appropriate SVC routine. Figure 1 illustrates the relationship between the programs that issue the O/C/EOV SVCs, the interruption handler of the supervisor, and the system services requested by the O/C/EOV routines.

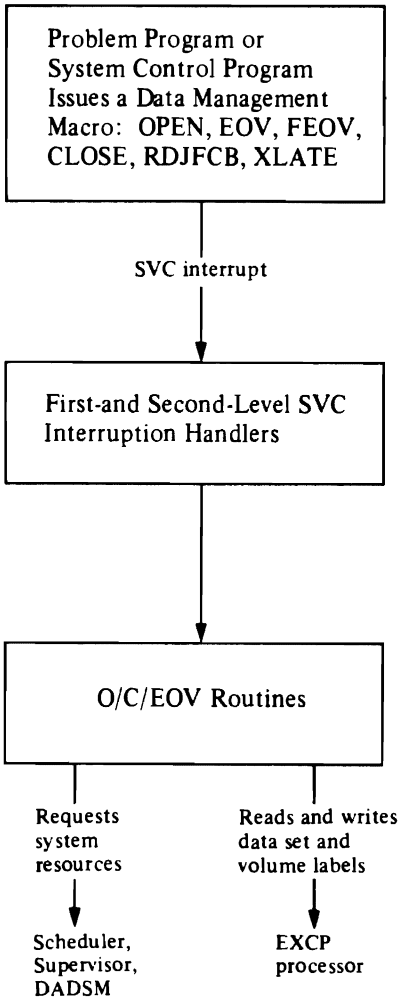

Problem Program or System Control Program Issues a Data Management Macro: OPEN, EOV, FEOV, CLOSE, RDJFCB, XLATE
SVC interrupt
First-and Second-Level SVC Interruption Handlers
O/C/EOV Routines
Requests system resources
Reads and writes data set and volume labels
Scheduler, Supervisor, DADSM
EXCP processor

Figure 1. The O/C/EOV Routines Are Invoked by the SVC Interrupt Handler and, in Turn, Call Other System Control Program Routines

Opening a Data Set

A data set is opened as a result of the issuance of the OPEN macro which causes the supervisor call (SVC) second-level interrupt handler (SLIH) to pass control to the Open component (SVC 19 or SVC 22 if OPEN TYPE=J is issued).

The Open component (1) completes and modifies the fields of the DCB or ACB, the UCB, etc., (2) makes sure that the volume containing or to contain the data set is available and ready, and (3) reads and writes volume and data set labels associated with the data set. Open also calls access method open executors to build a DEB. At the conclusion of Open processing, the data set will be ready to be read from or written to by the access method processing routines. These access method processing routines will be ready to respond to READ, WRITE, GET, and PUT requests and any other I/O options specified in the DCB or ACB.

Closing a Data Set

A data set is closed as a result of the issuance of the CLOSE macro instruction, which causes the SVC second-level interrupt handler (SLIH) to pass control to the Close component. The Close component restores the fields of the DCB or ACB so that they are the same as before the data set was opened. Labels are read and written during Close processing and storage acquired during Open processing, including the DEB and I/O buffers, is released. Close is invoked via SVC 20.

BSAM and VSAM data sets can be temporarily closed. This involves all of the operations performed by Close, except that the DCB or ACB is not restored and the DEB is not released. The processing program can resume reading or writing the data set without reopening it. The temporary close routines (called *TCLOSE*) are called by the SVC SLIH when a CLOSE TYPE=T macro is issued, which results in control being passed to the SVC 23 routines.

Data Set Organization

Relatively few of the operations described in this book are performed for data sets with the VSAM organization or with the data set organizations associated with teleprocessing (BTAM, TCAM, and VTAM). Instead, the Open and Close routines recognize these data set organizations and pass control to the VSAM and VTAM Open and Close routines (see *OS/VS2 Virtual Storage Access Method (VSAM) Logic* and *OS/VS2 VTAM Logic* for descriptions of these routines).

Device Type

The processing performed by the Open and Close components depends upon the type of device the data set being opened and closed resides on. SYSIN and SYSOUT data sets require very little processing since they reside on unit record devices before or after they are spooled by the job entry subsystem: no volume mounting is required, and since no labels are allowed, no label processing is necessary. Data sets on direct-access and magnetic tape volumes require the most processing by Open and Close. In the "Method of Operation" and "Program Organization" sections the Open and Close routines are grouped and presented based on the media on which the data set to be processed resides.

End-of-Volume (EOV) Processing

The primary function of the EOV component is to make it possible for problem programs to process multivolume sequential data sets without knowing when the end of one volume is reached and processing begun on the next volume.

Input Data Sets: When an input data set is being read and the access method READ or GET routine senses that end-of-data has been reached on a given volume (as described by the DEB), these routines issue the EOV macro instruction to determine whether the end of the data set has been reached or merely the end of a segment of the data set has been reached. EOV responds by (1) giving control to the user's EODAD routine if the end-of-data has been reached or (2) getting the next segment (volume or member of a concatenation) of the data set ready to process.

Output Data Sets: When an output data set is being written on a secondary-storage medium (direct-access or magnetic tape volume), and the access method PUT or WRITE routine senses that the end of the space on the volume has been reached, the PUT or WRITE routine issues the EOV macro instruction to get more space on which to finish writing the data set. When direct-access data sets are being written, the EOV routines will obtain space on the same volume, or, if necessary, on a different volume. Additional space for magnetic tape data sets is always obtained on a new volume.

Concatenating Input Data Sets: The EOV component also makes it possible to read two or more data sets using the same DCB or ACB. When end-of-data is reached on the first data set, the EOV component makes sure that the volume containing the next data set to be read is mounted, verifies the data set labels, and positions the volume to be read.

Data sets on different media (e.g., magnetic tape and direct-access volumes) can also be concatenated using the EOV functions. These kinds of concatenations are referred to as *concatenation of data sets with unlike characteristics* and include any data sets that require different DCBs, IOBs, or channel programs. The concatenation is achieved by closing the DCB and reopening it. EOV calls the Close component which, when complete, calls Open. All SYSIN data sets are processed as if they had unlike characteristics.

The EOV component is invoked by the SVC SLIH as a result of the issuance of either the EOV (SVC 55) or FEOV (SVC 31) macro instruction.

As with Open and Close, very little processing is done for VSAM EOV requests and EOV processing is not supported for the data set organizations associated with teleprocessing, BTAM, TCAM, and VTAM. EOV immediately gives control to the VSAM EOV routines, which do not return control when complete; instead VSAM EOV returns to the VSAM access method routine that issued the EOV macro.

Problem Determination

The Problem Determination routines gather information about abnormal termination conditions associated with determinate errors that arise during Open, Close, or EOV processing. The information (data area and control block contents) is gathered and written to the GTF data set. Procedures for invoking these routines appears in the "Diagnostic Aids" section.

System Recovery

Recovery routines are included with the O/C/EOV routines to ensure that a system failure does not result from an error encountered during O/C/EOV processing.

The purposes of these routines are (1) to release all resources acquired by the O/C/EOV routines in behalf of the failing task; (2) to attempt to return all DCBs and ACBs being processed to closed status (called *force closing* the data sets); and (3) to provide diagnostic information about the failure.

The O/C/EOV recovery routines are invoked by the recovery/termination manager (R/TM) via the extended STAE (ESTAE) interface, as described in *OS/VS2 System Programming Library: Supervisor.*

The DADSM routines use the same recovery routines to release resources and provide diagnostic information.

Data Management Resource Manager (Task Close)

Task Close is called by the recovery/termination manager (R/TM) to free all of the system resources associated with open data sets which were acquired on behalf of a terminating task. Task Close is called during normal and abnormal termination of both tasks and address spaces. Although the functions performed depend on the conditions under which Task Close was called, the operations usually performed are to close DCBs and ACBs and pass control to access method resource management routines so they can do likewise.

METHOD OF OPERATION

This section consists of method of operation diagrams, prefaced with information about how to read the diagrams. The paragraphs that follow describe the functions of the O/C/EOV-DADSM service routine, IFG019RA. This description is included to provide general information about how the O/C/EOV routines use the service routine, which is necessary in understanding how the O/C/EOV routines do their job.

The O/C/EOV-DADSM Service Routine

The IECRES macro is issued by the O/C/EOV and DADSM modules to link to the O/C/EOV-DADSM service routine, IFG019RA. The service routine provides several functions:

- Transferring control to achieve parallel processing of DCBs and ACBs during Open and Close processing, and overlapping of I/O operations with other O/C/EOV processing.
- Getting and chaining work areas.
- Establishing the recovery routine address to receive control if an abnormal termination condition arises during O/C/EOV or DADSM processing.
- Copying DCBs and ACBs from the user's storage to protected storage and from protected storage back to the user's storage.

The functions performed by the service routine depend on the operands coded in the IECRES macro. Some of the operands used most often are discussed below; Diagrams 5.1 and 5.2 describe the service routine. Note that IFG019RA is sometimes referred to in the source code as the *resident routine*.

Parallel Processing of DCBs and ACBs and Overlapping I/O

More than one DCB/ACB address may be coded in an OPEN or CLOSE macro instruction. To provide for parallel processing of the DCBs and ACBs during such a concurrent open or close, the service routine (IFG019RA) synchronizes the processing of the control blocks and transfers control in a manner that allows each DCB or ACB to be processed by each module. The SYNCH and LOAD operands of the IECRES macro are issued by the O/C/EOV modules to achieve parallel processing. The service routine uses the module names passed in the WTG table to determine which O/C/EOV module is to receive control next.

The WAIT operand of the IECRES macro is used by the Open and Close modules to overlap I/O with concurrent processing of other DCBs or ACBs in a parallel open or close. When I/O is required, for example, to read a DSCB, the module that issues the EXCP macro to initiate the I/O follows the EXCP macro with an IECRES WAIT instruction. This allows continued processing of any other DCB or ACB being processed in parallel.

Getting and Chaining Work Areas

The GET operand of the IECRES macro is issued to link to the service routine to obtain virtual storage for the work areas used by the O/C/EOV and DADSM routines. More than one work area is required for many of the functions performed, and the work areas are obtained, as needed, at different stages of the processing. To make it easy to locate and release the work areas, the service routine forward-and-backward chains all of the work areas obtained for a given request.

Establishing the Recovery Routine

At the same time the first block of virtual storage is obtained for a given O/C/EOV or DADSM request, the service routine issues an ESTAE macro to establish the address of the routine that is to get control from the recovery/termination manager (R/TM) if an abnormal termination condition arises in subsequent processing. The service routine also ensures that the recovery routine is loaded.

Copying DCBs and ACBs

To avert malicious and inadvertent access to one user's data sets by another unauthorized user, the O/C/EOV routines process a copy of the user's DCB in protected storage. Another of the functions provided by the service routine is copying the user's DCB/ACB into protected storage during initialization. The service routine also copies the modified DCB back to the user's storage at the conclusion of processing (and whenever a user's exit routine is given control).

How to Use the Method of Operation Diagrams

Figure 2 is an index to help you find the diagrams or group of diagrams that you're interested in by name.

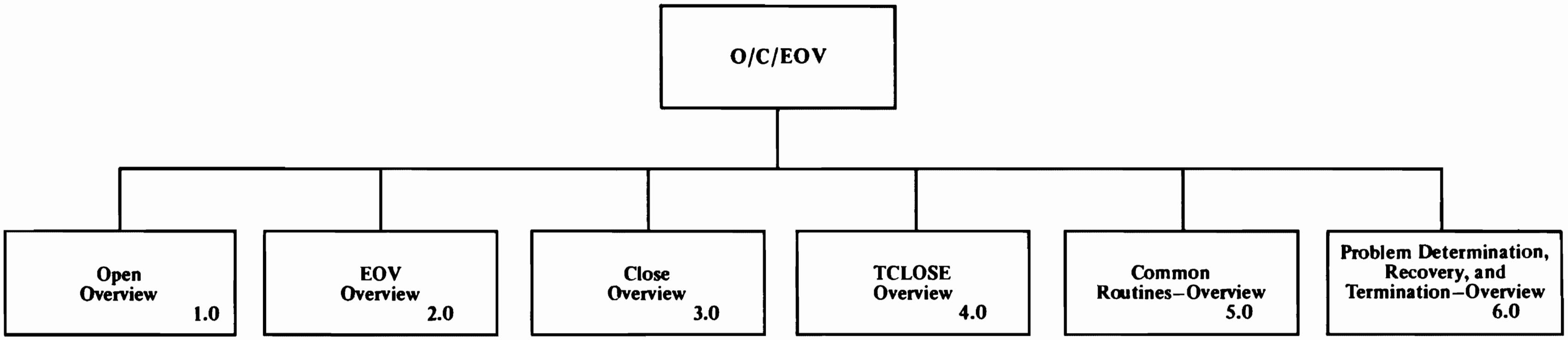

O/C/EOV
Open Overview 1.0
EOV Overview 2.0
Close Overview 3.0
TCLOSE Overview 4.0
Common Routines–Overview 5.0
Problem Determination, Recovery, and Termination–Overview 6.0

Figure 2. Contents: Method of Operation Diagrams

Overview Diagrams

Diagrams 1.0, 2.0, 3.0, 4.0, 5.0, and 6.0 relate input and output (on the left and right of each diagram, respectively) to the process steps in the center block of the overview diagram. The numbers on the process steps of these diagrams are the diagram numbers of subordinate diagrams. The processing steps in the overview diagrams summarize the processing that is described in the subordinate (bottom-level) diagram.

Reading the Bottom-Level Diagrams

These diagrams relate processing steps to the input and output they use. The input—control blocks, register contents, records read from I/O devices—is on the left. Input is what the processing steps use. The processing steps in the center block describe the function of a routine or series of routines in the source code. Each processing step is preceded by a number that is keyed to the extended description on the facing page. The processing block also shows flow of control using the on-page and off-page connectors described in the legend. The output of the processing steps consists of modified control blocks, register contents, and records written to I/O devices. The nature of the use of the input—for reference, virtual storage moved from one location to another, or records read into virtual storage from I/O devices—is determined from the type of data arrow used. The arrows and their meanings are described in the legend. Similarly, control block modification, register settings, and records written to I/O devices in the output block is determined from the arrow type associated with the output.

Reading the Extended Description

The extended descriptions provide more detail about a processing step. They also identify a processing step with the object modules and labels within these object modules where you can find the source code. Each processing step has a number that corresponds with a number in the extended description.

The bold characters preceding the paragraphs of the extended description—for example, **IGC0001I**—are the object module names (and microfiche names) of the source code modules in which you can find the implementation of the functions. The characters in brackets following the paragraphs of the extended description—for example, [OIN85000]—are labels in the source code.

Legend

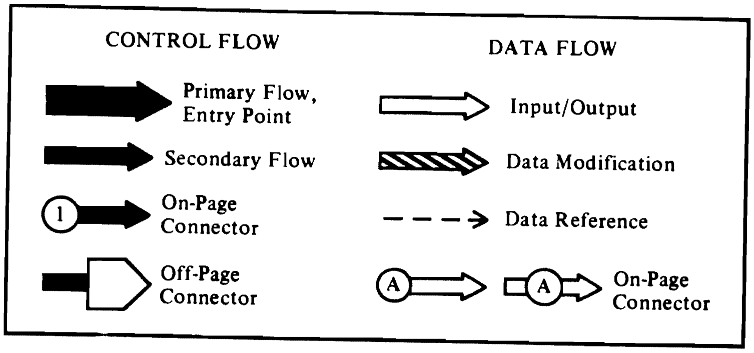
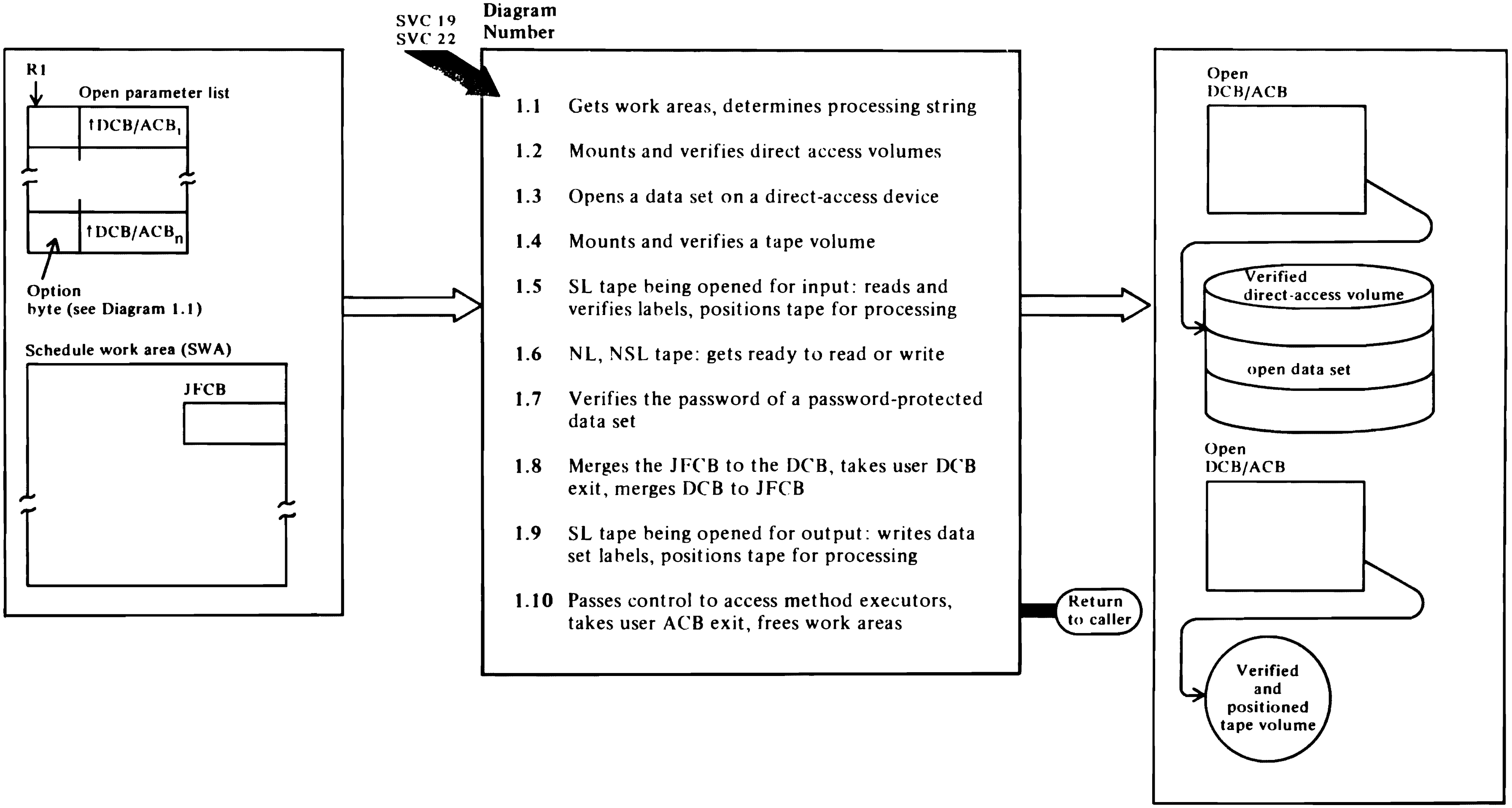


Diagram 1.0. Open Overview



1.1 Gets work areas, determines processing string

1.2 Mounts and verifies direct access volumes

1.3 Opens a data set on a direct-access device

1.4 Mounts and verifies a tape volume

1.5 SL tape being opened for input: reads and verifies labels, positions tape for processing

1.6 NL, NSL tape: gets ready to read or write

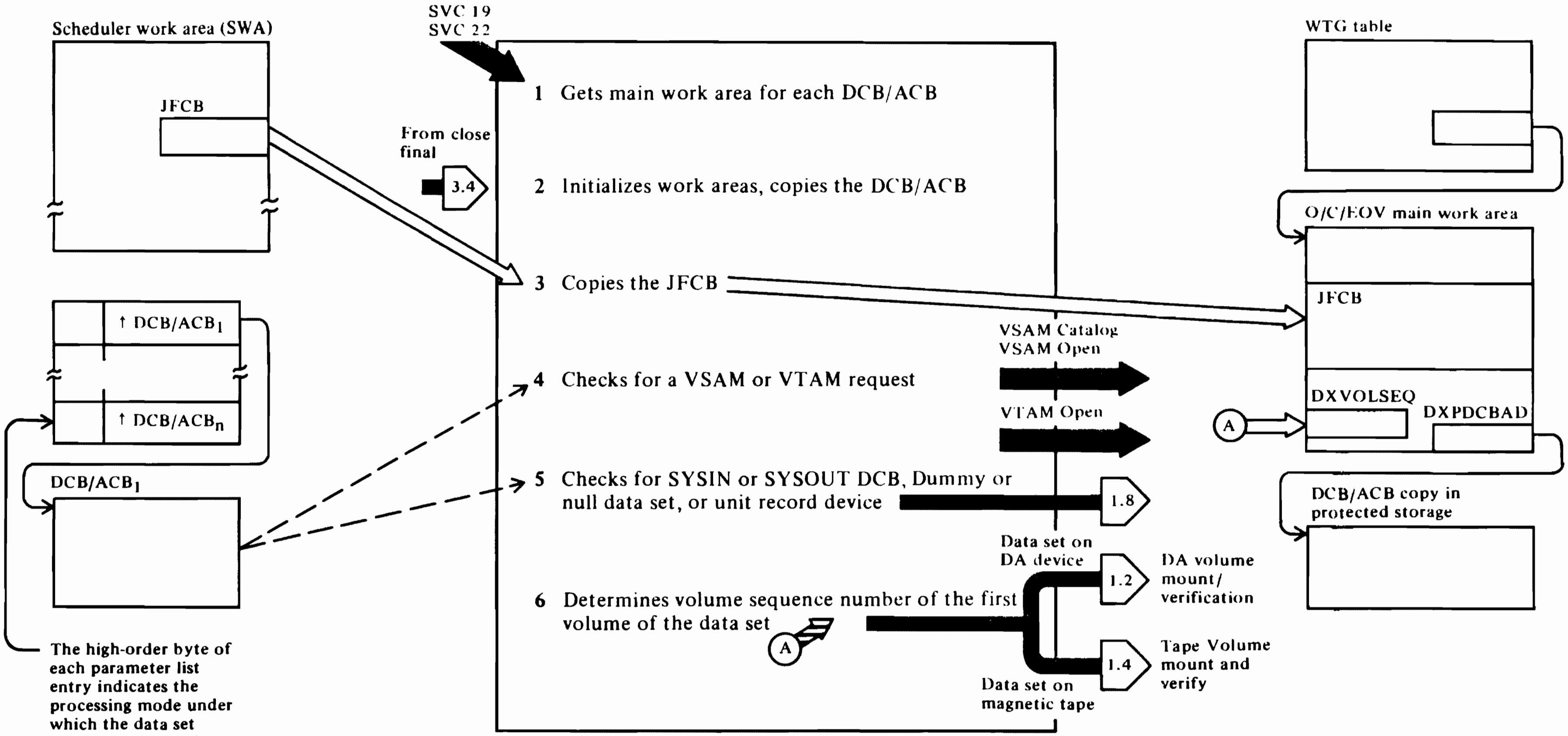
1.7 Verifies the password of a password-protected data set

1.8 Merges the JFCB to the DCB, takes user DCB exit, merges DCB to JFCB

1.9 SL tape being opened for output: writes data set labels, positions tape for processing

1.10 Passes control to access method executors, takes user ACB exit, frees work areas

Diagram 1.1. Open Initialization



1 Gets main work area for each DCB/ACB

2 Initializes work areas, copies the DCB/ACB

3 Copies the JFCB

4 Checks for a VSAM or VTAM request

5 Checks for SYSIN or SYSOUT DCB, Dummy or null data set, or unit record device

6 Determines volume sequence number of the first volume of the data set

The high-order byte of each parameter list entry indicates the processing mode under which the data set is to be opened. The valid options are:

..00....	DISP	0011	INOUT
..01....	REREAD	0111	OUTIN
..11....	LEAVE	0001	RDBACK
....0000	INPUT	0100	UPDAT
....1111	OUTPUT		

The last entry in the parameter list is indicated by the high-order bit being set to 1.

Diagram 1.1. Open Initialization

Steps 1, 2, and 3 can be entered either via SVC 19 or SVC 22, OPEN or OPEN, TYPE=J. These steps are repeated here to relate the functions to the modules and labels of both the OPEN and OPEN, TYPE=J source code.

IGC0001I—OPEN (SVC 19)

1 This function is entered from the SVC second-level interrupt handler (SLIH) via SVC 19 (the OPEN macro). [IGC0001I]

Virtual storage is gotten for the base prefix, WTG table, and associated work areas. The prefix and WTG table are initialized for this OPEN request. See "Data Area Relationships During O/C/EOV Processing" for an example of the work areas obtained and how they are chained during open processsing. See Diagram 5.1 for details concerning the establishment of work areas via the IECRES GET,PREFIX=WTG macro. This macro call also establishes the address of the recovery routine, IFG0RR0A (see Diagram 6.3). For details concerning the use of the WTG table, see "WTG Table." [OIN00000]

Virtual storage for a O/C/EOV main work area is obtained for each valid entry in the parameter list. The parameter list entries can represent (1) DCBs specifying an access method or EXCP request or (2) ACBs for a VSAM data set, a VSAM or system catalog, a spooled (SYSIN or SYSOUT) data set, or a VTAM data set. The way each type of DCB and ACB is identified is described in "DCBs and ACBs Processed." [OIN10225]

2 This routine is a second entry point to the Open function, entered from Close when data sets with unlike characteristics (also called *unlike attributes*) are being concatenated. To concatenate data sets with unlike characteristics, it is necessary to modify the characteristics contained in the DCB. These DCB fields are modified by closing the DCB and reopening it. Each work area, except those gotten for VSAM catalogs, will be initialized with the control blocks needed to read data set labels, etc., and indicators will be set in the ACB or DCB to show that an open for the data set is in process. This routine also issues the IECRES INIT macro to copy the DCB/ACB from the user's storage to protected storage. [IGC0101I]

3 The JFCB for the data set being opened will be copied from the scheduler work area (SWA) to the O/C/EOV main work area. [OIN11670]

If none of the data sets represented by the entries in the DCB/ACB parameter list can be opened, the WTG table and associated work areas are freed and control is returned to the caller. [OIN1300]

IGC0002B—OPEN TYPE=J (SVC 22)

1 This function is entered from the SVC second-level interrupt handler (SLIH) via SVC 22 (OPEN macro, TYPE=J). See *OS/VS2 System Programming Library: Data Management* for a description of the uses of the RDJFCB and OPEN TYPE=J macros. A WTG table is gotten and initialized for this request. [IGC0002B]

2 Virtual storage is obtained for an O/C/EOV main work area for each DCB and valid ACB in the parameter list (VSAM and VTAM ACBs are not valid because OPEN TYPE=J is not supported). A DEB/DCB, IOB, and ECB are initialized in each work area for reading data set labels, etc. This routine also issues the IECRES INIT macro to copy the DCB/ACB from the user's storage to protected storage. [OIN21300]

3 The user's JFCB is copied from the area pointed to by the X'07' entry in the DCB exit list to the O/C/EOV main work area. Direct access data set names are enqueued upon and, to maintain data set security, the system's JFCB for the data set is read in and compared with the user-supplied JFCB to verify that no fields have been modified by the user that would violate the integrity of the system or the security of the data set. [OIN21450]

If none of the data sets represented by the entries in the parameter list can be opened, the WTG table is freed and control is returned to the caller.

IGC00011

4 Checks the ACB to determine whether a VSAM or master catalog is being opened and, if so, transfers control to the VSAM catalog routine, IFG0191X (see *OS/VS2 Catalog Management Logic*). [OIN13200]

IFG0193A

Obtains virtual storage for a DEB extension for all ACBs and DCBs being processed, except for SYSIN/SYSOUT DCBs (the DEB extension is obtained for the associated ACB). Initializes the DEB extension. [OIN32800]

If a VTAM ACB is being opened, this routine transfers control to the VTAM open string, the first load of which is IFG0192F. [OIN32820]

Checks the JFCB to determine whether a VSAM data set is being opened and, if so, transfers control to VSAM open module IFG0192A (see *OS/VS2 Virtual Storage Access Method (VSAM) Logic*). [OIN32930]

If the open option is OUTINX or EXTEND, IFG0193A saves the JFCDISP flags, sets the JFCDISP to MOD, and changes the open option to OUTIN (when OUTINX is specified) or to OUTPUT (when EXTEND is specified).

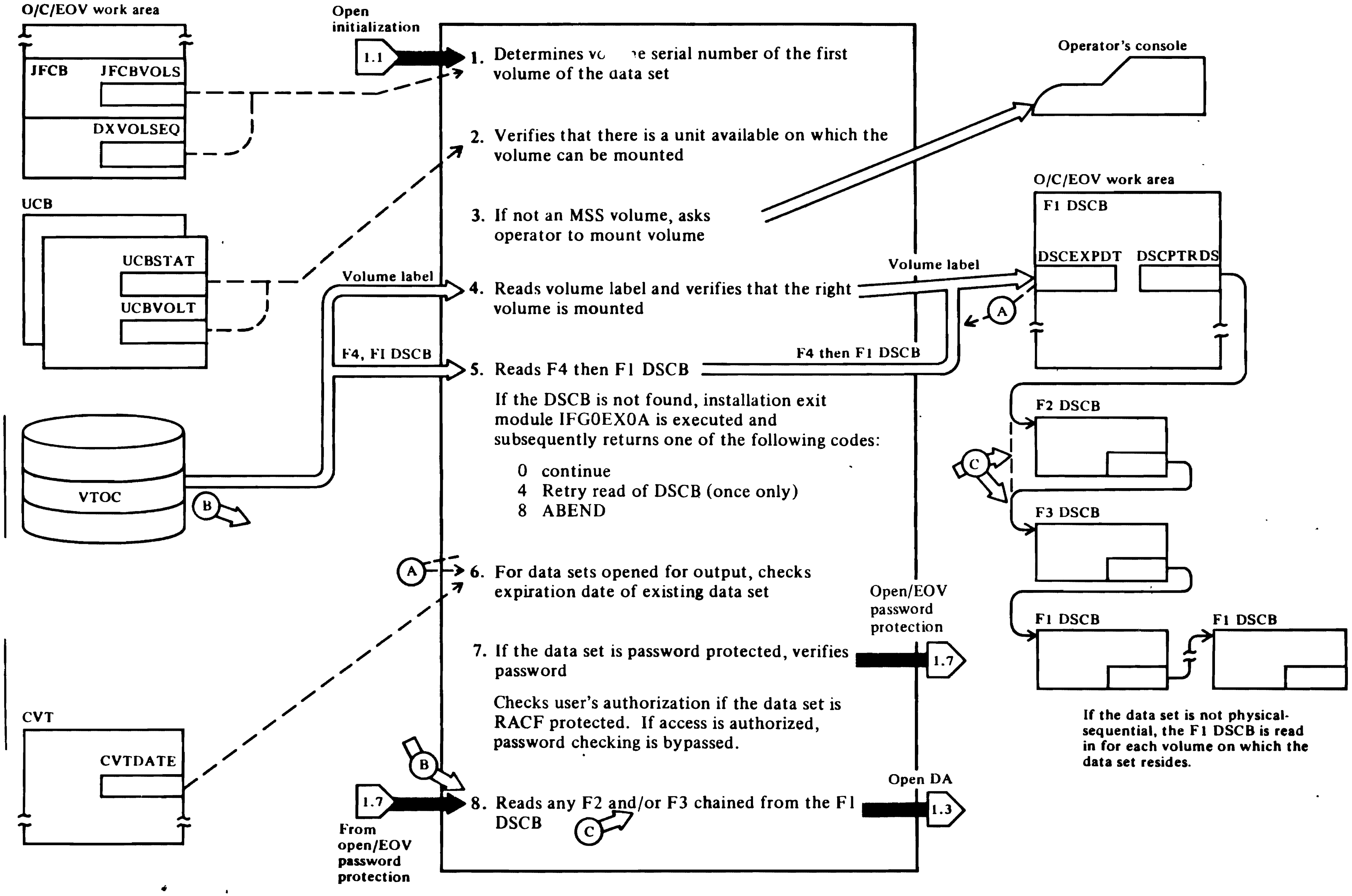
Checks the DCB and JFCB to determine whether an ISAM data set is being opened to be processed using the VSAM-ISAM compatibility interface, and, if so, transfers control to the VSAM open module IFG0192I (see *OS/VS2 Virtual Storage Access Method (VSAM) Logic*). [OIN33200]

5 Checks for spooled ACB or DCB (a SYSIN or SYSOUT data set) and, if present, transfers control to the Open merge function. [OIN32937]

Checks the TIOT DD entry for the data set to determine whether it is a dummy data set or a data set on a unit record device and, if so, transfers control to the Open merge function. [OIN33800]

6 The remaining device types and data set organizations (excluding those in steps 4 and 5) require volume verification and perhaps volume mounting. Before control is transferred to the volume mount and verification function, this routine determines the volume sequence number of the first volume of the data set to be mounted, using the volume sequence number in the JFCBVLSQ field and the disposition specified in the DISP operand of the DD statement. The sequence number of the first volume to be mounted is saved in the DXVOLSEQ field of the O/C/EOV main work area. For physical sequential data sets, this number must be less or equal to the number of volume serial numbers specified in the DD statement; this prevents attempts to read JFCB extensions that don't exist, and also prevents processing nonspecific (scratch) requests for any but the first volume of the data set. [OIN34660]

Diagram 1.2. Open Direct-Access, Volume Mount and Verify



1. Determines volume serial number of the first volume of the data set
2. Verifies that there is a unit available on which the volume can be mounted
3. If not an MSS volume, asks operator to mount volume
4. Reads volume label and verifies that the right volume is mounted
5. Reads F4 then F1 DSCB

   If the DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

   0 continue
   4 Retry read of DSCB (once only)
   8 ABEND
6. For data sets opened for output, checks expiration date of existing data set
7. If the data set is password protected, verifies password

   Checks user's authorization if the data set is RACF protected. If access is authorized, password checking is bypassed.
8. Reads any F2 and/or F3 chained from the F1 DSCB

If the data set is not physical-sequential, the F1 DSCB is read in for each volume on which the data set resides.

Diagram 1.2. Open Direct Access, Volume Mount and Verify

IFG0194A

1 This routine uses the volume sequence in the DXVOLSEQ field of the O/C/EOV work area in conjunction with the volume serial numbers in the JFCB and JFCB extension to determine the volume serial number of the next volume to be processed (the *new* volume), referred to in the source code as volume *n*. If another volume is to be processed after the *new* volume, this routine determines its serial number also so that a look-ahead mount message can be issued to the operator to mount this volume. The volume serial number of volume *n–1* is also saved if that volume is to be demounted. [VUS03300] However, the look-ahead mount is bypassed for all MSS volumes. [VUS06500]

For nonspecific requests (requests for scratch volumes), the characters SCRTCH are passed to the message issuing and volume verification routines in the DXVOLSR2 field. [VUS04000]

Physical sequential data sets:

Physical sequential data set organizations include BSAM, QSAM, and BDAM create. Data sets being accessed using the EXCP access technique are also considered to have physical sequential organization. This routine finds the volume serial number of the first and second volumes to be mounted. These volumes are referred to as volume *n* and volume *n+1* in the source code. [VUS03300]

IFG0194J

When a look-ahead mount message is required to tell the operator to mount the next volume (volume *n+1*) of a physical sequential data set, this routine issues message IEC501E. No verification is made that volume *n+1* has been mounted until end-of-volume is reached on volume *n*. The look-ahead mount is bypassed for all MSS volumes.

IFG0194A

ISAM, BDAM, and concatenated BPAM data sets:

This routine finds the volume serial number of the first volume to be mounted. For multivolume and concatenated data sets, this routine is reentered after each volume is mounted and verified to determine the next volume serial number so that all volumes can be mounted in parallel. [VUS03220] However, the look-ahead mount is bypassed for all MSS volumes. [VUS06500]

2 Checks the UCB entries in the TIOT to determine whether a unit allocated to this task can be used to mount the volume. [VUS05000]

Checks the available units for one that is compatible with the volume to be processed. [VUS08000]

Increments the data management count in the UCBDMCT of the unit to be used to indicate that the unit is in use. [VUS05400]

IFG0194J

3 When a volume must be demounted in order to mount the required volume, this routine issues the IEC502E message to the operator unless it is a MSS volume. If a MSS volume is being demounted, the ICBMNTDE TYPE=DMNT macro is issued to demount the volume without operator intervention. [WTO01900]

Issues message IEC501A to the operator to tell him that a specific volume is to be mounted on a specific device. If the request is to mount a MSS volume, the WTO message is bypassed. The ICBMNTDE TYPE=MNT macro is issued to mount the volume without operator intervention. [WTO02000]

If the volume to be mounted is being opened for a TSO task, a message is sent to the TSO terminal user to indicate that operator action is required regarding his data set. [WTO08800]

If the volume to be mounted or demounted is a MSS volume, no WTO message is issued. A SVC 126 is issued which processes the mount or demount request without operator intervention. The SVC is called by issuing the ICBMNTDE macro. Any error detected results in a 413 ABEND. [WTO08900, WTO08950]

IFG0194C

4 If the volume was not already mounted, this routine reads the volume label (cylinder 0, track 0, record 3) into the O/C/EOV main work area. [DAMV0100]

If the wrong volume is detected, the volume is rejected and this routine retries the read when another volume is ready.

If the volume mount/demount request is for a MSS volume and not a SCRATCH request, it is considered a hardware error and results in a 413 ABEND for OPEN, or a 737 ABEND for EOV. [DAMV0112]

Moves the address of the VTOC from the volume label to the UCB. [DAMV0120]

Verifies that the correct volume is mounted. [DAMV0200]

5 Reads the format-4 DSCB for the volume, then the format-1 DSCB for the data set into the O/C/EOV main work area. [DAMV1010]

For VIO data sets, this routine moves the format-1 DSCB maintained in the scheduler work area (SWA) to the O/C/EOV main work area. [DAMV0290]

For nonphysical-sequential data sets, checks for an ascending volume sequence number in the format-1 DSCB. [DAMV2300]

If the DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

0 continue
4 Retry read of DSCB (once only)
8 ABEND

IFG0194A

For data sets being opened for output or for modification (DISP=MOD), the last volume of the data set is mounted first, unless a sequence number was provided in the DD statement (the JFCBVLSQ field will contain this sequence number). If the last track of the data set is not on the last volume of the data set, a search of the other volumes, beginning with the first volume of the data set, is made to find the end of the data set. [VUS02500]

IFG0194C

6 When an existing data set is being opened for output, this routine checks to verify that the expiration date in the format-1 DSCB has passed. [DAMV2410]

IFG0194J

When the expiration date has not passed, this routine writes message IEC507D to see if the operator wants to override the expiration date for the data set. [WTO03000]

IFG0194C

7 This routine checks for data sets that are password protected, and when such a data set is detected, it passes control to the Open/EOV password function (Diagram 1.7) to verify the password. [DAMV2500]

If security bypass is not authorized, this routine checks the caller's authorization to access a RACF-protected data set. If the format-1 DSCB indicates that the data set is RACF-protected, the RACHECK macro is issued to determine the caller's authorization. If the caller is authorized to access the data set, password checking is bypassed. If the caller is not authorized to access the RACF-protected data set, the job terminates with an ABEND. If the data set was not RACF-protected, password checking occurs (unless security bypass is authorized). [DAMV2500, DAMV4500]

8 This routine gets virtual storage, chains it from the O/C/EOV work area, and reads in format-2 and format-3 DSCBs. Format-2 DSCBs exist for ISAM data sets, and format-3 DSCBs are used for every data set that has more than three extents on a volume. [DAMV3000]

IFG0194C

8 This routine checks for data sets that are password protected, and when such a data set is detected, it passes control to the Open/EOV password function (Diagram 1.7) to verify the password. [DAMV2500]

9 This routine gets virtual storage, chains it from the O/C/EOV work area, and reads in format-2 and format-3 DSCBs. Format-2 DSCBs exist for ISAM data sets, and format-3 DSCBs are used for every data set that has more than three extents on a volume [DAMV3000]

Diagram 1.3. Open Direct-Access

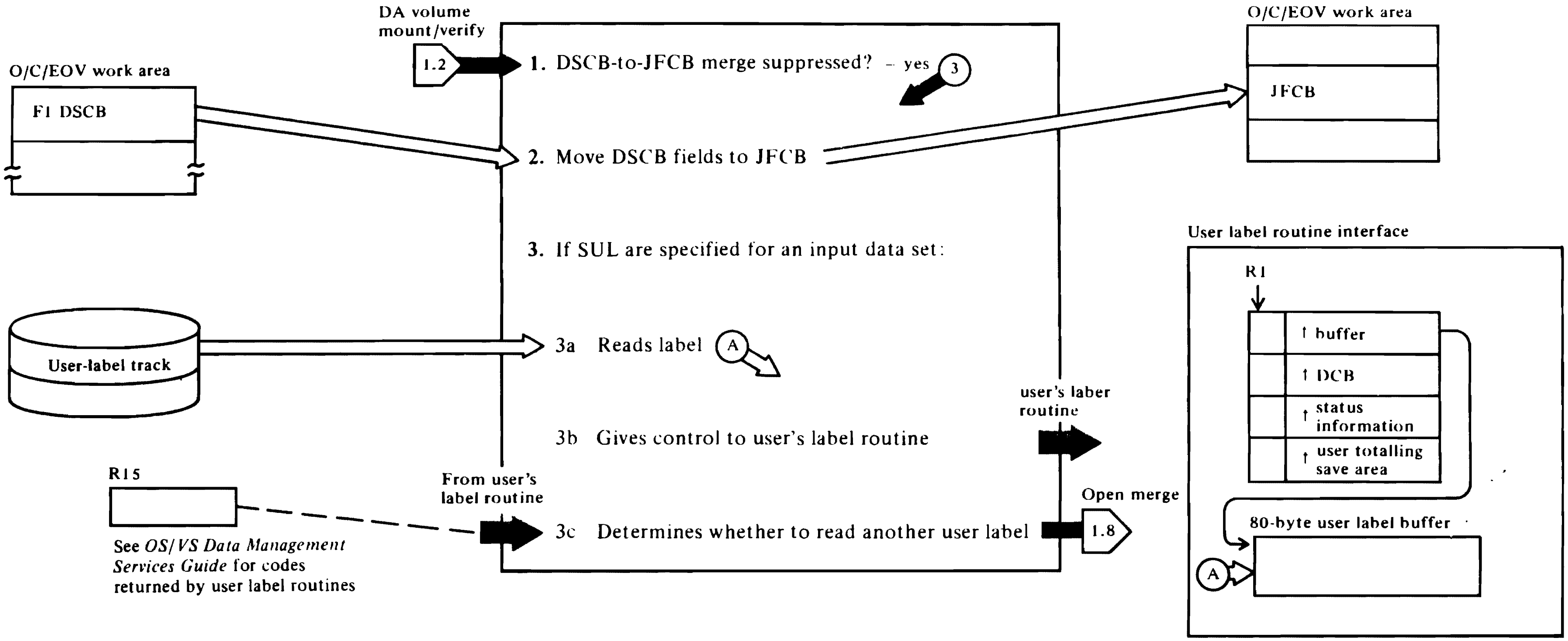

DA volume mount/verify
1.2
1. DSCB-to-JFCB merge suppressed? – yes
3
O/C/EOV work area
F1 DSCB
2. Move DSCB fields to JFCB
O/C/EOV work area
JFCB
3. If SUL are specified for an input data set:
User-label track
3a Reads label
A
3b Gives control to user's label routine
user's laber routine
From user's label routine
3c Determines whether to read another user label
Open merge
1.8
R15
See *OS/VS Data Management Services Guide* for codes returned by user label routines
User label routine interface
R1
↑ buffer
↑ DCB
↑ status information
↑ user totalling save area
80-byte user label buffer
A

Diagram 1.3. Open Direct Access

IFG0195A

This function is entered from the volume mount and verification function after the first volume of the data set or a specified volume containing the data set has been mounted, its label verified, and the format-1 DSCB for the data set has been read into the O/C/EOV main work area. (Format-2 and format-3 DSCBs will also have been read in if they exist for the data set on this volume.)

1 At entry this routine determines whether a VTOC is being opened (indicated by the presence of a format-4 DSCB in the work area, rather than a format-1 DSCB). If a VTOC is being opened, the merging of the DSCB into the JFCB is skipped, and control is immediately transferred to the JFCB-to-DCB merge function (see Diagram 1.8). [IFG0195A]

This routine determines whether the caller has requested that the format-1 DSCB fields not be moved (merged) into the JFCB. The caller of OPENJ can suppress merging by setting the JFCBTSDM field to X'04'. [ODA11800]

2 If the DSCB-to-JFCB merge has not been suppressed, the following fields will be merged if they were not supplied in the JFCB: RECFM, KEYLEN, DSORG, LRECL, RKP, BFALN, BLKSIZE (unless ISAM load mode is specified in the DCB), and OPTCD. The JFCBMASK field is modified to indicate which fields in the JFCB have been modified with information from the format-1 DSCB. [ODA14600]

IFG0195J (alias for IFG0554L)

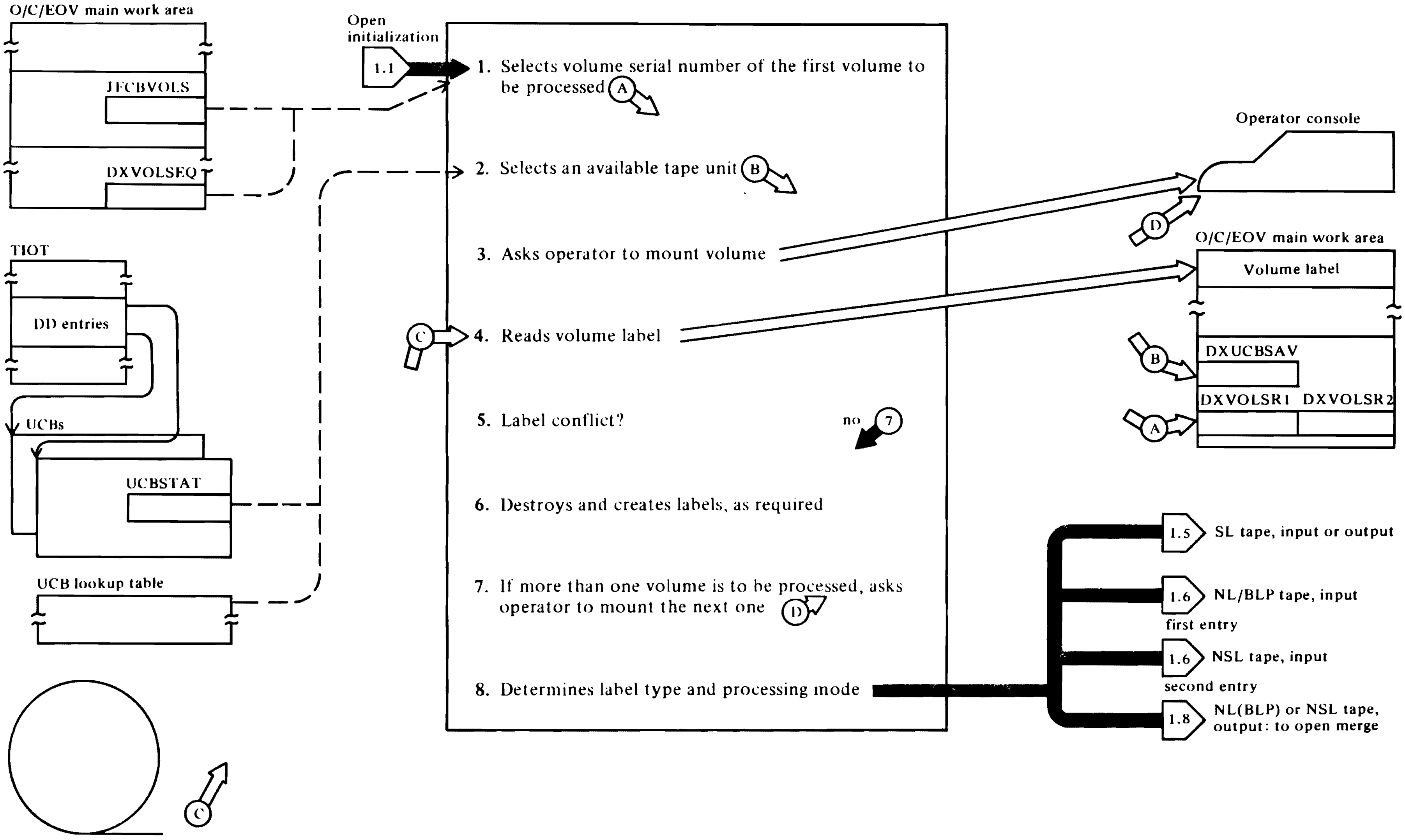
3 This routine examines each DCB in the parameter list and the JFCBLTYP field of each associated JFCB to determine whether user labels are to be processed for input data sets. (Output user labels will be written after the JFCB-to-DCB merge and after the user's DCB exit has been taken). [IFG0554L]

3a This routine is entered when user labels are to be read and verified by the user for a direct access data set being opened for input. This routine gets virtual storage to read the labels into, reads in the first label, and transfers control to the user's label routine identified in the DCB exit list, if there is an active entry. When he has verified the label, the user returns a code in register 15 to indicate whether another label should be read or whether label processing is complete. [ODA20200]

Reads label, moves it into the user's label buffer, copies the DCB being used by Open (the DCB copy) into the user's storage, transfers control to the user's label routine, and verifies the DEB when the user returns control. [ULSYNCH]

3c Examines contents of register 15 to determine whether to read another label, write an updated label, or terminate label processing. [ULT130]

Diagram 1.4. Open Tape, Volume Mount and Verify



1. Selects volume serial number of the first volume to be processed
2. Selects an available tape unit
3. Asks operator to mount volume
4. Reads volume label
5. Label conflict?
6. Destroys and creates labels, as required
7. If more than one volume is to be processed, asks operator to mount the next one
8. Determines label type and processing mode

Diagram 1.4. Open Tape, Volume Mount and Verify

IFG0194A

1 This function is entered from the Open initialization function when it has been determined that a tape data set is to be opened. Using the volume sequence number (DXVOLSEQ field) to determine how many volumes the data set to be processed resides on, this routine searches the JFCB and its extension to find the first volume (referred to as volume *n* in the source code). If multiple volumes are being processed, the volume serial number of the next volume (referred to as volume *n+1* in the source code) is also determined so that a look-ahead mount message can be issued to the operator. If VOL=REF was specified in the DD statement, this routine reads in the referenced list of volume serial numbers and selects the first volume to be processed, which is the volume represented by the last volume serial number in the list. [VUS03300]

2 Finds a tape unit allocated to this task that is available for use with the selected volume (note that the volume may already be mounted). [VUS05120]

For nonspecific (scratch) requests, this routine puts the volume serial number of the volume to be used in the JFCB and calls the QMNGRIO routine to copy the JFCB to the SWA. [VUS06000]

IFG0194J

3 If the volume is not mounted, this routine issues message IEC501A to the operator requesting that he mount the volume. [WTO02000]

IFG0194F

The validity of a specification of PROTECT=YES is tested.

This routine attempts to read the first record on the tape and determine which of the following label types is being processed: standard label (SL), ASCII standard label (AL), or unlabeled (NL). The label type specified in the JFCBLTYP field is checked to ensure that the correct label type has been read. An NL tape is assumed to be a nonstandard labeled (NSL) tape if JFCBLTYP specifies NSL, in which case, no volume verification is attempted. If BLP is specified in the JFCBLTYP field, volume verification is bypassed. Each installation must ensure that this option is not misused. This can be done by authorizing a reader procedure that will accept the LABEL=BLP operand of the DD statement. [TMV02300]

RACF authorization to the verified AL or SL tape is checked. If defined, and authorized, sets DEBXDSSI indicating RACF defined.

5 Several types of label, density, and mode conflicts can occur during volume verification. *OS/VS Tape Labels* describes these conflicts and their resultant actions. This routine checks for these conflicts. [TMV02300]

IFG0193C

6 This object module, IFG0193C, performs the same function as IFG0553C. If this routine has not been replaced by an installation-supplied label editor routine (which must be named OMODVOL1), it performs the label editing function. It is entered when label and density conflicts are encountered during volume verification. The type of label or density conflict is indicated in the JFCBMASK+5 field. An installation-supplied OMODVOL1 routine can resolve the conflict and reset the appropriate bit, or can reject the volume by setting the high-order bit in the UCBDMCT field of the UCB. OMODVOL1 is entered in key 0 and is passed the address of the copy of the DCB in protected storage. The address of the DCB is passed in register 2. OMODVOL1 must return to IGG0190A, which is an alias for IFG0193D. Reference information concerning the label editor routines is provided in *OS/VS Tape Labels*. [OMO01000]

IFG0193D

This routine is called by the user's OMODVOL1 routine or by IFG0193C, the standard label editing routine. [DSL00000]

When a standard label is to be destroyed, this routine reads labels until a HDR1 label has been read in. [DSL01100]

Checks RACF ALTER authorization to the volume. If authorized, bypasses password check. If not authorized, requests that another volume be mounted. If not defined, performs password check.

Checks the HDR1 label for a password-protected data set, and if such a data set is encountered, issues an error message to the operator, then requests that another volume be mounted. [DSL08200]

Checks expiration date, and if the expiration date has not passed, issues a message to the operator to get permission to ignore the expiration date. [DSL03000]

Issues a message to the operator requesting permission to destroy the label. [DSL05000]

Checks the HDR2 label to ensure that the existing data set is not an existing checkpoint data set. If it is, issues a message to the operator. [DSL06500]

Writes a tape mark over the standard volume label to destroy the label, then determines from the JFCBLTYP field which type of label is requested. Sets UCBTFL1 accordingly. [DSL07000]

If tape is RACF defined and the user is authorized, deletes RACF definition of the volume.

IFG0193E

Entered when a standard label is to be created, this routine issues a message to the operator to ask him to specify the volume serial number and owner of the data set. [CSL01000]

Check RACF ALTER authorization to volume. Abnormally terminate if not authorized to specific request. Reissue operator message if not authorized to nonspecific request. Continue if volume not RACF defined. SET DEBXDSSI bit if authorized and continue.

Verifies operator's specification and completes the label. Translates EBCDIC to ASCII for AL tapes. [CSL03500]

Rewinds the tape to load point and writes the label. Writes a tape mark after HDR1 label. [CSL04000]

IFG0194J

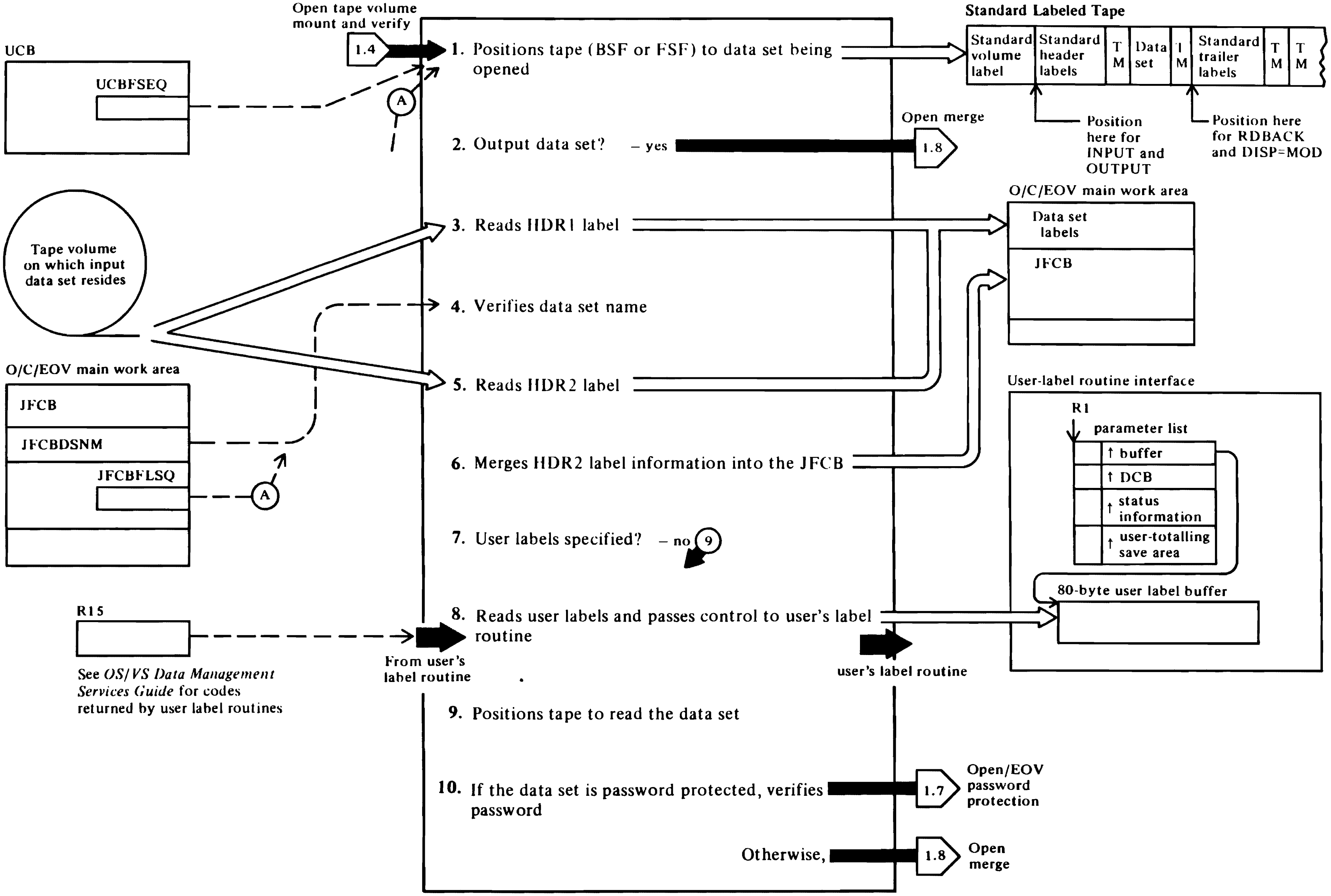
7 If more than one volume is being processed, and the second volume is not mounted, this routine issues the look-ahead mount message (IEC501E) for the next volume to be processed. No verification is made that volume *n+1* has been mounted until EOV is readied on volume *n+1*. [WTO02500]

IFG0194A

8 When the volume has been mounted and verified, labels accepted, and a look-ahead mount message issued, this routine transfers control to the routine, determined during Open initialization, that will handle the specified label type and processing mode (input or

output): SL tape, input or output, Diagram 1.5; NL (BLP) tape input, Diagram 1.6, first entry; NSL tape, input, Diagram 1.6, second entry; NL (BLP) or NSL tape, output, Diagram 1.8.

Diagram 1.5. Open SL Tape



1. Positions tape (BSF or FSF) to data set being opened

2. Output data set? – yes

3. Reads HDR1 label

4. Verifies data set name

5. Reads HDR2 label

6. Merges HDR2 label information into the JFCB

7. User labels specified? – no 9

8. Reads user labels and passes control to user's label routine

9. Positions tape to read the data set

10. If the data set is password protected, verifies password

Otherwise,

See *OS/VS Data Management Services Guide* for codes returned by user label routines

Diagram 1.5. Open SL Tape

IFG0195B

1 This function is entered from the tape volume mount and verification function when it has been determined that the tape volume has standard labels (AL, AUL, SL, or SUL). The volume serial number will have been verified. The tape may be being opened for input or output.

The block count in the DCB (DCBBLKCT) is zeroed in all DCBs except EXCP DCBs, which don't have the device-dependent portion of the DCB. [OTA25200]

This routine ensures (1) that checkpoint data sets reside only on password-protected volumes and (2) that non-checkpoint data sets are not written on volumes that contain checkpoint data sets. [OTA25400]

If the file sequence number in the UCB (UCBDMCT) is zero, the file sequence must be obtained from the first header label. The file sequence number and count are set to one in case there is a tape mark instead of an HDR1 label. [OTA25480]

The tape is positioned to the start of the file for input and output. The tape is positioned to the end of the file for RDBACK and output DISP=MOD. [OTA26100]

When positioning to the start of the file, the tape is rewound if the tape starts at the desired file sequence. [OTA26850]

When opening for output to file 1, UCBFSEQ is set to 1. [OTA26700]

The tape is backspaced or forward spaced files according to the relationship of UCBFSEQ and JFCBFSEQ.

If the HDR1 label of the desired file is passed while spacing to the end of the data set, the DSNAME is checked so that any ABEND or DISP=MOD previous volume recovery can save an average of 8 minutes it might take to space to the trailer labels. [OTA26910]

Trailer labels are checked for EOV labels while forward spacing files or if no spacing is done. Encountering an EOV label causes the next volume to be used instead of the current one. [OTA26920]

2 Tests the option byte in the open parameter list to determine whether the data set is being opened for output or OUTIN. If either of these options is specified, standard label processing is performed after the Open merge function. See the Open merge and the Tape, SL output label functions, Diagrams 1.8 and 1.9, respectively. [OTA27000]

3 Reads the label at which the volume is positioned. For input data sets this will be the HDR1 label. For data sets being opened for DISP=MOD or to be read backward, this will be the EOV1 or the EOF1 label. [OTA27100]

This routine ensures (1) that checkpoint data sets reside only on password-protected volumes and (2) that non-checkpoint data sets are not written on volumes that contain checkpoint data sets. [OTA25400]

IFG0195H

4 Compares the last 17 characters of the data set name field of the JFCB (JFCBDSNM) with the data set identifier field in the HDR1 label. [OTA28500]

If they do not match and the data set is a generation data set, the tape may have been created under DOS. [OTA28800]

After the DSNAME has been verified, the block count is moved from the label to the DCB (DCBBLKCT). (This is only done if the label read is a trailer label and, in the case of an EXCP DCB, the device-dependent section of the DCB is present.) [OTA29300]

For data sets being opened for input then output (INOUT), this routine checks the expiration date, and if it is not passed, asks the operator to specify whether or not the volume can be used. [OTA29500]

IFG0195K

5 Reads HDR2 label and verifies that it is the expected label type. [OTA31400]

If the HDR2 label indicates that a checkpoint data set is being opened, this routine turns on the indicator in the DSAB that identifies it as a checkpoint data set. [OTA32100]

6 If the JFCBTSDM switch, which indicates whether the label is to be merged into the JFCB, is not set, the RECFM, blocking attribute, CCTYPE, BLKSIZE, LRECL, tape recording technique, and BUFOF are merged into their counterpart fields in the JFCB that contain zeros. [OTA32300]

7 Checks to see if user labels are specified. [OTA33900]

IFG0195N

8 Checks to see if the user's DCB points to an active user label exit routine. For DD statements specifying DISP=MOD, the exit must be an active user trailer (UTL) exit; for data sets be opened for input, the user header label (UHL) exit must be active. [OTA35000]

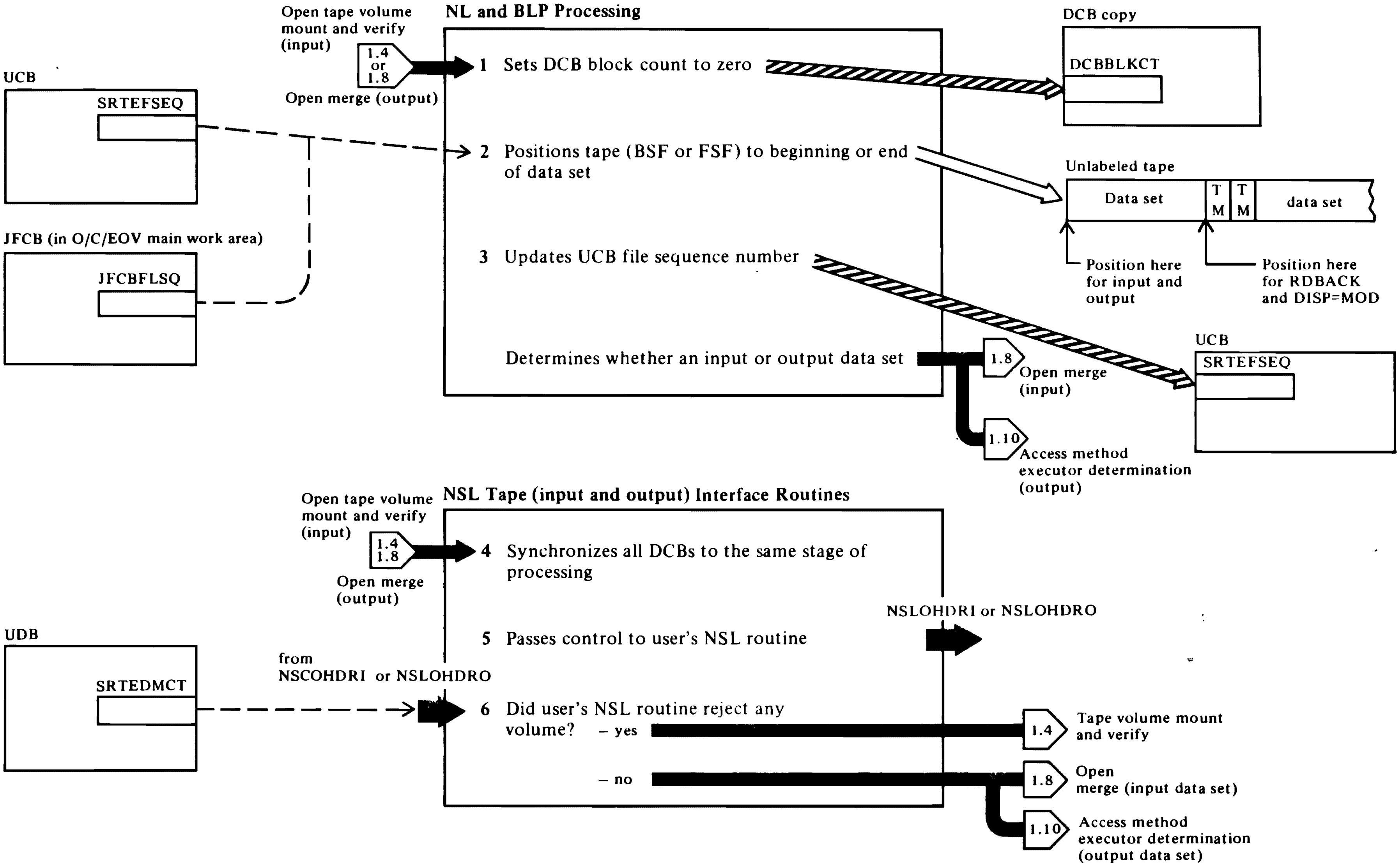
Gets storage and reads user labels into it: translates ASCII labels to EBCDIC; verifies that they are user labels; moves labels into the user's buffer and transfers control to the user to verify the labels. When control is returned, this routine continues to read labels, one at a time until eight labels have been read or until the user indicates, using a return code in register 15, that no more need be read. [OTA35700]

Copies the DCB in protected storage to user's storage before passing control to the user's label routine. [OTA37400]

9 When all labels have been read, this routine repositions the tape to the beginning of the data (to the end of the data for DISP=MOD). [OTA36700]

10 If the data set is password-protected, this routine transfers control to the Open password-protection function (Diagram 1.7), which, when the password has been successfully supplied, passes control to the Open merge function. If the data set is not password-protected, control is passed immediately to the Open merge function. [OTA37000]

Diagram 1.6. Open Tape, NSL and NL(BLP)



NL and BLP Processing

1 Sets DCB block count to zero

2 Positions tape (BSF or FSF) to beginning or end of data set

3 Updates UCB file sequence number

Determines whether an input or output data set

NSL Tape (input and output) Interface Routines

4 Synchronizes all DCBs to the same stage of processing

5 Passes control to user's NSL routine

6 Did user's NSL routine reject any volume? – yes

– no

Open tape volume mount and verify (input)
1.4 or 1.8
Open merge (output)
UCB
SRTEFSEQ
JFCB (in O/C/EOV main work area)
JFCBFLSQ
DCB copy
DCBBLKCT
Unlabeled tape
Data set
T M
T M
data set
Position here for input and output
Position here for RDBACK and DISP=MOD
UCB
SRTEFSEQ
1.8
Open merge (input)
1.10
Access method executor determination (output)
Open tape volume mount and verify (input)
1.4
1.8
Open merge (output)
UDB
SRTEDMCT
from NSCOHDRI or NSLOHDRO
NSLOHDRI or NSLOHDRO
1.4
Tape volume mount and verify
1.8
Open merge (input data set)
1.10
Access method executor determination (output data set)

Diagram 1.6. Open Tape, NSL and NL (BLP)

IFG0195C

1 When the label processing requested is no label (NL) or bypass label processing (BLP), this routine is entered after the correct tape volume has been mounted and/or verified. Sets the block count to zero in the copy of the user's DCB. [OTA50100]

Sets the file sequence number and count to one in the UCB if they are currently set to zero. [OTA50200]

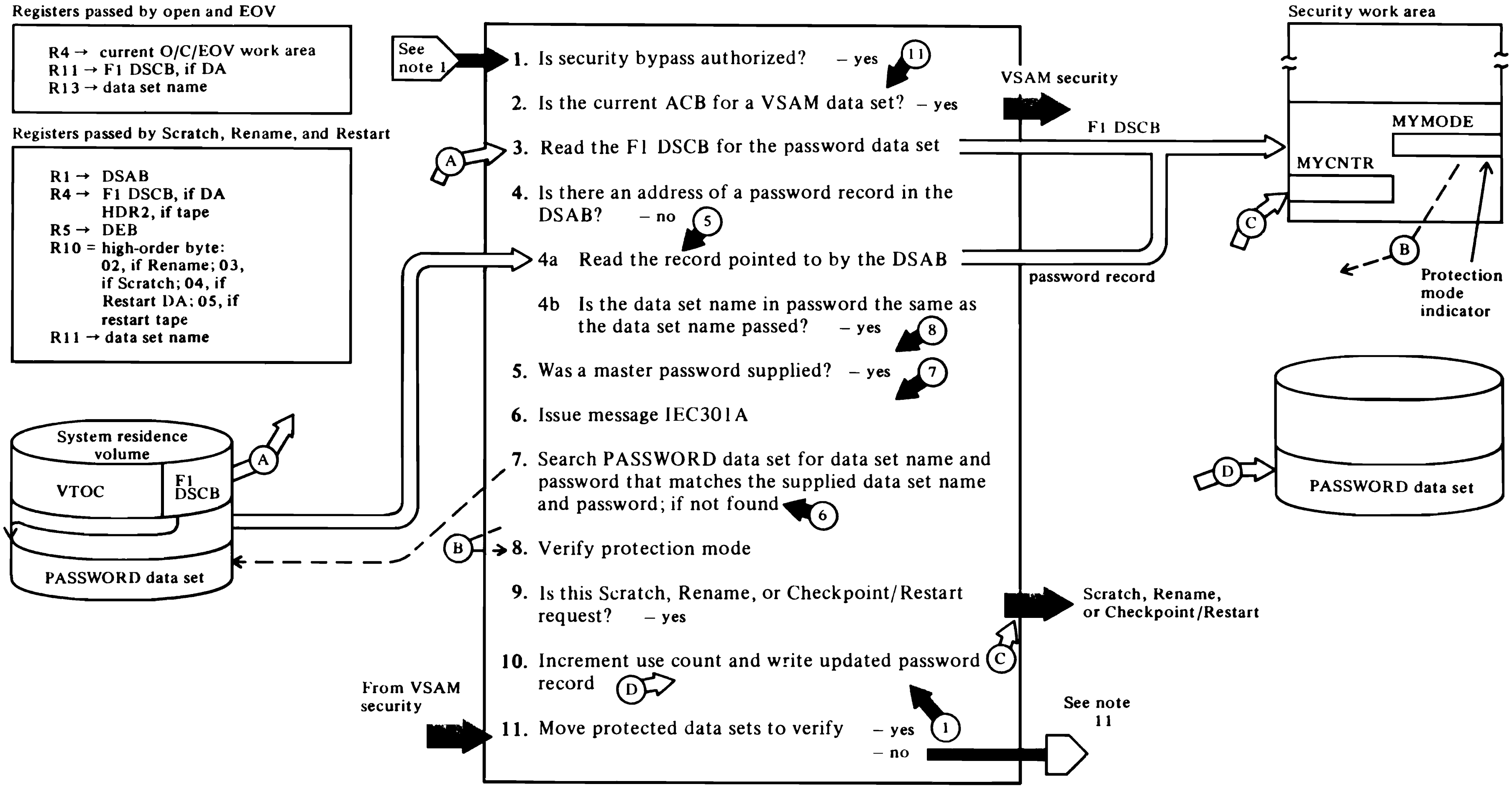
2 The tape positioning done for NL and BLP data sets depends on (1) the setting of the JFCB and UCB file sequence numbers (the JFCBFLSQ field indicates the sequence number of the data set to be processed on the tape and the SRTEFSEQ field of the UCB contains the sequence number of the data set to which the device is currently positioned) and (2) whether the data set is being opened (a) for input or output or (b) to be read backward or updated (DISP=MOD). [OTA50600]

IFG0195D for Input Data Sets
IFG0196O for Output Data Sets

4 This routine is entered when a tape volume with a nonstandard label has been mounted. Parallel DCB processing is synchronized, register 5 is set to point to the DCB parameter list, and control is transferred to NSLOHDRI for input data sets or NSLOHDRO for output data sets. [OTA60100] [OTA70100]

6 When control is returned from NSLOHDRI or NSLOHDRO, this routine checks to see if any of the mounted volumes were rejected. If so, control is passed to the tape volume mount and verification routine to attempt to get the right volume mounted. NSLOHDRI and NSLOHDRO set the UCBDMCT field to X'80' to indicate rejection. [OTA60500] [OTA70800]

Diagram 1.7. Open/EOV Password Protection



Registers passed by open and EOV

R4 → current O/C/EOV work area
R11 → F1 DSCB, if DA
R13 → data set name

Registers passed by Scratch, Rename, and Restart

R1 → DSAB
R4 → F1 DSCB, if DA
HDR2, if tape
R5 → DEB
R10 = high-order byte:
02, if Rename; 03,
if Scratch; 04, if
Restart DA; 05, if
restart tape
R11 → data set name

1. Is security bypass authorized? – yes 11
2. Is the current ACB for a VSAM data set? – yes
3. Read the F1 DSCB for the password data set
4. Is there an address of a password record in the DSAB? – no 5

4a Read the record pointed to by the DSAB

4b Is the data set name in password the same as the data set name passed? – yes 8

5. Was a master password supplied? – yes 7
6. Issue message IEC301A
7. Search PASSWORD data set for data set name and password that matches the supplied data set name and password; if not found 6
8. Verify protection mode
9. Is this Scratch, Rename, or Checkpoint/Restart request? – yes
10. Increment use count and write updated password record
11. Move protected data sets to verify – yes 1 – no

Diagram 1.7. Open/EOV Password Protection

This function is entered to verify that the problem program requesting access to a data set is authorized to process a password-protected data set in the mode that he is requesting (i.e., to read from or write to the data set). This function is entered from: Direct Access Volume Mount and Verification (IFG0194C); Standard Label Tape Positioning, Input, and Output (IFG0195N, IFG0195K, and IFG0196N, respectively); and EOV Tape Input, New Volume (IFG0553H). It is also entered from the Scratch and Rename components of DADSM (see *OS/VS2 DADSM Logic*), and from Checkpoint/Restart (see *OS/VS2 Checkpoint/Restart Logic*) to verify that the task being restarted still has access to the password-protected data sets. For tape data sets (only SL tapes can be password protected), this function is entered after the tape volume and label have been verified and other security checks (expiration date, file protection ring) have been passed. For SL output tapes this function is called after open merge processing. For SL input tapes, this function is called before open merge processing.

IFG0195T for Open and EOV requests
SECLOADA for Scratch, Rename, and Restart requests

1 These routines check the JSCBSWT1 field of the JSCB to determine whether password protection processing can be bypassed. [0EP01000 or SRP01000]

This routine checks the caller's authorization to access a data set that is RACF-protected if security bypass is not authorized. If the format-1 DSCB indicates that the data set is RACF-protected, the RACHECK macro is issued to determine the caller's authorization. If the caller is authorized to access the data set, password checking is bypassed. Otherwise, the job terminates with an ABEND. If the data set is not RACF-protected, password checking occurs (unless security bypass is authorized). If the tape volume has been shown to be RACF-protected, password checking is bypassed.

These routines next get a fetch-protected work area for used by the password-protection routines and initialize it.

IFG0195V

2 If the format-1 DSCB represents a VSAM data space, control is transferred to VSAM Security (IDA0192G) to ensure security protection for each data set in the data space. [TAS02000]

3 This routine reads the format-4 DSCB, determines the extents of the VTOC, then reads the format-1 DSCB of the PASSWORD data set into the security work area. [TAS03000]

4a This routine checks the DSAB to see if the address of a password record is in the DSABPTTR field; if so, this routine enqueues on the data set name of the PASSWORD data set and reads the record from the PASSWORD data set pointed to by the current DSAB in the DSABPTTR field into the security work area. [TAS04000]

4b If the data set name in the password record is the same as the name of the password-protected data set, this routine verifies that the protection mode indicator in the password record is compatible with the processing mode indicated in the option byte of the DCB parameter list for this data set. If they are compatible, this routine indicates successful password processing, increments the use count in the password record, and writes it back to the PASSWORD data set, then returns to the Open, EOV, DADSM, or Checkpoint/Restart function that called. [TAS04500]

5 If the password record pointed to by the DSABPTTR does not contain the same data set name as the password-protected data set, this routine checks for the presence of a master password in the TSBPSWD field of the TSB. The TSB exists only for TSO tasks. [TAS05000]

READPSWD

6 Issues a message to the operator or the TSO terminal to get the password. One retry is made to allow for operator error. If the correct password is not supplied, a 913 ABEND is issued using the DMABCOND interface to the problem determination routines. [OSQ41000]

IFG0195U

7 This routine searches the PASSWORD data set for the matching DSNAME concatenated with the password supplied by the operator or TSO terminal user. If the password record containing the matching DSNAME and password is found, the record is read into the security work area. [PCR01000]

8 If the search is successful, the protection-mode indicator in the password record (which can indicate a read-only data set or a data set that can be both read and written to) is compared with the use mode of the data set (from the option byte of the current DCB parameter list entry). If the protection-mode indicator is incompatible with the user's processing mode (for example, the user is opening the data set for output, but the protection-mode indicator allows reading only), a 913 ABEND is issued. [PCR01100]

SECLOADA

9 For Scratch or Rename requests, the security work area is freed and control is returned to Scratch or Rename. [SRP09000]

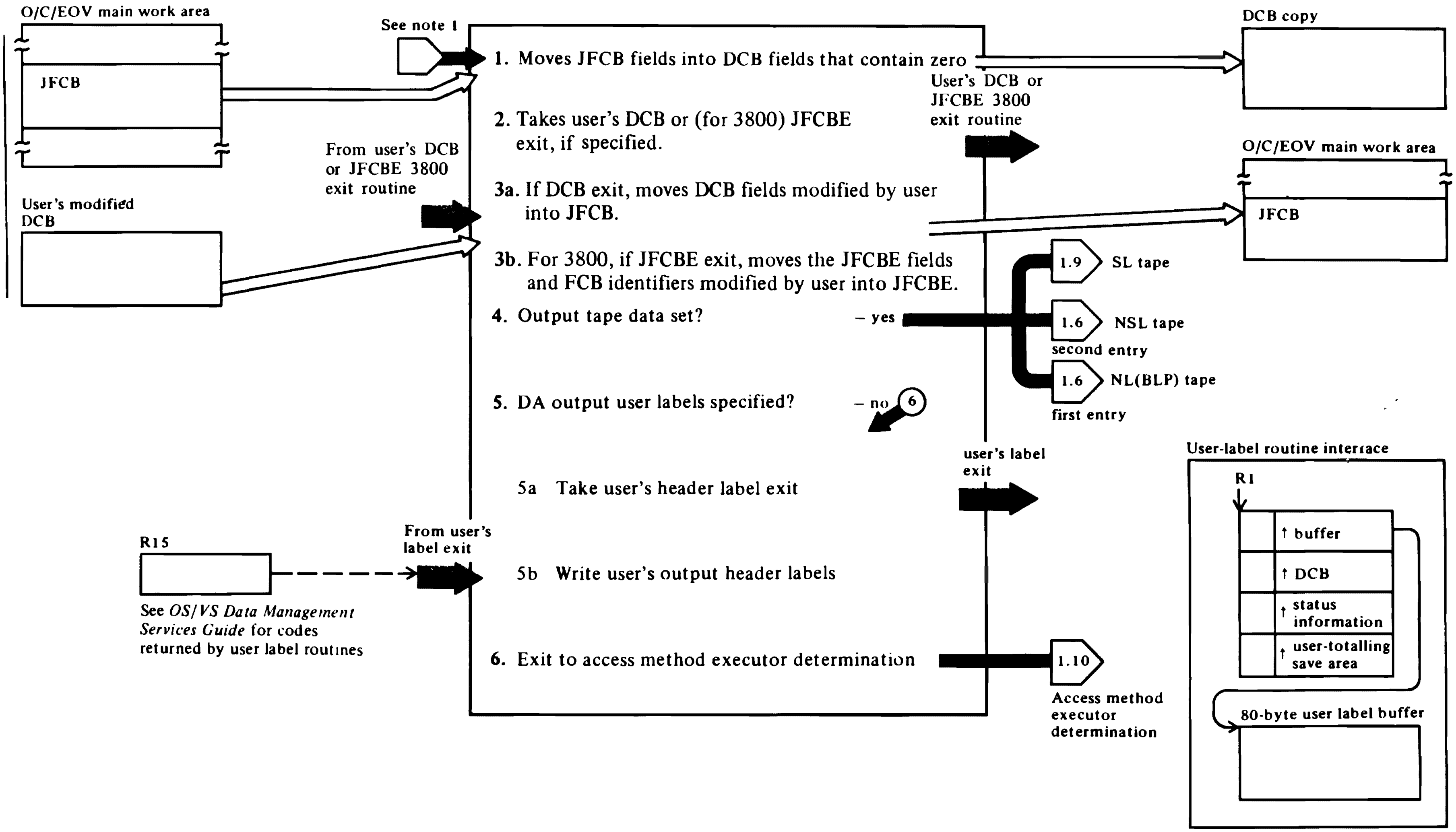
IFG0195U

10 If the processing mode (input or output) is compatible with the protection mode, (read-only, read-and-write), the use count in the password record is increased by one, the updated record is written to the PASSWORD data set and the PASSWORD data set is dequeued. The address of the password record is put in the DSABPTTR field of the DSAB. [PCR02000]

IFG0195T

11 When all of the data sets represented in the DCB/ACB parameter list have been verified, control is passed to the Open or EOV calling routine named in Step 1. [OEP09000]

Diagram 1.8. Open Merge



1. Moves JFCB fields into DCB fields that contain zero

2. Takes user's DCB or (for 3800) JFCBE exit, if specified.

3a. If DCB exit, moves DCB fields modified by user into JFCB.

3b. For 3800, if JFCBE exit, moves the JFCBE fields and FCB identifiers modified by user into JFCBE.

4. Output tape data set?

5. DA output user labels specified?

5a Take user's header label exit

5b Write user's output header labels

6. Exit to access method executor determination

O/C/EOV main work area
JFCB
User's modified DCB
See note 1
From user's DCB or JFCBE 3800 exit routine
R15
From user's label exit
See *OS/VS Data Management Services Guide* for codes returned by user label routines
DCB copy
User's DCB or JFCBE 3800 exit routine
O/C/EOV main work area
JFCB
– yes
1.9
SL tape
1.6
NSL tape
second entry
1.6
NL(BLP) tape
first entry
– no
6
user's label exit
1.10
Access method executor determination
User-label routine interface
R1
↑ buffer
↑ DCB
↑ status information
↑ user-totalling save area
80-byte user label buffer

Diagram 1.8. Open Merge

1 This function is entered by all of the callers of Open, except those opening VSAM or VTAM data sets. It is entered after Open initialization for SYSIN, SYSOUT, and dummy data sets (Diagram 1.1); it is entered following volume mounting and verification and SL input processing for data sets on tape and direct-access devices (Diagram 1.3); it is entered after label processing (Diagrams 1.5 and 1.6) for input data sets on tape; it is entered after volume mounting and verification for output data sets on tape (Diagram 1.4); and it is entered after password processing (Diagram 1.7) for password-protected data sets.

IFG0196J

Tests for graphics and teleprocessing DCBs. Different fields are merged for these DCBs. [OMG11600]

Determines, based on data set organization, which fields should be merged from the JFCB to the DCB. For all access methods, OPTCD, RECFM, BLKSIZE, DEN, BFTEK or BFALN, BUFL, and BUFNO are merged. For ISAM, BDAM, and SAM data sets, additional fields are merged. A record of the fields merged is kept in the JFCBMASK field. [OMG15000]

IFG0196K

For graphics and telecommunications DCBs, such as those representing the TCAM line group or the TCAM message queue, this routine moves the OPTCD, RECFM, DEN, BFTEK-BFALN, BUFL, and BUFNO in the JFCB into the fields of the DCB that contain zeros. [OMG25000]

IFG0196L

For a TSO task opening for input (INPUT, INOUT, RDBACK, or UPDAT), this routine sets up LRECL and BLKSIZE in the DCB, if not specified by the user. [IFG0196L]

2 If the user has specified that control be given to his DCB exit routine, the DCB in protected storage is copied into the user's storage and control is transferred to the user's DCB exit routine. This exit allows the problem program to modify the DCB before the reverse merge (DCB-to-JFCB) is made. When control is returned, the user's DCB is copied into protected storage so that his modifications will be reflected in the DCB used by Open. [OMG30600]

If the user has specified that control be given to his JFCBE exit routine, storage in the user's protection key is obtained, and the JFCBE is copied to it from the SWA, and control is passed to the user's JFCBE exit routine. If the user's JFCBE exit routine requests that the JFCBE be updated, the user's copy is copied to the SWA. The JFCBE is the JFCB extension for the 3800.

For VIO data sets, this routine turns off the write-verify bit in the DCBOPTCD field of the DCB. Write-verification is not necessary for VIO data sets. [OMG31300]

This routine checks for requests to open a VTOC for modification. For these requests, the caller's authorization to update VTOCs is verified via the TESTAUTH macro instruction. [OMG32200]

IFG0196M

3 All appropriate fields of the JFCB are merged unconditionally when a data set is being opened for OUTPUT or OUTIN. Only the fields that contain zeros are merged when the data set is being opened for INPUT, INOUT, or RDBACK. [OMG44800]

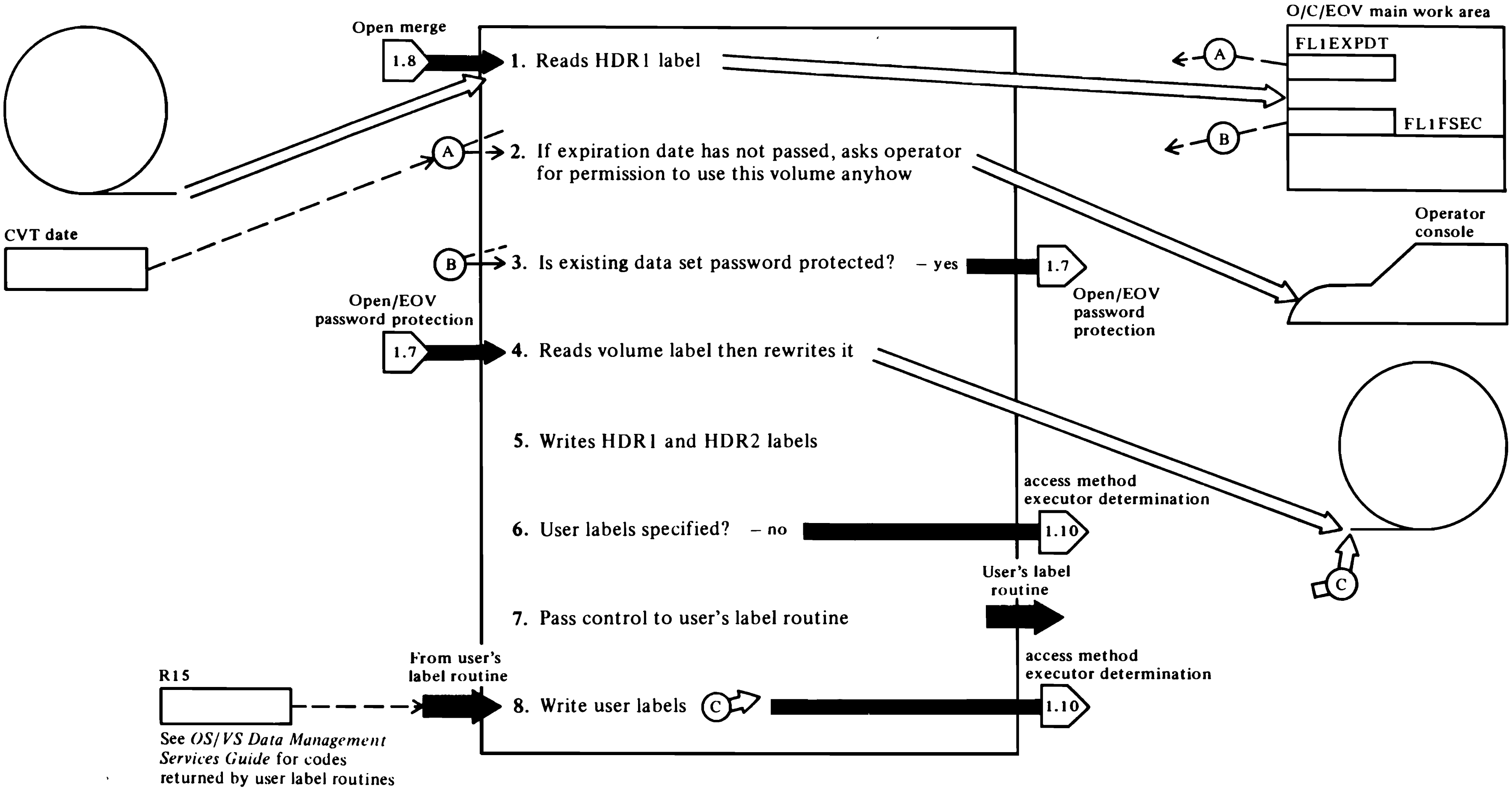
4 Checks for a tape data set being opened for output, determines label processing required, and transfers control accordingly. For tape input data sets, label processing was done previously. [OMG46860]

IFG0196P (alias for IFG0555J)

5 For direct-access data sets with user labels, this routine passes control to the user's output header label routine if there is one marked as active in the DCB exit list. Before control is transferred to the user's routine, the DCB in protected storage is copied to the user's storage. The user's label routine builds output user labels and returns control. [ULSYN6P]

When control is returned from the user's label routine, this routine writes user labels or terminates label processing, based on the contents of register 15 passed by the user's label routine. [ULSYNCH3]

Diagram 1.9. Open Tape, SL Output



1. Reads HDR1 label
2. If expiration date has not passed, asks operator for permission to use this volume anyhow
3. Is existing data set password protected? – yes
4. Reads volume label then rewrites it
5. Writes HDR1 and HDR2 labels
6. User labels specified? – no
7. Pass control to user's label routine
8. Write user labels

See *OS/VS Data Management Services Guide* for codes returned by user label routines

Open merge
1.8
CVT date
A
B
Open/EOV password protection
1.7
R15
From user's label routine
C
O/C/EOV main work area
FL1EXPDT
FL1FSEC
A
B
Operator console
1.7
Open/EOV password protection
access method executor determination
1.10
User's label routine
access method executor determination
1.10
C

Diagram 1.9. Open Tape, SL Output

This function is entered to complete standard label processing for tape data sets being opened for output. It is entered after the tape has been positioned to the HDR1 label. It is entered after the merge processing (JFCB-to-DCB, user's DCB exit, and DCB-to-JFCB) so that the user can be sure that the DCB and JFCB are updated before using information from these blocks in the user labels.

IFG0196N

1 Reads HDR1 label, verifies that it is a HDR1 label, and translates ASCII labels to EBCDIC. [OTA38300]

This routine finds the 17 or fewer characters in the JFCBDSNM field and compares them to the HDR1 data set name field. [OTA37850]

IFG0196Q

2 Checks current date (from CVT) with the expiration date specified in the HDR1 label. [OTA41100]

IFG0194J

If the expiration date has not passed, this routine writes the IEC507D message to the operator to get permission to use the data set anyhow.

IFG0196Q

If the operator does not want to use the current data set, this routine returns to the volume mount and verification routines (IFG0194A) to see if another volume can be mounted. [OTA42800]

3 Determines whether the current data set is password protected. Since the new data set is being opened for output, the current data set will be destroyed. Before destroying the current data set, this routine passes control to the Open/EOV password-protection function to verify that the current data set can be destroyed. [OTA41600]

4 This routine receives control either (1) when the current data set on the volume is not password protected or (2) when the password has been verified. It reads the current volume label, updates it as required, using information from the UCB, and writes the new volume label. [OTA41800]

IFG0196T

5 If PROTECT=YES was specified, the tape volume is defined to RACF and the PROTECT=YES indicator in the JFCB is set to zero. The DEB extension is set to indicate the current volume is RACF-defined.

Using information from the UCB for the unit and TIOT and JFCB for the data set, this routine constructs the HDR1 and HDR2 labels and writes them on the volume. Contents of the HDR1 and HDR2 labels is described in *OS/VS Tape Labels.* [OTA43100]

6 Determines whether user labels are specified for the data set. If not, goes to the access method executor determination function. [OTA45300]

IFG0196U

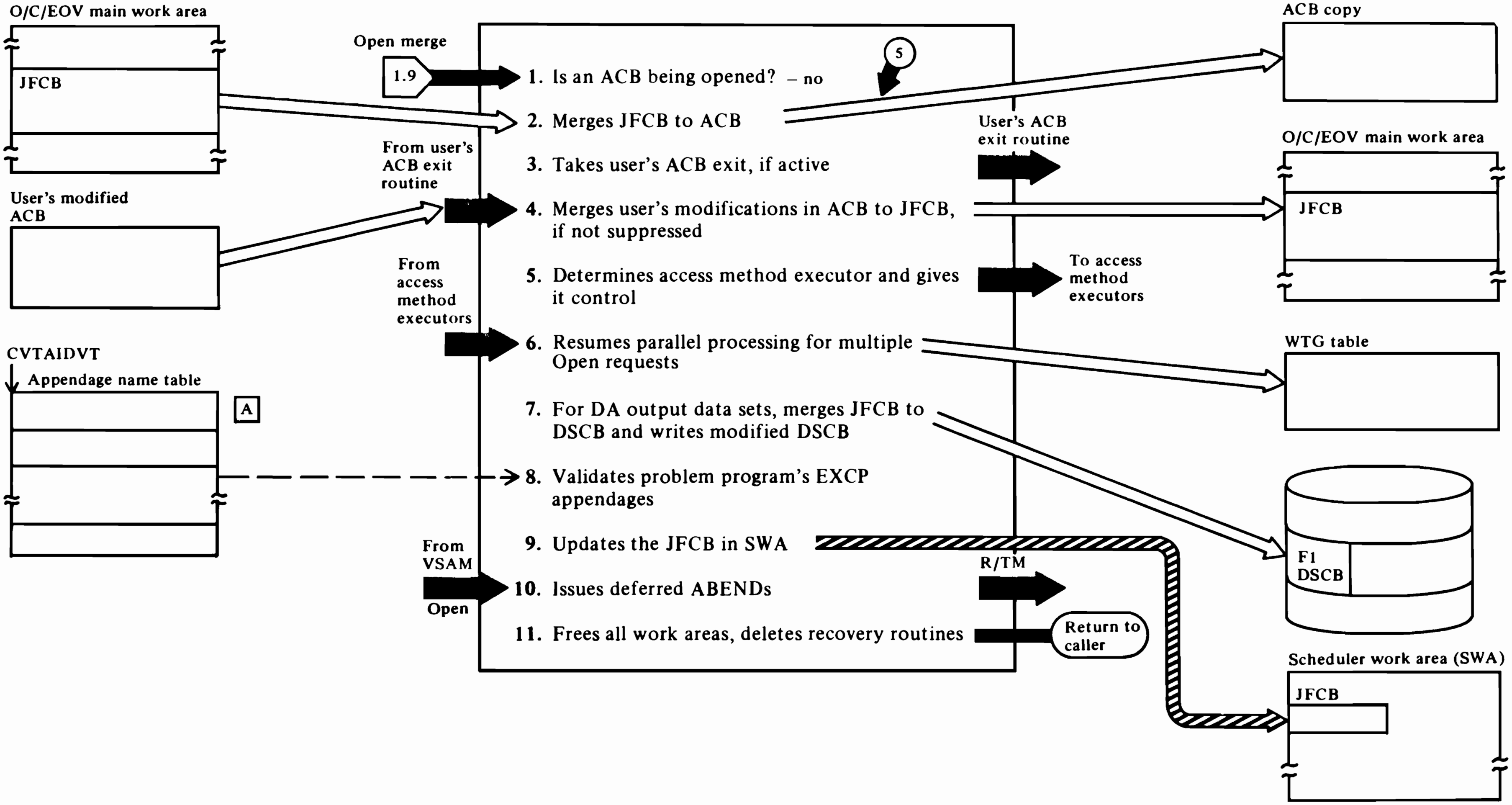
7 If the user's DCB exit list contains an active UHL address, this routine gets virtual storage in which the user's labels will be constructed, builds the user label routine interface, copies the DCB from protected storage to user storage and transfers control so that the user can construct a label. [OTA46300]

8 When control is returned from the user's UHL routine, this routine determines whether to write the label or to terminate user label processing, based on the contents of register 15 passed by the UHL routine. The codes are described in *OS/VS Data Management Services Guide*. [OTA46900]

When all labels have been written, this routine writes a tapemark after the last label, frees virtual storage gotten for user label processing, and transfers control to the access method executor determination function. [OTA47600]

When eight labels have been written, or when the user indicates that no more labels are to be written, this routine transfers control to the access method executor determination function. [OTA47800]

Diagram 1.10. Open Access Method Executor Determination and Final Processing



1. Is an ACB being opened? – no
2. Merges JFCB to ACB
3. Takes user's ACB exit, if active
4. Merges user's modifications in ACB to JFCB, if not suppressed
5. Determines access method executor and gives it control
6. Resumes parallel processing for multiple Open requests
7. For DA output data sets, merges JFCB to DSCB and writes modified DSCB
8. Validates problem program's EXCP appendages
9. Updates the JFCB in SWA
10. Issues deferred ABENDs
11. Frees all work areas, deletes recovery routines

Diagram 1.10. Open Access Method Executor Determination and Final Processing

This function is entered following the JFCB-to-DCB and DCB-to-JFCB merges and after label processing is complete.

IFG0196V

1 This routine determines which access method executor routines are to receive control, based on (1) whether a DCB or an ACB is being opened and (2) the DSORG and MACRF of the data set to be processed. The control block type, DSORG, and MACRF represented by each parameter list entry are examined and the module ID of the appropriate open executor is entered in the WTG table. [OMG92000]

For VTAM ACBs, no access method processing is necessary. The WTG entry for VTAM ACBs is set to zeros. [OMG91200]

Parallel processing for multiple DCBs and ACBs is suspended in this routine, to be resumed when the last access method executor returns control to IFG0196W (alias IGG0190S), the access method return routine. The access method executors also use the WTG table (1) to determine which DCB to process next and (2) to transfer control to the appropriate access method executor string.

2 For SYSIN and SYSOUT ACBs the CCTYPE, LRECL, BLKSIZE, and RECFM fields of the JFCB are merged into the corresponding fields of the ACB that contain zeros. [OMG90450]

3 If an ACB exit routine has been provided and is active, this routine copies the ACB from protected storage to the user's storage and transfers control to the user's ACB exit routine, so that the user can modify the ACB before beginning to process the data set. When control is returned to this routine, the user's ACB is copied back into protected storage. [OMG90495]

4 The JFCBTSDM field is tested to determine whether the user wants to merge from his ACB to the JFCB. If he wants the merge, this routine merges CCTYPE, LRECL, BLKSIZE, and RECFM. [OMG90520]

5 Puts an access method executor ID in the WTG table entry for each DCB, copies the DCB or ACB from protected storage to the user's storage, and when all parameter list entries have been processed, transfers control to the first open executor in the WTG table. When each open access method executor string completes the processing of a DCB or ACB, it transfers control to the module identified in the next WTG table entry. The last entry in the WTG table is the module name of the access method executor return routine, IFG0196W (alias IGG0190S). See Chart 1.2 of the "Program Organization" section for a list of the access method executor modules that can receive control from Open. [OMG90580]

IFG0196W, whose alias is IGG0190S

6 This routine reestablishes the WTG table entries so that parallel processing of multiple-DCB open requests can be resumed. [OFN00400]

If the access method executors did not chain the DEB extension from the DEB, this routine does so. [OFN00900]

7 For output data sets on direct access devices represented by a DCB, this routine merges JFCB fields into the format-1 DSCB. The fields merged are OPTCD, RECFM, KEYLEN, DSORG, BLKSIZE, LRECL, and RKP. The merge is performed for (1) OUTPUT and OUTIN options, (2) the UPDAT option if DSORG=DA in both the DSCB and JFCB, (3) if DSORG=IS and load mode is being used. [OFN20000]

If the format-1 DSCB has been modified during open processing, this routine writes it back to the VTOC. [OFN22000]

IFG0196X

8 When the DCB specifies that the EXCP access technique is being used (MACRF=E) and appendages are specified, this routine determines whether the problem program is authorized (i.e., executing in a system storage key (key 0 through 7), or authorized under APF). If authorized, this routine enters the appendage IDs in the AVT. If not authorized, it compares the IDs with the list of authorized appendages contained in the IEAAPP00 member of SYS1.PARMLIB before entering the appendage IDs in the AVT. The IEAAPP00 member is read into a storage area called the EXCP appendage name table during system initialization. [OFN41200]

Restores the Open option byte in each entry of the Open parameter list for DCBs that are still being opened. [OFN50000]

If the JFCDISP flags were saved by IFG0193A (see Diagram 1.1), IFG0196X restores the JFCDISP flags.

A Format of EXCP appendage name table

+0(0)	No. of Unused Bytes	Table Size
+4(4)	No. of EOE APP. IDs	Offset to 1ST EOE ID
+8(8)	No. of SIO APP. IDs	Offset to 1ST SIO ID
+12(C)	No. of PCI APP. IDs	Offset to 1ST PCI ID
+16(10)	No. of CHE APP. IDs	Offset to 1ST CHE ID
+20(14)	No. of ABE APP. IDs	Offset to 1ST ABE ID
+24(18)	Appendage ID$_1$	Appendage ID$_2$
+28(1C)	Appendage ID$_3$	Appendage ID$_4$
	Appendage ID$_{n-1}$	Appendage ID$_n$

9 Updates the JFCB in the scheduler work area (SWA). This is done so that any modification made to the JFCB during Open processing will be reflected when trailer labels are written at EOV or Close or when the data set is reopened. [OFN50000]

IFG0198N

10 The VSAM open string returns to this module when it has completed VSAM open processing. Only portions of the Open initialization and final processing are performed for VSAM data sets.

This routine sets a bit in the DSAB to indicate an open data set and increments the use count in the DSABOPCT field. [OFN73810]

This routine restores the option byte of each parameter list entry of each DCB/ACB that is no longer being opened. [OFN70000]

Synchronizes processing of all DCBs and ACBs to this point in processing. [OFN70400]

Modifies the DSABOPCT field of the DSAB for the data set to indicate that it is open and modifies the

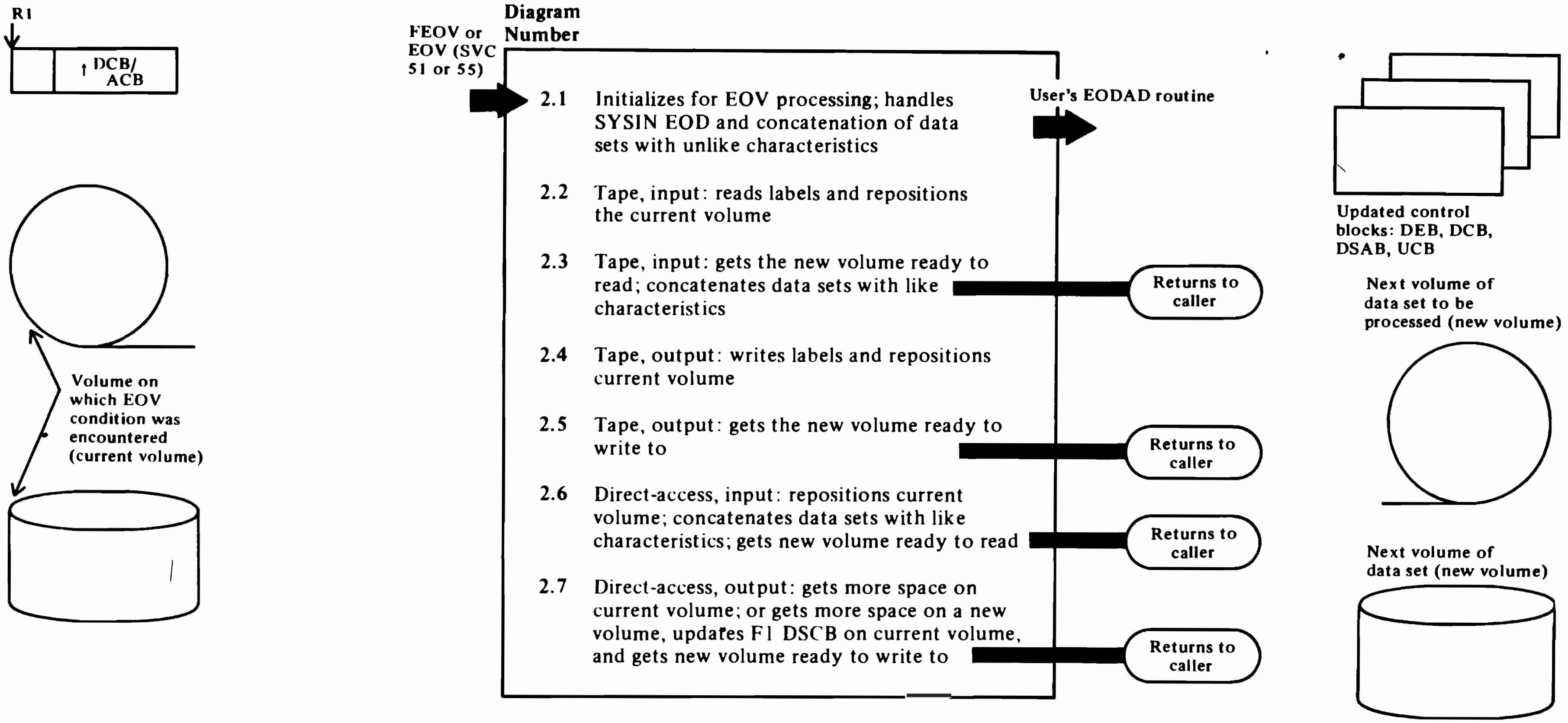
DSABs of data sets opened for input or update. [OFN73810]

If a checkpoint data set security interface exists, this routine updates the DCB/ACB in the caller's storage to the status of the DCB in protected storage and moves the address of the security interface error routine into the READ, WRITE, and CHECK routine address locations in the DCB. [OFN73930]

If an abnormal termination has been deferred by the user's DCB ABEND exit routine so that the balance of a multiple open could be completed, this routine issues the ABEND macro to transfer control to the VS2 recovery/termination manager (R/TM). See *OS/VS2 Scheduler and Supervisor Logic* for a description of the R/TM function. [OFN75400]

11 Frees virtual storage obtained for all work areas obtained in previous processing, loads a return code in register 15, and returns to caller. [OFN76400]

Diagram 2.0. EOV Overview



2.1 Initializes for EOV processing; handles SYSIN EOD and concatenation of data sets with unlike characteristics

2.2 Tape, input: reads labels and repositions the current volume

2.3 Tape, input: gets the new volume ready to read; concatenates data sets with like characteristics

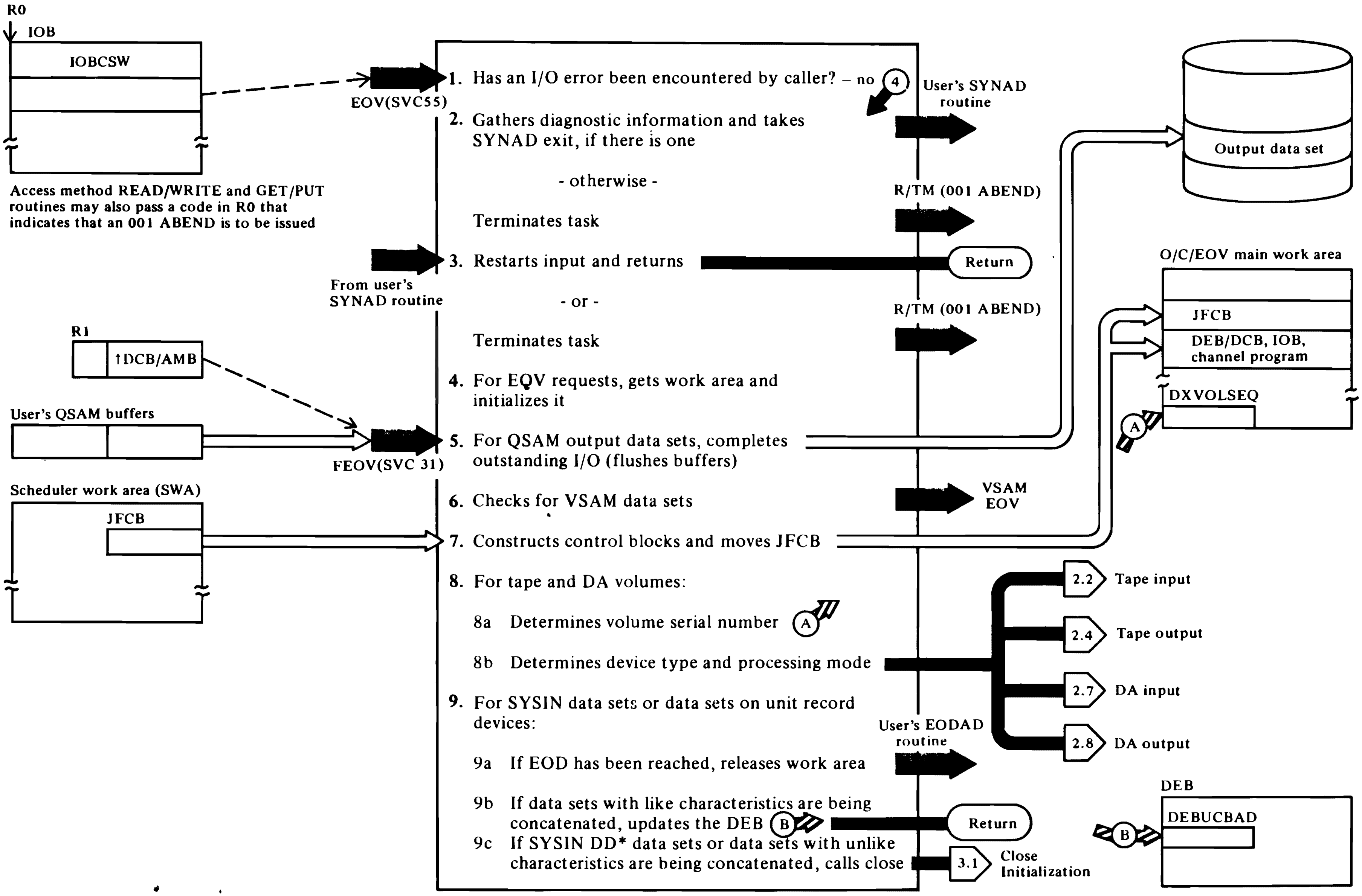
2.4 Tape, output: writes labels and repositions current volume

2.5 Tape, output: gets the new volume ready to write to

2.6 Direct-access, input: repositions current volume; concatenates data sets with like characteristics; gets new volume ready to read

2.7 Direct-access, output: gets more space on current volume; or gets more space on a new volume, updates F1 DSCB on current volume, and gets new volume ready to write to

Diagram 2.1. EOV (FEOV) Initialization



Access method READ/WRITE and GET/PUT routines may also pass a code in R0 that indicates that an 001 ABEND is to be issued

1. Has an I/O error been encountered by caller? – no (4)
2. Gathers diagnostic information and takes SYNAD exit, if there is one

   - otherwise -

   Terminates task
3. Restarts input and returns

   - or -

   Terminates task
4. For EQV requests, gets work area and initializes it
5. For QSAM output data sets, completes outstanding I/O (flushes buffers)
6. Checks for VSAM data sets
7. Constructs control blocks and moves JFCB
8. For tape and DA volumes:

   8a Determines volume serial number (A)

   8b Determines device type and processing mode
9. For SYSIN data sets or data sets on unit record devices:

   9a If EOD has been reached, releases work area

   9b If data sets with like characteristics are being concatenated, updates the DEB (B)

   9c If SYSIN DD* data sets or data sets with unlike characteristics are being concatenated, calls close

Diagram 2.1. EOV (FEOV) Initialization

The EOV component is entered from the SVC second-level interruption handler (SLIH) as a result of the EOV or FEOV macro instructions, which expand to issue an SVC 55 and SVC 31, respectively. The EOV macro instruction (SVC 55) is issued by the BSAM and QSAM processing routines and by other programs using the EXCP access technique (1) to get the next volume of a multivolume data set mounted, (2) to get the next data set ready to process when two or more input data sets are being concatenated(see *OS/VS2 JCL* for a definition of concatenation of data sets), and (3) to obtain more space to continue writing an output data set. SVC 55 is also used by the SAM, BDAM, and ISAM processing routines to pass control to the user's SYNAD routine and to issue a system 001 ABEND (permanent I/O error).

The primary purpose of the EOV macro is to (1) indicate an end-of-data condition so that the problem program's end-of-data (EODAD) routine will receive control on input and (2) to assure that the next volume of a physical-sequential data set is mounted, verified, and ready to read from or write to. It is used for this purpose by all of the system access method routines that process sequential data sets: BSAM, QSAM, and VSAM.

No EOV processing is provided for VSAM data spaces or data sets in the descriptions of the EOV functions that follow. See *OS/VS2 Virtual Storage Access Method (VSAM) Logic* for descriptions of how the EOV condition is handled for VSAM data sets.

The FEOV (force EOV) macro instruction (SVC 31) is used by programs processing physical sequential data sets to get the next volume mounted, verified, and ready to process.

IGC0005E—EOV (SVC 55)

1 This routine is entered for four purposes: (1) to abnormally terminate a task (steps 2 and 3); (2) to gather diagnostic information to pass to the user's SYNAD routine (step 2); (3) to restart I/O operations in response to the return code passed from the user's SYNAD routine, (step 3); and (4) to process an EOF or EOV condition (step 4). At entry to the EOV/SYNAD function, a test is made to determine whether the caller of EOV is entering to abnomally terminate a task. Two ways are used to terminate a task: if the caller passes X'00 00 10 00' in register 0, a message is issued and an ABEND macro with an 001 system code is issued by IFG0551D; for other system codes, the DMABCOND macro is used to call the EOV ABEND interpretation routine (IFG0550P) before issuing the ABEND (see Diagram 6.1). See also *OS/VS2 SAM Logic* for descriptions of the routines that issue the EOV macro to abnormally terminate a task, to obtain diagnostic information, and to signal EOV and EOF.

This routine tests the IOBCSW unit exception bit to determine whether a permanent I/O error has been encountered by the calling routine. If not, EOV processing commences at step 4. If, however, an I/O error has been encountered, this routine continues to step 2 to gather diagnostic information or to terminate the task. [EIN01200]

IFG0551B

2 When the access method routine issues an EOV macro because a permanent I/O error has been encountered, this routine determines whether the user has provided a SYNAD routine. If a SYNAD routine is active, this routine sets codes in register 0 and register 1 (see *OS/VS Data Management Macro Instructions* for the codes passed) then exits to the access method routine that issued the SVC 55. If no SYNAD routine is provided and the data set is being processed using BSAM, an immediate 001 ABEND is issued. If QSAM is being used, the error option specified by the user is used (see step 3). [EIN05700]

3 When the user's SYNAD routine returns control to the access method routine, the access method routine issues another EOV macro, and this routine is given control. This routine will restart I/O operations if: (1) the data set is being read (an input data set) and (2) if the access method is QSAM, the *accept* or *skip* option is coded in the DCBEROPT field. Otherwise, IFG0551D is given control to issue an 001 ABEND. [EIN06600]

IFG0551D

Builds a message containing job name, step name, etc., and issues the ABEND. [EIN08400]

IGC0005E

4 Gets O/C/EOV main work area, initializes it, and copies user's DCB into protected storage. Note that VSAM (VS2 Master) catalog requests make use of an AMB instead of an ACB. The AMB is obtained in protected storage and therefore need not be copied. [EIN01700]

If end-of-data was reached for a BPAM data set (DSORG=PO), this routine passes control to the user's EODAD routine, or if no EODAD routine was supplied, abnormally terminates the task. [EIN02200]

IGC0003A—FEOV (SVC 31)

5 This routine is entered via SVC 31 (FEOV macro instruction) to get the next volume of a data set mounted, verified, and ready to process. FEOV is not supported for SYSIN, SYSOUT, dummy, VSAM or VTAM data sets; requests for FEOV processing for these kinds of data sets result in an immediate return to the caller. [EIN03200]

Gets O/C/EOV main work area, initializes it, and copies the user's DCB into protected storage. [EIN03420]

For QSAM data sets opened for output (or update), this routine ensures that all I/O requests against the DCB have been completed. [EIN03650]

IFG0551F

6 If the EOV request was issued for a VSAM catalog (AMBDSORG=X"0008," AMBID=X"40," and AMBINFL=X"10"), this routine transfers control to IFG0550Y to provide the JFCB for the catalog. IFG0550Y returns to this routine. If a VSAM data set is being processed, but is not a VSAM master or user catalog, this routine copies the JFCB from the SWA to the O/C/EOV main work area. [EIN10800]

If the EOV request is for a VSAM data set (or for a VSAM catalog), this routine transfers control to IFG0557A, the first load of the VSAM EOV component. Unlike Open and Close processing, VSAM EOV does not return to this EOV component when it has completed processing the EOV request for the VSAM data space, because only one ACB/DCB at a time is processed with each issuance of the EOV macro. [EIN11251]

7 This routine constructs a DEB/DCB and IOB to be used in subsequent I/O operations, such as reading and writing the DSCB, writing tape marks, labels, end-of-file marks, etc. This routine also copies the JFCB for the data set from SWA to the O/C/EOV main work area. A PURGE macro instruction is issued to halt any I/O scheduled or started using the current DEB. This DEB is disconnected from the user's DCB and connected to the DCB copy. Hence, if an attempt is made to start I/O against this DEB, the DEB validity check will fail and a 16E ABEND will result. At conclusion of EOV processing, the DEB is reconnected to the user's DCB if the EOD has been

reached, or the DEB is released if EOV or EOD with concatenation conditions are encountered. [EIN11250]

If user-totaling has been specified, this routine adjusts the count in the user-totaling save area so that it reflects the last record on the volume (for EOV conditions) or the last record of the data set (for EOF conditions). [EIN11800]

IFG0551H

8a For data sets on tape and direct-access devices, this routine examines the volume serial numbers in the JFCBVOLS field, looking for the volume serial number that matches the volume serial number in the UCB associated with the volume on which the EOF or EOV condition was encountered. As many as five volume serial numbers are contained in the JFCB and as many as 15 more in each JFCB extension. [EIN13600]

When the volume serial number is located, its relative position in the JFCB is saved in the DXVOLSEQ field of the main work area to indicate which volume of the data set is currently being processed. [EIN13700]

8b This routine determines the next EOV function based on device type and on whether the data set is opened for input or output. For data sets on tape, determines label type. [EIN15600]

IFG0552X

9a This routine is entered when (1) EOD is reached for a SYSIN data set or a data set on a unit record device and (2) no concatenation is in progress. This routine first releases the work areas obtained in previous EOV processing, then alters the old PSW to transfer control to the users's EODAD routine. [ECM60500]

9b Data sets with like characteristics are those that can be processed correctly using the same DCB, IOB, and channel program. This means, generally, that the DCB must be opened for physical-sequential processing, and that the device type, record format, and blocksize of the data sets must be the same. Note also that concatenated SYSIN data sets are treated as if they were data sets with unlike characteristics.

IFG0551F

Copies the JFCB for the next data set in the concatenation from the SWA to the O/C/EOV main work area. [EIN11250]

IFG0551H

Determines that module IFG0552X is to receive control when SYSIN data sets are being processed. [EIN155500]

IFG0552X

For concatenated, like data sets on unit record devices, this routine puts a UCB address in the DEB. This UCB address represents the unit from which the next data set of the concatenation will be read. [ECM61600]

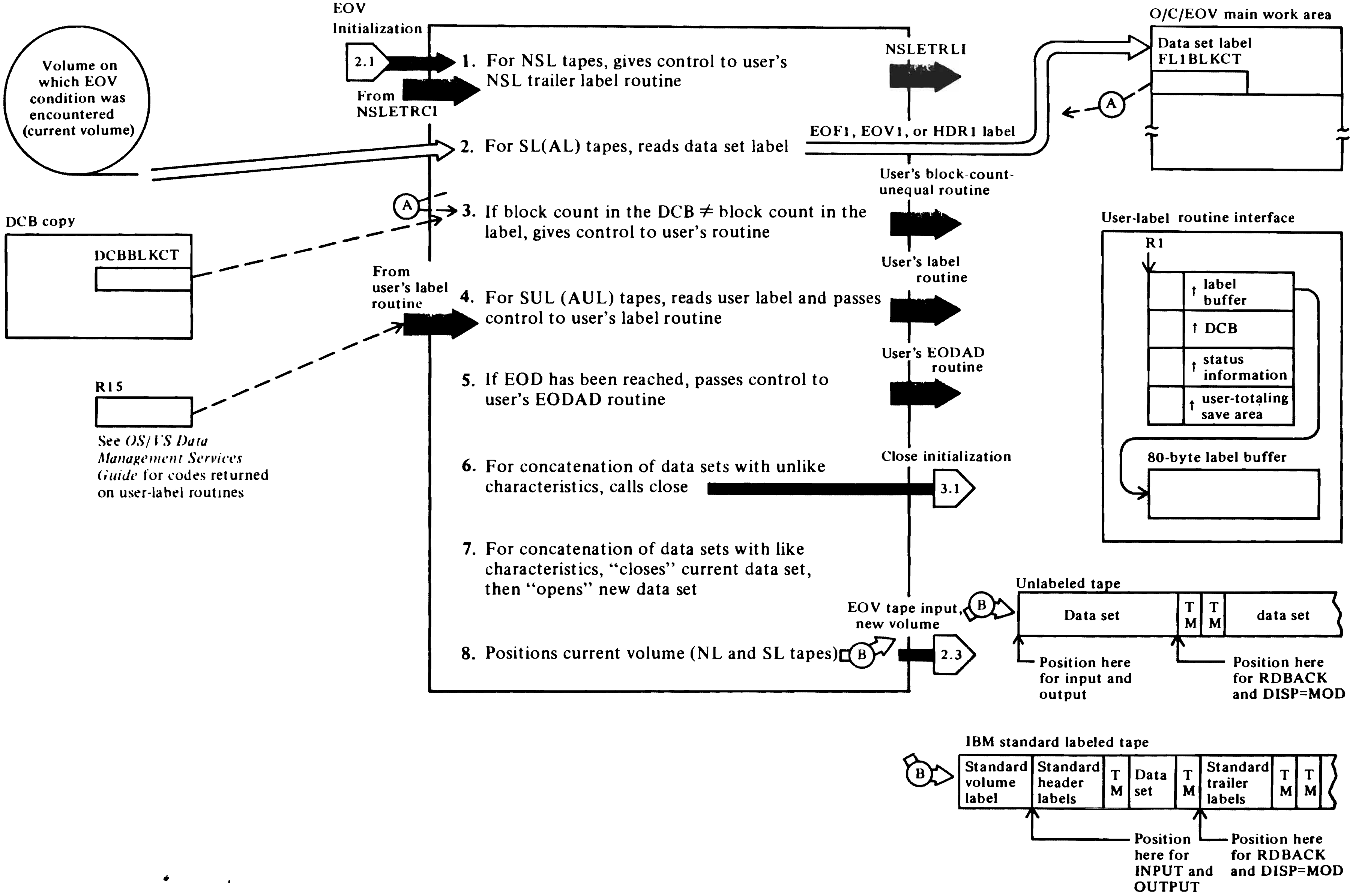
Data sets on tape and DA devices with like characteristics are concatenated as shown in the EOV tape input and direct-access input functions (Diagrams 2.2 and 2.6).

9c If the data sets have unlike characteristics or the current or next data set in the concatenation is a SYSIN data set, this routine passes the address of the current DCB in register 1, along with the work areas, to the second entry point in the initial load of the Close component (see Diagram 3.1). At the conclusion of Close processing of the current DCB, the Close final function passes control to the second entry point in the first load of the Open component (see Diagram 1.1) to reopen the DCB with the characteristics of the new data set. [ECM61900]

IFG0551L

This routine is entered when the sequential access methods (BSAM or QSAM) are being used. Reschedules any I/O requests that were purged during previous EOV processing; primes buffers, if required. Exits to the calling routine, a QSAM synchronizing routine or a BSAM check routine. [EIN02290]

Diagram 2.2. EOV Tape Input—Current Volume



1. For NSL tapes, gives control to user's NSL trailer label routine

2. For SL(AL) tapes, reads data set label

3. If block count in the DCB ≠ block count in the label, gives control to user's routine

4. For SUL (AUL) tapes, reads user label and passes control to user's label routine

5. If EOD has been reached, passes control to user's EODAD routine

6. For concatenation of data sets with unlike characteristics, calls close

7. For concatenation of data sets with like characteristics, "closes" current data set, then "opens" new data set

8. Positions current volume (NL and SL tapes)

See *OS/VS Data Management Services Guide* for codes returned on user-label routines

Diagram 2.2. EOV Tape Input—Current Volume

When an input data set on a tape device is being processed, this function is entered to handle labels and positioning of the volume on which the EOV or EOF condition was encountered (this volume is called the *current* volume). When this function is completed, the next volume to be processed (called the *new* volume) must be mounted, verified, and positioned to be read.

IFG0551H

1 A test is made to determine whether NSL processing is to be deferred until the data set is closed. If not, an exit is taken to the user's EOV input trailer label processing routine, NSLETRLI. Before transferring control to NSLETRLI, the copy of the DCB in protected storage is copied into the user's storage. [EIN15950]

IFG0552R

2 This routine is entered to read the EOF1 or EOV1 (HDR1 if the tape is being read backward) label and verify that the block count in the DCB is equal to the block count in the label. [ETI0400]

3 If the block count in the DCB is not equal to the block count in the data set label, this routine copies the DCB in protected storage to the user's storage and takes the exit to the user's block-count-unequal routine.

IFG0553B (alias for IFG0202H)

If data set activity is being monitored, this routine constructs a type-14 SMF record and writes it to the SMF data set. [CCM02040]

IFG0552T

4 If the data set has standard user labels (SUL or AUL), this routine is called to read in the labels and pass control to the user's label routine so that the labels can be verified. [ETI04100]

Positions tape to the logical end-of-file or end-of-volume: after the tape mark that delimits the user trailer labels if the tape was being read forward, or at load point if the tape was being read backward. [ETI05900]

Determines whether EOF or EOV condition was encountered for this request. [ETI06100]

IFG0552X

5 This routine is entered when EOV has been reached and there is no more to read. This routine releases the work areas obtained in previous processing, then transfers control to the user's EODAD routine. [ECM60500]

6 If the data sets with unlike characteristics are being concatenated, this routine passes the address of the current DCB in register 1, along with the work areas obtained during previous EOV processing to the second entry point in the initial load of the Close component (see Diagram 3.1). At the conclusion of Close processing of the current DCB, the Close final function passes control to the second entry point in the first load of the Open component (see Diagram 1.1) to reopen the DCB with the characteristics of the new data set. [ECM61900]

IFG0552V

7 When a data set concatenation (like characteristics) is in process, this routine puts the pointer to the next DSAB in the O/C/EOV main work area, and puts the new DSAB/TIOT pointer in the DEB extension. [ETI11200]

8 If only one tape unit is allocated to the data set or if the unit on which a volume of the data set is to be mounted is not ready, no positioning is done. [ETI08500]

If a positioning option was coded in the FEOV or OPEN macro, this routine positions the current tape according to the option specified. The FEOV options and the resultant action are: "REWIND," rewind the tape, and "LEAVE," position to the logical end of the data set (the first record of the data set if it was being read backward, or the last record of the data set if it was being read forward). The OPEN positioning options are: "REREAD," position to the logical beginning of the data set, and "LEAVE," position to the logical end of the data set. [ETI08700]

If neither FEOV or OPEN positioning options were specified, the disposition specified or implied in the DD statement is used to position the data set. [ETI10500]

During tape positioning, the UCB file sequence number and count are increased by one each time a tape is positioned to the end of the data set (following trailer labels). [ETI11500]

Decreases data management count by 1 (sets it to 0), then determines whether (1) data sets are being concatenated or (2) an EOV condition is being processed. [ETI11000]

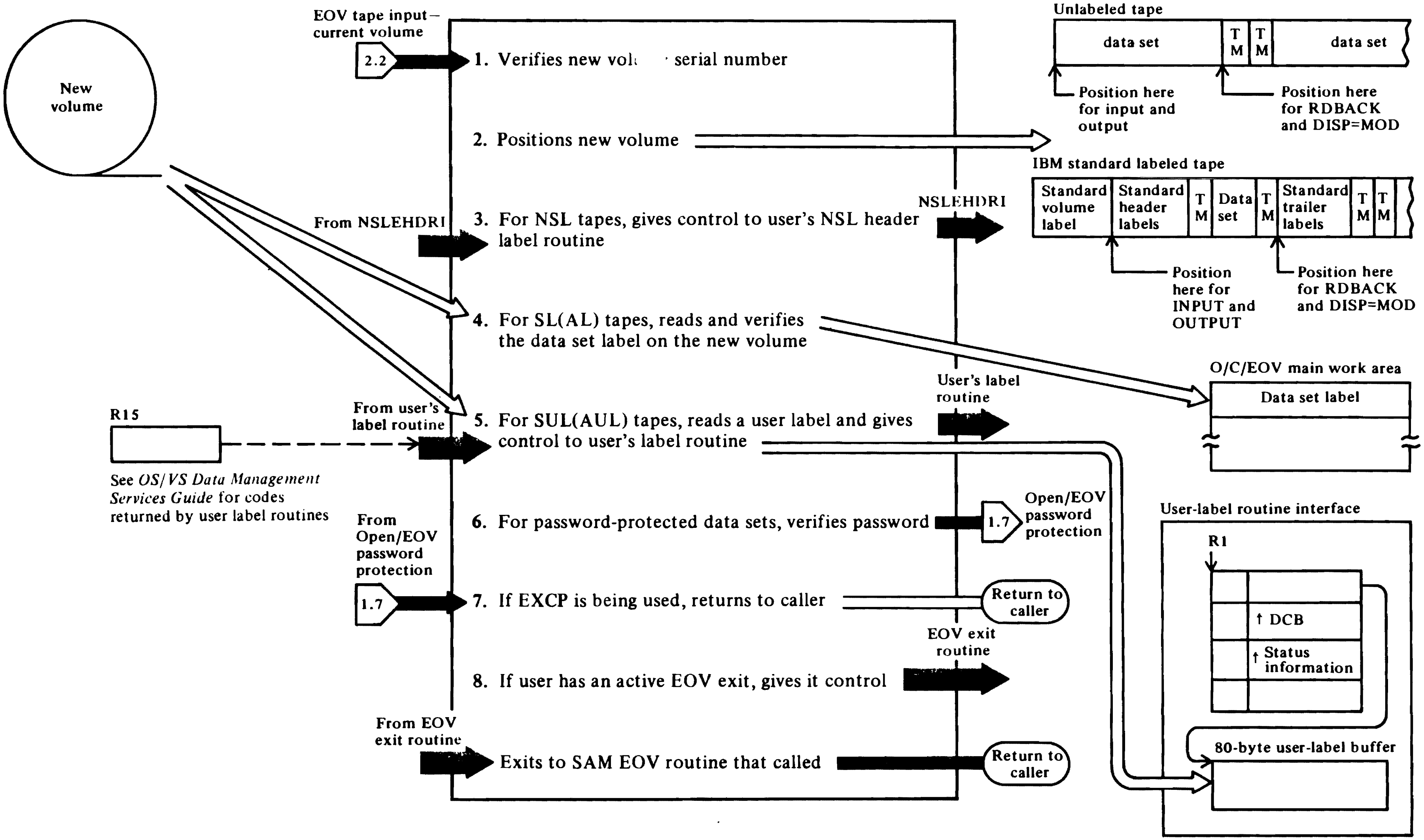
IFG0551F

Copies the JFCB for the new data set in the concatenation into the O/C/EOV main work area. [EIN11250]

IFG0551H

Copies the volume sequence number in the JFCB (JFCBVLSQ) to the O/C/EOV main work area (DXVOLSEQ) so that the new volume can be verified or mounted and verified (see Diagram 2.2). [ETI13700]

Diagram 2.3. EOV Tape Input—New Volume



1. Verifies new volume serial number
2. Positions new volume
3. For NSL tapes, gives control to user's NSL header label routine
4. For SL(AL) tapes, reads and verifies the data set label on the new volume
5. For SUL(AUL) tapes, reads a user label and gives control to user's label routine
6. For password-protected data sets, verifies password
7. If EXCP is being used, returns to caller
8. If user has an active EOV exit, gives it control

Exits to SAM EOV routine that called

See *OS/VS Data Management Services Guide* for codes returned by user label routines

New volume
EOV tape input—current volume
2.2
From NSLEHDRI
From user's label routine
R15
From Open/EOV password protection
1.7
From EOV exit routine
NSLEHDRI
User's label routine
1.7
Open/EOV password protection
Return to caller
EOV exit routine
Return to caller
Unlabeled tape
data set
T M
T M
data set
Position here for input and output
Position here for RDBACK and DISP=MOD
IBM standard labeled tape
Standard volume label
Standard header labels
T M
Data set
T M
Standard trailer labels
T M
T M
Position here for INPUT and OUTPUT
Position here for RDBACK and DISP=MOD
O/C/EOV main work area
Data set label
User-label routine interface
R1
↑ DCB
↑ Status information
80-byte user-label buffer

Diagram 2.3. End-of-Volume, Tape Input—New Volume

This function is entered at the conclusion of processing of the volume on which the EOV or EOF condition was encountered, the *current* volume (from Diagram 2.2).

IFG0554A (alias for IFG0194A)

1 This routine is entered during EOV processing to (1) ensure that the second and subsequent volumes of multivolume tape data sets are mounted and (2) when data sets on tape are being concatenated, to verify or mount and verify the the first volume of the second and subsequent data sets of the concatenation. Using the volume sequence number (DXVOLSEQ field), which has been changed to indicate that the next volume is to be processed. this routine searches the JFCB and its extension to find the volume to be mounted next (referred to as volume *n* in the source code. If multiple volumes are being processed, the volume serial number of the next volume (referred to as volume *n* + *1* in the source code) is also determined so that a look-ahead mount message can be issued to the operator. [VUS03300]

Finds a tape unit allocated to this task that is available for use with this volume (note that the volume may already be mounted). Adds one to the data management count in the UCBDMCT field of the UCB to indicate that the unit has been selected. [VUS05120]

IFG0554J (alias for IFG0194J)

If the volume is not mounted, this routine writes a message to the operator requesting that he mount the volume. [WTO02000]

IFG0554F (alias for IFG0194F)

This routine attempts to read the first record on the tape and determines which of the following label types is being processed: standard label (SL), ASCII standard label (AL), or unlabeled (NL). The label type specified in the JFCBLTYP field is checked to ensure that the correct label type has been read. An NL tape is assumed to be a nonstandard labeled (NSL) tape if JFCBLYTP specifies NSL, in which case volume verification is bypassed. Note: each installation must ensure that this option is not misused; this is done by authorizing a reader procedure that accepts DD statements specifying LABEL=BLP. [TMV02300]

RACF authorization to the verified AL or SL tape is checked. If defined and authorized, sets DEBXDSSI indicating RACF-defined.

IFG0554A (alias for IFG0194A)

The JFCB is updated in the scheduler work area (SWA) when a new volume is accepted.

IFG0554J (alias for IFG0194J)

If more than two volumes are being processed, and volume *n+1* is not mounted, this routine issues the look-ahead mount message (IEC501E) for the next volume to be processed. No verification that the volume is mounted is made until end-of-volume is reached for the preceding volume. [WTO02500]

When volume *n* has been mounted and verified, labels accepted, and look-ahead mount message issued, this routine determines label type and processing mode (input or output) and transfers control accordingly.

IFG0553F

2 Positions the tape to the logical beginning of the data set, according to the direction in which the tape is being read, forward or backward. If the tape has standard labels, tries to get to the HDR1 (EOV1 if reading backward) label. [ET118700]

3 For data sets with nonstandard labels, this routine copies the DCB from protected storage to the user's storage and transfers control to the NSLEHDRI routine to verify the labels. (If the volume is rejected by NSLEHDRI based on the content of the labels, the volume mount and verification function, step 1, is reentered to try to get the right volume mounted.) [ET121200]

4 Reads the label at which the tape is positioned and verifies that it is HDR1 or EOV1 label as expected for the direction in which the tape is being read. [ET123100]

Compares the data set name in the JFCB (last 17 characters) with the data set name in the label just read. [ET123700]

5 If user labels are specified in the JFCB, and a user label exit is active in the DCB exit list, this routine gets storage for the user's labels and reads them in, one at a time, passing control each time, to the user's label routine so the labels can be verified. [ET124600]

Before control is passed to the user's label routine, the DCB in protected storage is copied to the user's storage. [ET125200]

When user label processing is complete, this routine positions the new volume to the first record of the data set on this volume (last record if the tape is being read backward). [ET126100]

IFG0553H

6 If the data set is password protected, determines the protection mode of the data set. If an EOV condition is being processed, the password may have already been verified during open processing of the first volume. If a concatenation is in process, the password will have to be verified for each data set in the concatenation that is password protected. [ET128600]

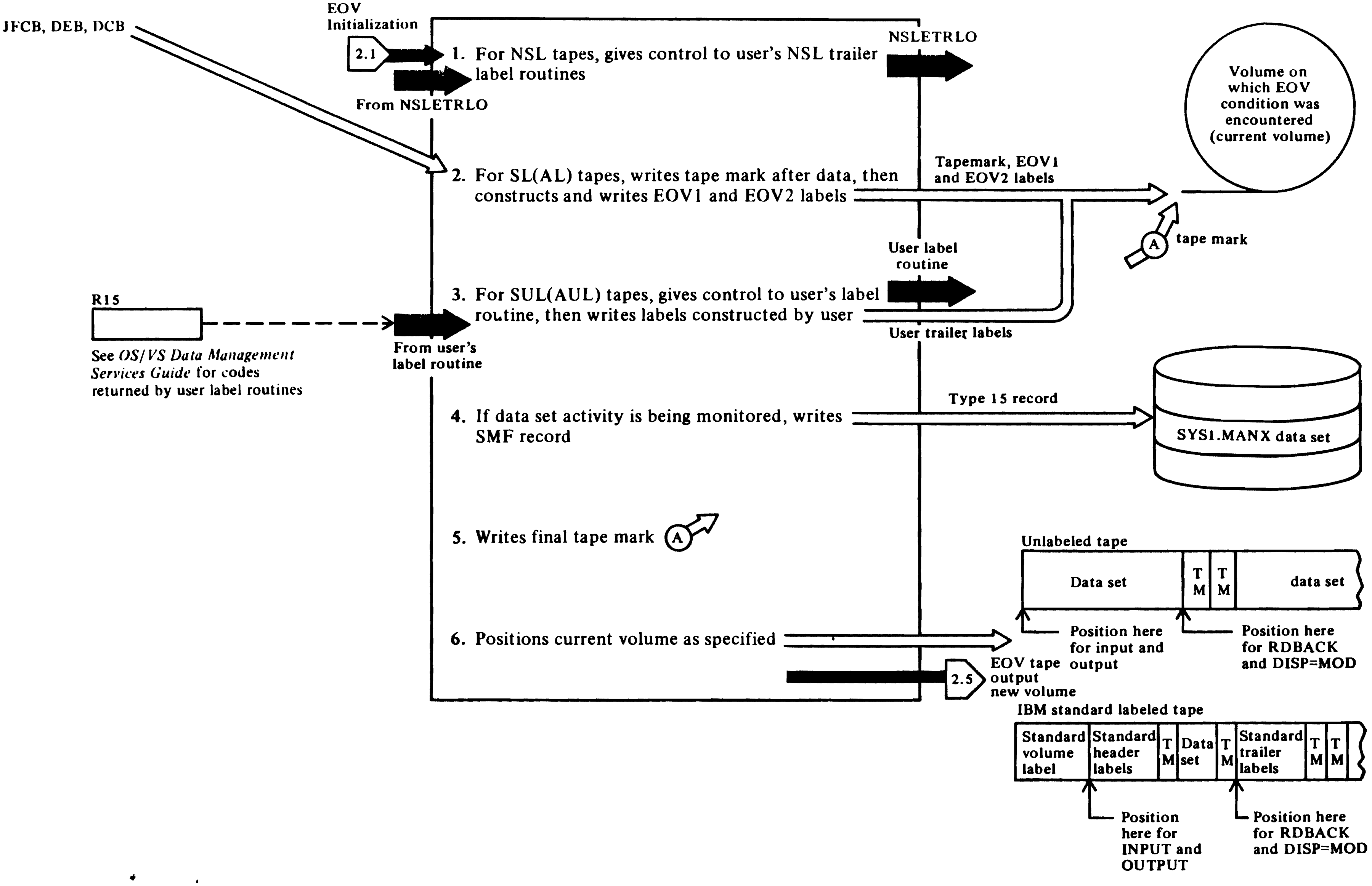
7 The address of the new UCB is put in the DEB and the DEBVLSQ field is modified using the contents of the DXVOLSEQ in the O/C/EOV main work area. EOV tape input processing is now complete. This routine determines whether to release the work areas and return to the caller (EXCP access technique being used) or to transfer control to the SAM EOV executor, IFG0551L (sequential access method being used). [ET130300]

IFG0551L

8 This routine is entered when the sequential access methods (BSAM or QSAM) are being used. If CHKPT=EOV is specified, control is passed to IFG0558X, which takes a checkpoint and then returns. If the user's DCB exit list contains an active EOV exit routine address, the EOV exit routine is given control. [EIN21400]

Reschedules any I/O requests that were purged during previous EOV processing; primes buffers, if required. Exits to the calling routine, a QSAM synchronizing routine or a BSAM check routine. [EIN02290]

Diagram 2.4. EOV Tape Output—Current Volume



1. For NSL tapes, gives control to user's NSL trailer label routines
2. For SL(AL) tapes, writes tape mark after data, then constructs and writes EOV1 and EOV2 labels
3. For SUL(AUL) tapes, gives control to user's label routine, then writes labels constructed by user
4. If data set activity is being monitored, writes SMF record
5. Writes final tape mark
6. Positions current volume as specified

JFCB, DEB, DCB
EOV Initialization
2.1
From NSLETRLO
NSLETRLO
Volume on which EOV condition was encountered (current volume)
Tapemark, EOV1 and EOV2 labels
A
tape mark
User label routine
User trailer labels
R15
From user's label routine
See OS/VS Data Management Services Guide for codes returned by user label routines
Type 15 record
SYS1.MANX data set
A
Unlabeled tape
Data set
T M
T M
data set
Position here for input and output
Position here for RDBACK and DISP=MOD
2.5
EOV tape output new volume
IBM standard labeled tape
Standard volume label
Standard header labels
T M
Data set
T M
Standard trailer labels
T M
T M
Position here for INPUT and OUTPUT
Position here for RDBACK and DISP=MOD

Diagram 2.4. EOV Tape Output—Current Volume

When an output data set on a tape device is being processed, this function is entered to write tape marks and labels and reposition the tape that was being processed when the EOV condition was encountered.

IFG0551H

1 If nonstandard labels are specified, this routine copies the DCB in protected storage to the user's storage and passes control to the user's NSL output trailer label processing routine, NSLETRLO. Steps 2 and 3 of this function are not executed for NSL tapes; however, a final tapemark will be written when control is returned from NSLETRLO. [EIN16500]

IFG0551P

2 Writes a tape mark to delimit the last record of the data on the tape. This tape mark precedes the EOV1 label if an SL tape is being written. [IFG0551P]

Constructs the EOV1 label using information in the JFCB, DEB, and DCB. [ETO00200]

Writes EOV1 label. [ETO01400]

IFG0551R

Constructs the EOV2 label using information from the JFCB, DCB, and TIOT. [ETO02000]

Writes EOV2 label. [ETO03000]

3 When user labels (SUL or AUL) are to be written, this routine gets a buffer in which the user's label routine is to construct the labels, establishes the interfaces, copies the DCB from protected storage to the user's storage, and transfers control to the user's label routine. When control is returned, this routine responds to the contents of register 15, which indicates whether to write another user label or to continue with the next step. [ETO03400]

IFG0553B (alias for IFG0202H)

4 If data set activity is being monitored, this routine builds and writes a type 15 SMF record to the SMF data set. [CCM02040]

IFG0551T

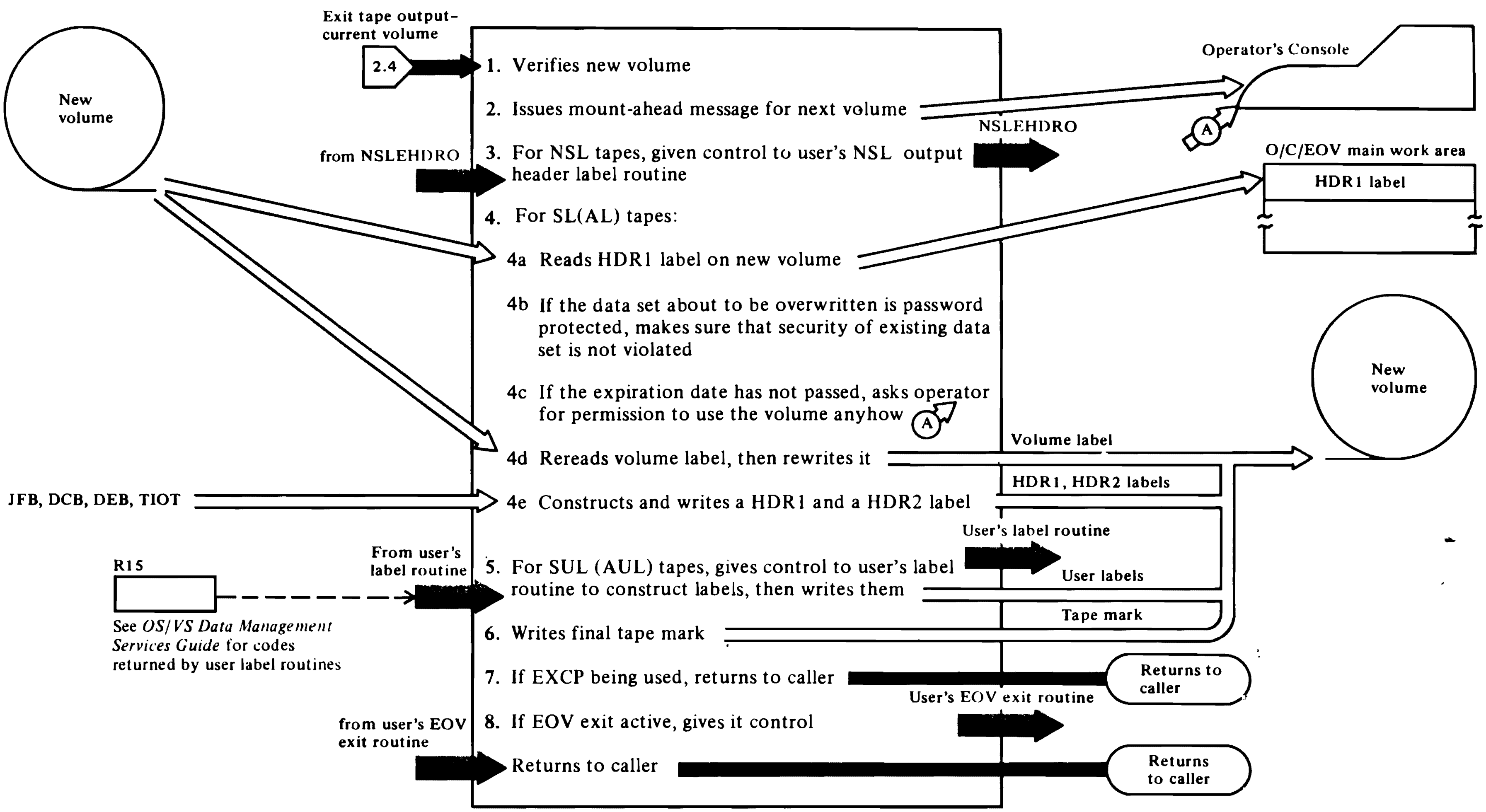
5 For all but NSL tapes, this routine writes a final tape mark to delimit the labels or the da' . on the current volume. [ETO05100]

6 This routine determines whether a positioning option (REREAD, REWIND, or LEAVE) was specified. If so, the volume is positioned accordingly. [ETO05700]

When the tape has been positioned, this routine adds one to the file count and file sequence number in the UCB (UCBFSCT), then sets the data management count (UCBDMCT) to zero. Processing is complete on the current volume. This routine now adds one to the volume sequence number (DXVOLSEQ) in the O/C/EOV work area to tell the volume mount/verify routines which volume is to be processed next. [ETO06300]

This function passes control to the EOV tape ouptut, new volume function (Diagram 2.5).

Diagram 2.5. EOV Tape Output—New Volume



1. Verifies new volume
2. Issues mount-ahead message for next volume
3. For NSL tapes, given control to user's NSL output header label routine
4. For SL(AL) tapes:
   - 4a Reads HDR1 label on new volume
   - 4b If the data set about to be overwritten is password protected, makes sure that security of existing data set is not violated
   - 4c If the expiration date has not passed, asks operator for permission to use the volume anyhow
   - 4d Rereads volume label, then rewrites it
   - 4e Constructs and writes a HDR1 and a HDR2 label
5. For SUL (AUL) tapes, gives control to user's label routine to construct labels, then writes them
6. Writes final tape mark
7. If EXCP being used, returns to caller
8. If EOV exit active, gives it control

Returns to caller

See *OS/VS Data Management Services Guide* for codes returned by user label routines

Diagram 2.5. EOV, Tape Output—New Volume

This function is entered after labels have been written on the current volume and the current volume has been positioned for subsequent processing. A look-ahead mount message will have been issued at the time the data set was opened, so the new volume may already be mounted.

IFG0554A (alias for IFG0194A)

1 This routine uses the volume sequence in the DXVOLSEQ field of the O/C/EOV work area in conjunction with the volume serial numbers in the JFCB and JFCB extension to determine the volume serial number of the next volume to be processed (the *new* volume), referred to in the source code as volume *n*. If another volume is to be processed after the *new* volume, this routine determines its serial number also so that a look-ahead mount message can be issued to the operator to mount this volume. The volume serial number of volume *n-1* is also saved if that volume is to be demounted. [VUS03300]

Finds a tape unit allocated to this task that is available for use with this volume (note that the volume may already be mounted). [VUS05120]

Finds a unit that is compatible with the volume being mounted. [VUS08000]

IFG0554J (alias for IFG0194J)

If the volume is not mounted, this routine writes a message to the operator requesting that he mount volume *n*. [WTO02000]

IFG0554F (alias for IFG0194F)

This routine attempts to read the first record on the tape and determines which of the following label types is being processed: standard label (SL), ASCII standard label (AL), or unlabeled (NL). The label type specified in the JFCBLTYP field is checked to ensure that the correct label type has been read. An NL tape is assumed to be a nonstandard labeled (NSL) tape if JFCBLYTP specifies NSL, in which case volume verification is bypassed. Each installation must ensure that this option is not misused. This can be done by authorizing a reader procedure that accepts DD statements that specify LABEL=BLP. [TMV02300]

RACF authorization to be verified AL or SL tape is checked. If defined and authorized, sets DEBXDSSI indicating RACF-defined.

Several types of label, density, and mode conflicts can occur during volume verification. *OS/VS Tape Labels* describes these conflicts and their resultant actions. This routine checks for these conflicts. [TMV02300]

IFG0553C (EMODVOL1)

This object module performs the same functions as IFG0193C (OMODVOL1). If this routine has not been replaced by an installation-supplied label editor routine (which must be named EMODVOL1), it performs the label editor functions. It is entered when label and density conflicts are encountered during volume verification. The type of label or density conflict is indicated in the JFCBMASK+5 field. An installation-supplied EMODVOL1 routine can resolve the conflict and reset the appropriate bit or the EMODVOL1 routine can reject the volume by setting the high-order bit in the UCBDMCT field of the UCB. EMODVOL1 is entered in key 0 and is passed the address of the copy of the user's DCB in protected storage. The address of the copy of the DCB is passed in register 2. EMODVOL1 routines must return to IGG0550P. Reference information concerning the label editor routines is provided in *OS/VS Tape Labels*. [OMO01000]

IFG0553D (alias for IFG0193D)

This routine can be called by the user's EMODVOL1 routine or by IFG0553C, the standard label editing routine. [DSL00000]

When a standard label is to be destroyed, this routine reads labels until a HDR1 label has been read in. [DSL01100]

Checks RACF ALTER authorization to the volume. If authorized, bypasses password check. If not authorized, requests that another volume be mounted. If not defined, performs password check.

Checks the HDR1 label for password-protected data set, and if one is encountered, issues an error message to the operator, then requests that another volume be mounted. [DSL08200]

Checks expiration date, and if the expiration date has not passed, issues a message to the operator to get permission to ignore the expiration date. [DSL03000]

Issues a message to the operator requesting permission to destroy the label. [DSL05000]

Checks the HDR2 label to ensure that the existing data set is not an existing checkpoint data set. If it is, issues a message to the operator. [DSL06500]

Writes a tape mark over the standard volume label to destroy the label, then sets UCBTFL1 accordingly. [DSL07000]

If tape is RACF-defined, deletes RACF definition of the volume.

IFG0553E (alias for IFG0193E)

Entered when a standard label is to be created, this routine issues a message to the operator to ask him to specify the volume serial number and owner of the data set. [CSL01000]

Verifies operator's specification and completes label. Translates EBCDIC to ASCII for AL tapes. [CSL03500]

Check RACF ALTER authorization to volume and consistency to previous volume. Abnormally terminate if not authorized to specific request. Reissue operator message if not authorized to nonspecific request. Continue if volume not RACF-defined. Set DEBXDSSI bit, if authorized, and continue.

Rewinds tape to load point and writes label. Writes tape mark after HDR1 label. [CSL04000]

IFG0554A (alias for IFG0194A)

When the volume has been mounted and verified, labels accepted, and look-ahead mount message issued, this routine determines label type and processing mode (input or output) and transfers control accordingly.

IFG0554J (alias for IFG0194J)

2 If more than one volume is being processed, and the second volume is not mounted, this routine issues the look-ahead mount message (IEC501E) for the next volume to be processed. No verification that the volume is mounted is made until end-of-volume is reached for the preceding volume. [WTO02500]

IFG0552B

3 Copies the DCB from protected storage to the user's storage and transfers control to NSLEHDRO. [ETO15200]

4a This routine is entered to process an SL tape. The label following the volume label is read and a test is made to verify that it is a HDR1 label. [ETO15800]

4b If the existing data set on the new volume is password protected, this routine makes sure that the security of the the existing data set will not be violated when it is overwritten. If the overwriting would violate the password protection, (that is to say, if the data set has a read-only protection mode indicator, it cannot be written on), control is passed to the volume mount and verification routines to ask the operator to mount another volume. [ETO16200]

IFG0554J (alias for IFG0194J)

4c If the expiration date has not passed, this routine writes the IEC507D message to the operator to get permission to overwrite the data set. [WTO03000]

IFG0552B

4d Rewinds the tape to load point. Constructs a new volume label using inforamtion in the JFCB and the UCB, then writes the label. [ETO17100]

If the previous volume was RACF-protected, the present volume is defined to RACF as part of the same volume profile as the previous volume.

IFG0552D

4e Using information from the JFCB, this routine constructs a HDR1 label and writes it. [ETO18900]

IFG0552F

Using information from the JFCB, DCB, and TIOT, this routine constructs an HDR2 label and writes it. [ETO20400]

5 If user labels are requested and the DCB exit list contains an active output user-label routine entry, this routine gets a buffer in which the user will construct user header labels, copies the DCB from protected storage to the user's storage, and transfers control to the user's exit routine. When control is returned, this routine responds to the code passed in register 15, either writing the label or discontinuing user label processing. [ETO21700]

6 Writes tape mark after the labels (after the data on nonlabeled tape). [ETO21800]

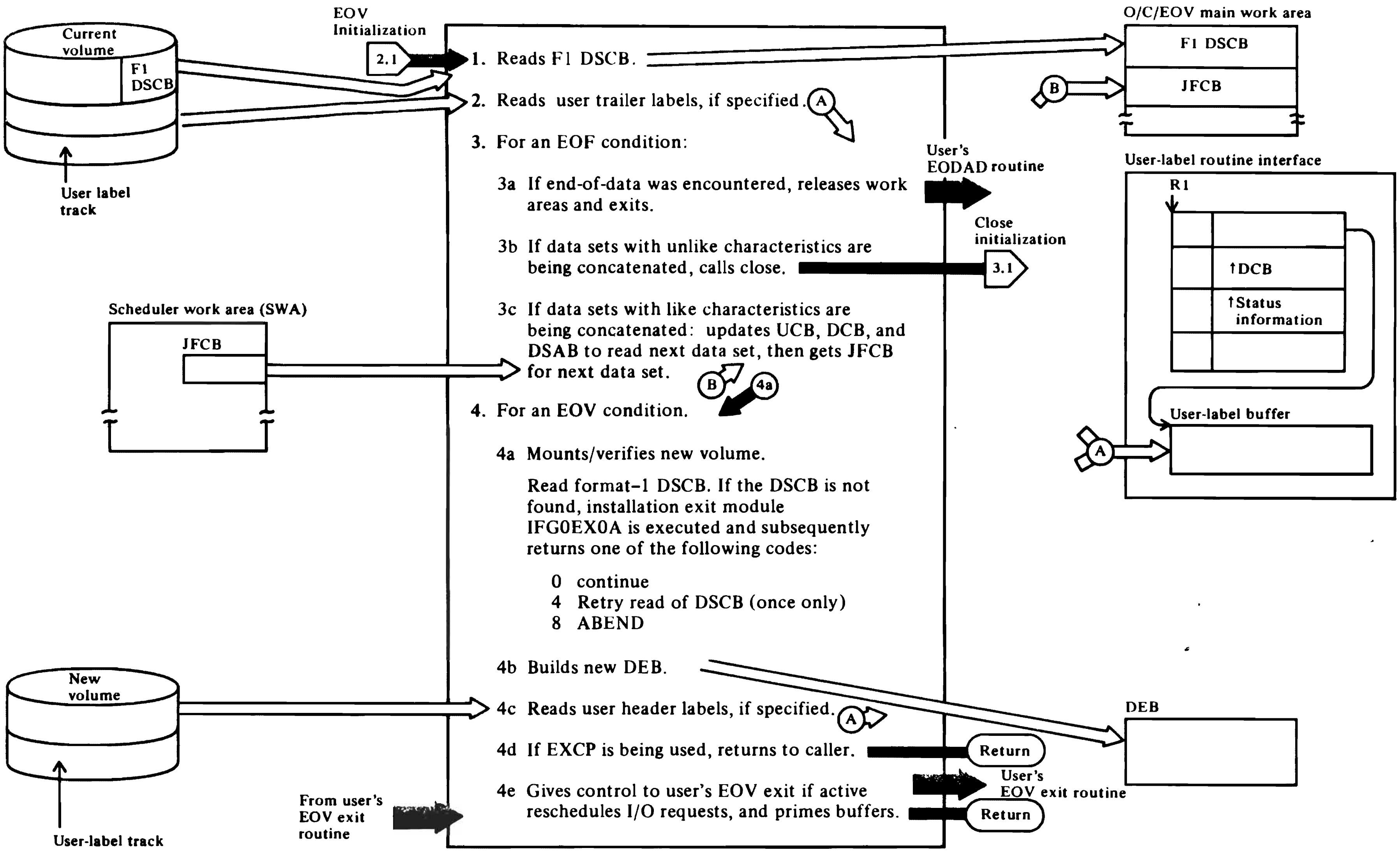
7 If the EXCP access technique is being used, this routine releases the work areas obtained in previous processing and returns to the routine that issued the EOV or FEOV macro. [ETO26650]

IFG0551L

8 This routine is entered when the sequential access methods (BSAM or QSAM) are being used. If CHKPT=EOV is specified, control is passed to IFG0558X, which takes a checkpoint and returns. If the user's DCB exit list contains an active EOV exit routine address, the EOV exit routine is given control. [EIN21400]

Reschedules any I/O requests that were purged during previous EOV processing; primes buffers, if required. Returns control to the issuer of the EOV macro, a QSAM synchronizing routine or a BSAM check routine. [EIN02290]

Diagram 2.6. EOV Direct-Access Input



1. Reads F1 DSCB.
2. Reads user trailer labels, if specified. (A)
3. For an EOF condition:
   - 3a If end-of-data was encountered, releases work areas and exits.
   - 3b If data sets with unlike characteristics are being concatenated, calls close.
   - 3c If data sets with like characteristics are being concatenated: updates UCB, DCB, and DSAB to read next data set, then gets JFCB for next data set. (B) (4a)
4. For an EOV condition.
   - 4a Mounts/verifies new volume.

     Read format-1 DSCB. If the DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

     0 continue
     4 Retry read of DSCB (once only)
     8 ABEND
   - 4b Builds new DEB.
   - 4c Reads user header labels, if specified. (A)
   - 4d If EXCP is being used, returns to caller.
   - 4e Gives control to user's EOV exit if active reschedules I/O requests, and primes buffers.

Diagram 2.6. EOV, Direct-Access Input

This function is entered from the EOV initialization function when an end-of-volume condition is encountered while processing an input data set on a direct-access device. The EOV condition means that (1) EOD has been reached; (2) the current data set is part of a concatenation of two or more data sets and EOD has been reached on one of the data sets; (3) a multivolume data set is being processed and EOV has been reached on one of the volumes; or (4) a mulitvolume data set is being processed and an FEOV macro was issued to get the next volume (or the next data set of a concatenation) ready to process.

IFG0553P

1 This routine reads the format-1 DSCB for the data set if (1) user trailer labels are to be processed or (2) a data set on which the records may span volumes is being read for updating or (3) if the format-1 DSCB is needed to determine whether this volume contains the last track written to the data set. This routine also determines whether SMF records should be written. [EDI00200]

For VIO data sets, this routine copies the format-1 DSCB maintained in the scheduler work area (SWA) into the O/C/EOV main work area. [EDI01250]

IFG0554L

2 This routine is entered when user labels are to be read and verified by the user for a direct access data set being opened for input. This routine gets virtual storage to read the labels into, reads in the first label, and transfers control to the user's label routine identified in the DCB exit list, if there is an active entry. When he has verified the label, the user returns a code in register 15 to indicate whether another label should be read or if label processing is complete. [IFG0554L]

Reads label, moves it into the user's label buffer, copies the DCB being used by EOV (the DCB copy) into the user's storage, and transfers control to the user's label routine. [ULSYNCH]

Examines contents of register 15 to determine whether to read another label, write an updated label, or terminate label processing. [ULT130]

IFG0553B (alias for IFG0202H)

If an SMF type-14 record is required (data set activity is being monitored by SMF), this routine builds JFCB and device independent DCB/DEB segment of the SMF record. [CCM02100]

IFG0202I

Completes DCB/DEB segment, builds UCB segment, and issues the SMFWTM macro to write the record to the SMF data set. [CCM02520]

IFG0552X

3 The following steps (3a, 3b, and 3c) deal with the end-of-file (EOF) condition. The EOF condition can result from reaching end-of-data (EOD) while processing concatenated data sets (steps 3b and 3c) or from reading end-of-data with no more to read (step 3a).

3a This routine is entered when EOV has been reached and there is no more to read. This routine releases the work areas obtained in previous processing, then transfers control to the user's EODAD routine. [ECM60500]

3b If the data sets have unlike characteristics or the current or next data set in the concatenation is a SYSIN data set, this routine passes the address of the current DCB in register 1, along with the work areas obtained during previous EOV processing to the second entry point in the initial load of the Close component (see Diagram 3.1). At the conclusion of Close processing of the current DCB, the Close final function passes control to the second entry point in the first load of the Open component (see Diagram 1.1) to reopen the DCB with the characteristics of the new data set. [ECM61900]

3c This routine is entered when concatenation of data sets with like characteristics is in process. Decrements data management count in the UCB (UCBDMCT field) to indicate that this unit will no longer be used to process the data set. [ECM61320]

If a VIO data set is being processed and the UCBDMCT field (the data management count of DCBs open to the data set) is zero, this routine issues the WICLOSE macro to indicate that the data set is no longer being used by this task. [ECM61340]

Updates the pointer to the DSAB in the O/C/EOV main work area (DXDSABAD), and in the DEB extension (DEBXDSAB). Updates the TIOT entry pointer in the DCB (DCBTIOT). [ECM61450]

IFG0551F

Copies the JFCB for the new data set in the concatenation to the O/C/EOV main work area. [EIN11250]

IFG0551H

Moves the volume sequence number in the JFCB (JFCBVLSQ) to the O/C/EOV main work area (DXVOLSEQ) so that the new volume can be verified or mounted and verified.

IFG0554A

4 This step is entered under two conditions: (1) from step 1 (IFG0553P) when an end-of-volume condition has been encountered and the next volume is to be mounted/verified and (2) from step 3c (IFG0551H) to get the next volume of a data set concatenation mounted/verified.

4a This routine uses the volume sequence in the DXVOLSEQ field of the O/C/EOV work area in conjunction with the volume serial numbers in the JFCB and JFCB extension to determine the volume serial number of the next volume to be processed (the *new* volume), referred to in the source code as volume *n*. If another volume is to be processed after the *new* volume, this routine determines its serial number also so that a look-ahead mount message can be issued to the operator to mount this volume. The volume serial number of volume *n-1* is also saved if that volume is to be demounted. [VUS03300]

Finds a unit allocated to this task that is available for use with this volume (note that the volume may already be mounted). [VUS05120]

IFG0194J

If the volume is not mounted, this routine writes a message to the operator requesting that he mount the volume. If the request is to mount a MSS volume, the WTO message is bypassed. The ICBMNTDE TYPE=MNT macro is issued to mount the volume without operator intervention. [WTO02000]

IFG0554C (alias for IFG0194C)

Reads the volume label (cylinder 0, track 0, record 3) into the O/C/EOV main work area. [DAMV0100]

If the volume mounted is not the one requested and if the volume is a MSS volume and not a scratch request, it is considered a hardware error and results in a 413 ABEND for OPEN, or a 737 ABEND for EOV. [DAMV0120]

Moves address of the VTOC from the volume label to the UCB. [DAMV0120]

Verifies that the correct volume is mounted. [DAMV0200]

If the MSS virtual volume is on a JES3-managed unit, then an SSOB parameter is built and the IEFSSREQ macro is issued to set up the volume for JES3 control.

Reads the format-4 DSCB for the volume, then the format-1 DSCB for the data set into the O/C/EOV main work area. [DAMV1050]

If the format-1 DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

0 continue
4 Retry read of DSCB (once only)
8 ABEND

For RACF: This routine checks the user's authorization to access a RACF-protected data set. For an output PS data set at EOV, this routine checks to ensure that the data set on the subsequent volume is consistent with the RACF definition for the data set on the previous volume.

This routine checks for data sets that are password protected, and when such a data set is detected, calls the Open/EOV password function to verify the password. [DAMV2504]

This routine gets virtual storage, chains it from the O/C/EOV main work area, and reads in format-3 DSCBs. [DAMV3000]

IFG0553X

4b When the new volume has been mounted and verified, this routine deletes the DEB containing the extents from the old volume and builds a new DEB containing the extents from the format-1 and format-3 DSCB from the new volume. If a data set spanning volumes is being updated, the old DEB is saved and its address is saved in DCBIODAD. Updates the DEBVOLSEQ field using the contents of DXVOLSEQ in the O/C/EOV main work area. [IFG0553X]

For physical sequential data sets which reside on MSS volumes, the ICBACREL TYPE=ACQ macro is issued to acquire space on virtual DASD for the extents of the data set. [EDI10720]

IFG0554L

4c This routine is entered when user header labels are to be read and verified by the user for a direct access data set being opened for input. This routine gets virtual storage to read the labels into, reads in the first label, and transfers control to the user's label routine identified in the DCB exit list, if there is an active entry. When he has verified the label, the user returns a code in register 15 to indicate whether another label should be read or if label processing is complete. [IFG0554L]

Reads label, moves it into the user's label buffer, copies the DCB being used by Open (the DCB copy) into the user's storage, transfers control to the user's label routine, and verifies the DEB when the user returns control. [ULSYNCH]

Examines contents of register 15 to determine whether to read another label, write an updated label, or terminate label processing. [ULT130]

IFG0553Z

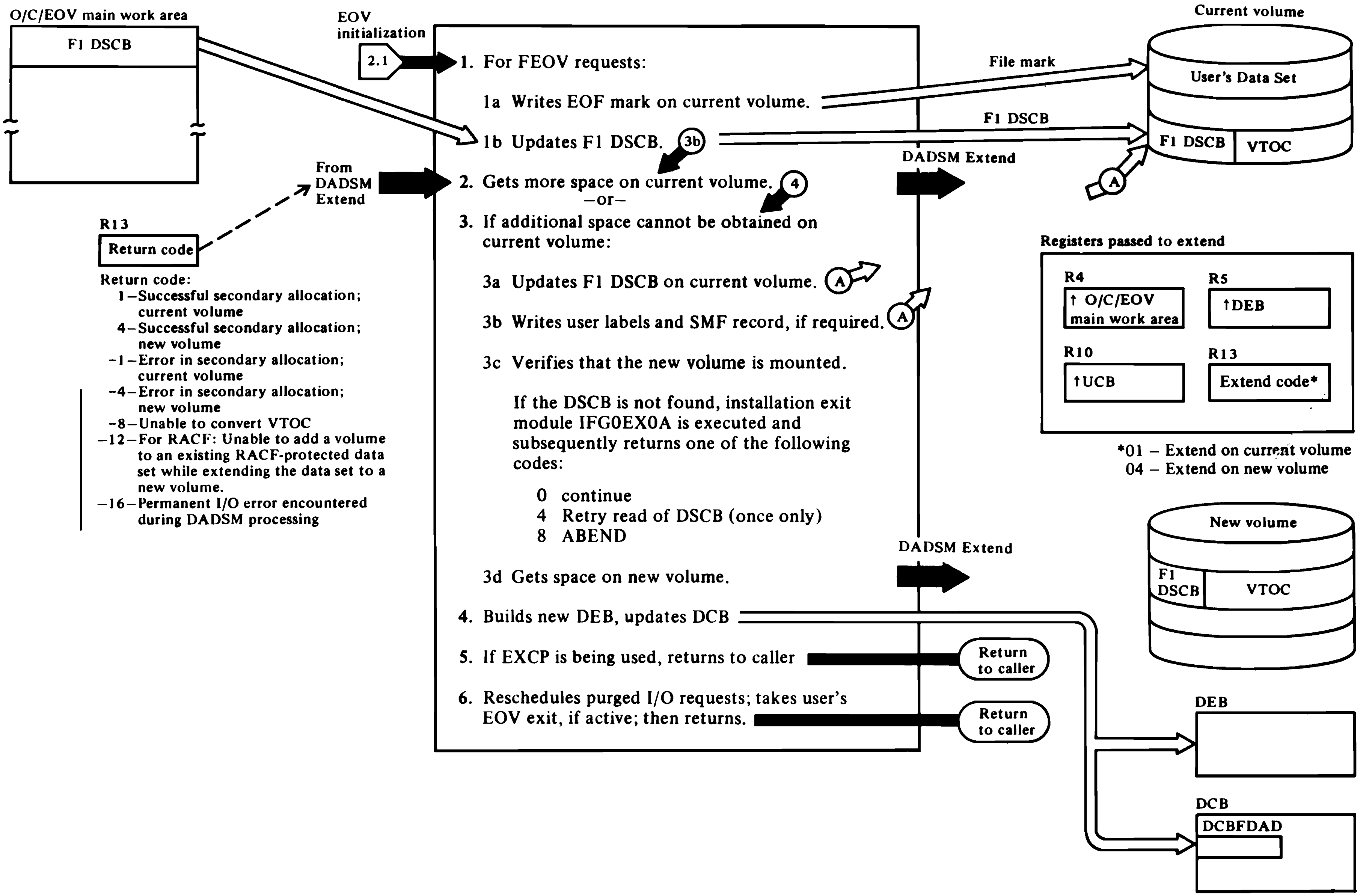
If members of partitioned (BPAM) data sets are being concatenated using BSAM or QSAM, this routine issues a BLDL macro and puts the absolute address (CCHHR) of the member in the DCBFDAD field. [EDI11500]

4d If the EXCP access technique is being used, this routine frees the work areas obtained in previous processing copies the DCB in protected storage to the user's storage and returns to the caller of EOV. [EDI11725]

IFG0551L

4e This routine receives control when an access method (BSAM or QSAM) is being used. If CHKPT=EOV is specified, control is passed to IFG0558X, which takes a checkpoint and then returns. This routine passes control to the user's EOV exit routine if there is an active address in the DCB exit list, resets all indicators showing the EOV condition in the DCB, schedules any pending I/O requests, and passes control to the the routine that issued the SVC 55 (EOV macro), a QSAM synchronizing routine or a BSAM check routine.

Diagram 2.7. EOV Direct-Access Output



1. For FEOV requests:

   1a Writes EOF mark on current volume.

   1b Updates F1 DSCB.

2. Gets more space on current volume.

   —or—

3. If additional space cannot be obtained on current volume:

   3a Updates F1 DSCB on current volume.

   3b Writes user labels and SMF record, if required.

   3c Verifies that the new volume is mounted.

   If the DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

   0 continue
   4 Retry read of DSCB (once only)
   8 ABEND

   3d Gets space on new volume.

4. Builds new DEB, updates DCB

5. If EXCP is being used, returns to caller

6. Reschedules purged I/O requests; takes user's EOV exit, if active; then returns.

Return code:

1	Successful secondary allocation; current volume
4	Successful secondary allocation; new volume
-1	Error in secondary allocation; current volume
-4	Error in secondary allocation; new volume
-8	Unable to convert VTOC
-12	For RACF: Unable to add a volume to an existing RACF-protected data set while extending the data set to a new volume.
-16	Permanent I/O error encountered during DADSM processing

Registers passed to extend

R4	R5
↑ O/C/EOV main work area	↑DEB
R10	**R13**
↑UCB	Extend code*

*01 – Extend on current volume
04 – Extend on new volume

Diagram 2.7. End-of-Volume, Direct-Access Output

This function is entered from EOV initialization when an output data set on a direct-access device was being processed and an EOV condition was raised, either as the result of an EOV macro or an FEOV macro. The EOV condition means that the space allocation was not sufficient to hold the data set on the current volume. This function first tries to get space allocated to the data set on the current volume, and if that is not successful, gets space on a new volume. When the EOV condition results from an FEOV macro, space will be obtained on a new volume if a new volume is specified.

IFG0554N

1 Steps 1a and 1b are executed only when an FEOV macro is issued or when a program using the EXCP access technique issues an EOV macro.

1a Writes a file mark to delimit the data set on the last track of the data set, unless it won't fit, in which case, it is written on the next track. If there is no next track on the volume, the file mark is written on the first track of the next extent of the data set, unless the current extent is the last, in which case, no file mark is written. [EDO00500]

1b Reads the format-1 DSCB into the O/C/EOV main work area. [EDO01700]

For VIO data sets, copies the format-1 DSCB from the scheduler work area (SWA) to the O/C/EOV main work area. [EDO01600]

Converts the CCHHR of the last track to a TTR and puts it in the DSCLSTAR field of the format-1 DSCB. Puts the number of bytes remaining on the last track of the extent in DSCTRBAL. Turns off the last-volume indicator (DSCDSIND). [EDO01800]

Writes the updated DSCB back to the VTOC. For VIO data sets, copies the DSCB back to the SWA. [EDO01950]

IFG0554P

2 This routine is entered from EOV initialization when an EOV macro is issued. This routine calls the DADSM Extend routine to get additional space, equal to the secondary quantity (JFCBSQTY), on the current volume. When control is passed to the DADSM Extend routine, register 13 is set to 1 to indicate that space is to be acquired on the current volume. [EDO02900]

When the DADSM Extend routine return to this routine, having acquired new space on the current volume, this routine reads the updated format-1 DSCB (and the format-3 DSCB if there is one). [EDO04000]

3 If the data set disposition (JFCBIND2) indicates a disposition of OLD or MOD, or if a new volume is required because the current volume doesn't have enough space available for the secondary allocation or if the volume already contains 16 extents of the data set, this routine sets up to complete processing on the current volume (write labels, write SMF data set activity records, etc.) before getting additional space on a new volume. [EDO03515]

IFG0554Z

3a Reads the format-1 DSCB from the VTOC of the current volume, updates the DSCLSTAR and DSCTRBAL fields and resets the last-volume indicator (DSCDSIND), then writes the updated DSCB back to the VTOC. [EDO14550]

IFG0555B

3b Gets a buffer for building user labels, makes sure that a user label track has been allocated, copies the DCB from protected storage to the user's storage, and gives control to the user's label building routine. When control is returned, this routine writes the label or discontinues label processing as specified by the user via register 15. [IFG0555B]

IFG0553B (alias for IFG0202H)

If an SMF type 15 record is required (data set activity is being monitored by SMF), this routine builds the JFCB segment and device-independent segment of the SMF record. [CCM02100]

IFG0202I

Completes DCB/DEB segment, builds UCB segment, and issues the SMFWTM macro to write the record to the SMF data set. [CCM02520]

IFG0554P

3c This routine calls the volume mount/verification routines to mount and verify or merely verify that a new volume is available. [EDO03550]

IFG0554A

This routine uses the volume sequence in the DXVOLSEQ field of the O/C/EOV main work area in conjunction with the volume serial numbers in the JFCB and JFCB extension to determine the volume serial number of the next volume to be processed (the *new* volume), referred to in the source code as volume *n*. If another volume is to be processed after the *new* volume, this routine determines its serial number also so that a look-ahead mount message can be issued to the operator to mount this volume. [VUS03300]

Finds a unit allocated to this task that is available for use with this volume (note that the volume may already be mounted). [VUS05120]

IFG0554J (alias for IFG0194J)

If the volume is not mounted, this routine writes a message to the operator requesting that he mount the volume. If the request is to mount a MSS volume, the WTO message is bypassed. The ICBMNTDE TYPE=MNT macro is issued to mount this volume without operator intervention. [WTO02000]

IFG0554C (alias for IFG0194C)

Reads the volume label (cylinder 0, track 0, record 3) into the O/C/EOV main work area. [DAMV0100]

If the volume mounted is not the one requested and if the volume is a MSS volume and not a scratch request, it is considered a hardware error and results in a 413 ABEND for OPEN, or a 737 ABEND for EOV. [DAMV0112]

Moves address of the VTOC from the volume label to the UCB. [DAMV0120]

Verifies that the correct volume is mounted. [DAMV0200]

If the MSS virtual volume is on a JES3-managed unit, then an SSOB parameter list is built and the IEFSSREQ macro is issued to set up this volume for JES3 control.

If the DSCB is not found, installation exit module IFG0EX0A is executed and subsequently returns one of the following codes:

0 continue
4 Retry read of DSCB (once only)
8 ABEND

IFG0554T

This routine is entered if a new volume needs to be mounted but no demountable unit can be found. If the DCB is not BSAM or QSAM with a B37 ABEND exit active (X'08' in high order byte of the DCB exit list), a B37 ABEND is issued.

If the DEB has 16 extents (which indicates that it is not possible to attempt to extend the data set on the current volume), the format-1 DSCB is read and the DSCLSTAR and DSCTRBAL fields are updated, the last-volume bit (DSCDSIND) is set, and the DSCB is rewritten; all the BSAM/QSAM IOBs are updated to no longer indicate an end-of-volume condition. The IOBs are set to indicate an I/O error so that this B37 condition will not be detected again. If Close called EOV or if there is no SYNAD address specified, an 001 ABEND is issued. Otherwise EOV exits to the caller, which should result in a subsequent EOV call when the SYNAD exit will be taken.

If the DEB has less than 16 extents, the user's B37 ABEND exit is taken. A return code of 0, passed by the user's B37 ABEND exit routine, means that some action was taken by the user that should allow that data set to be extended on the current volume; IFG0554P is given control. If a return code of 2 is passed, the format-1 DSCB is updated and a B37 ABEND is issued. If a return code of 1 is passed, the DSCLSTAR and DSCTRBAL fields of the format-1 DSCB are updated, the last volume bit is set in DSCDSIND, and the DSCB is written to the VTOC; then the IOBs are updated to indicate that a permanent I/O error has been encountered. [IFG0554T]

IFG0554P

3d When the new volume has been mounted and verified, and space has not been allocated on the new volume, this routine passes control to the DADSM Extend routine to get additional space on the new volume. Register 13 is set to 4 to indicate to the DADSM Extend routine that the space is to be acquired on a new volume. [EDO02900]

IFG0554C (alias for IEF0194C)

If the MSS virtual volume is on a JES3-managed unit, then an SSOB parameter list is built and the IEFSSREQ macro is issued to set up this volume for JES3 control.

IFG0555H

4 This routine is entered, after the DSCB has been read on the new volume, or after additional space has been acquired, to free the old DEB and construct a new one that contains the new extents. The old DEB is removed from the TCB-DEB chain and the new DEB added. Its address is added to the job step's DEB table. [EDO22700]

Updates the DCBFDAD to the beginning of the new extent. [EDO23300]

If any of the new DEB extents exist on MSS virtual volumes, the ICBACREL TYPE=ACQ macro is issued to acquire the space on a virtual volume. [EDO24120]

5 If the EXCP access technique, this routine frees the work areas obtained in previous processing and returns to the routine that issued the EOV macro. [EDO24900]

IFG0555J

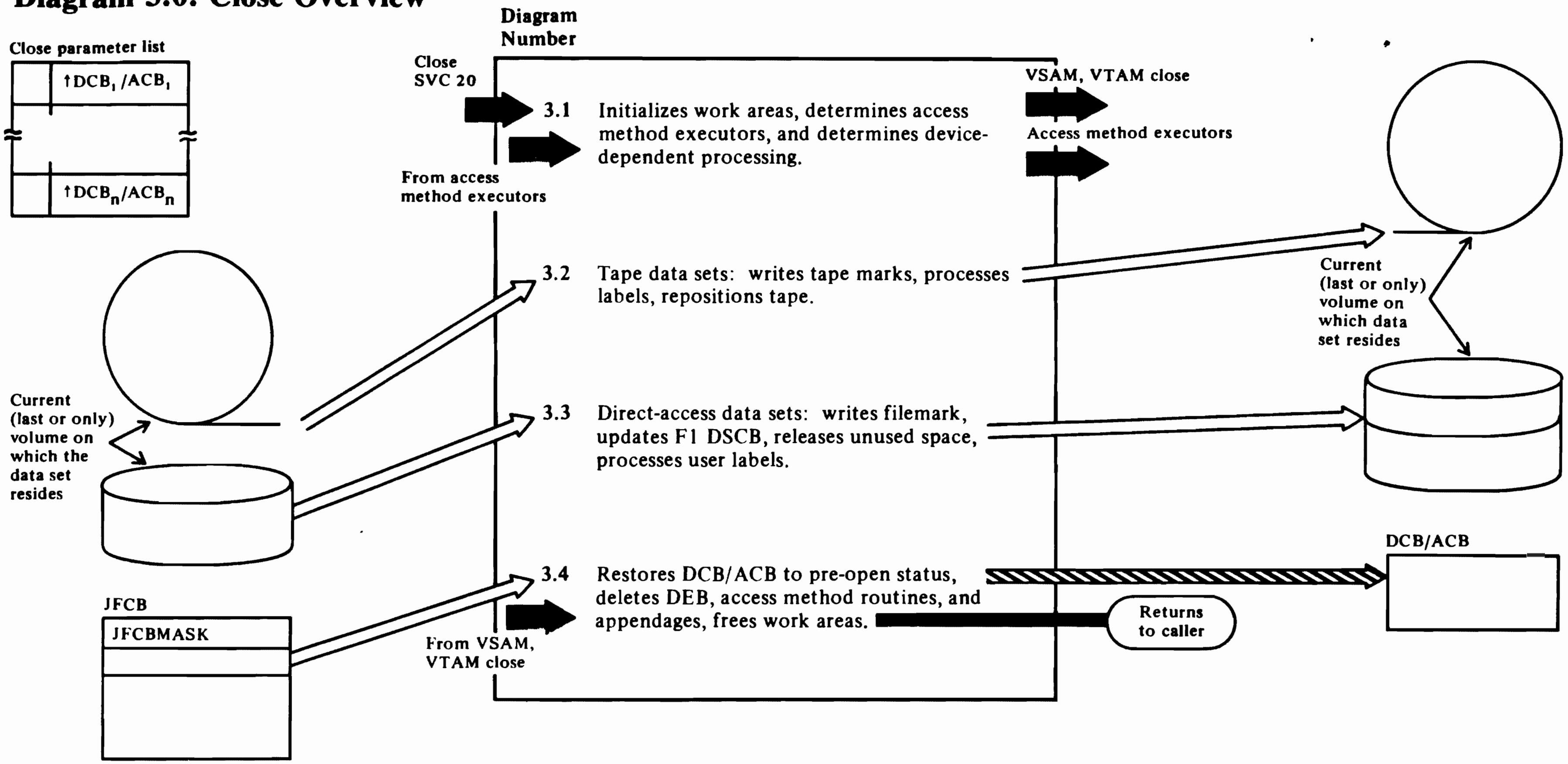
When a space has been acquired on a new volume, this routine writes user header labels (UHL) on the user-label track, if they are required. A buffer is acquired in which the user label can be constructed, the DCB in protected storage is copied to the user's storage and control is passed to the user's label building routine. When control is returned, this routine writes the UHL or terminates processing, based on the code passed in register 15. [IFG0555J]

IFG0551L

6 This routine is entered when the sequential access methods (BSAM or QSAM) are being used. If CHKPT=EOV is specified, control is passed to IFG0558X, which takes a checkpoint and then returns. If the user's DCB exit list contains an active EOV exit routine address, the EOV exit routine is given control. [EIN21400]

Reschedules any I/O requests that were purged during previous EOV processing; primes buffers, if required, then returns to the issuer of the EOV or FEOV macro. [EIN02290]

Diagram 3.0. Close Overview



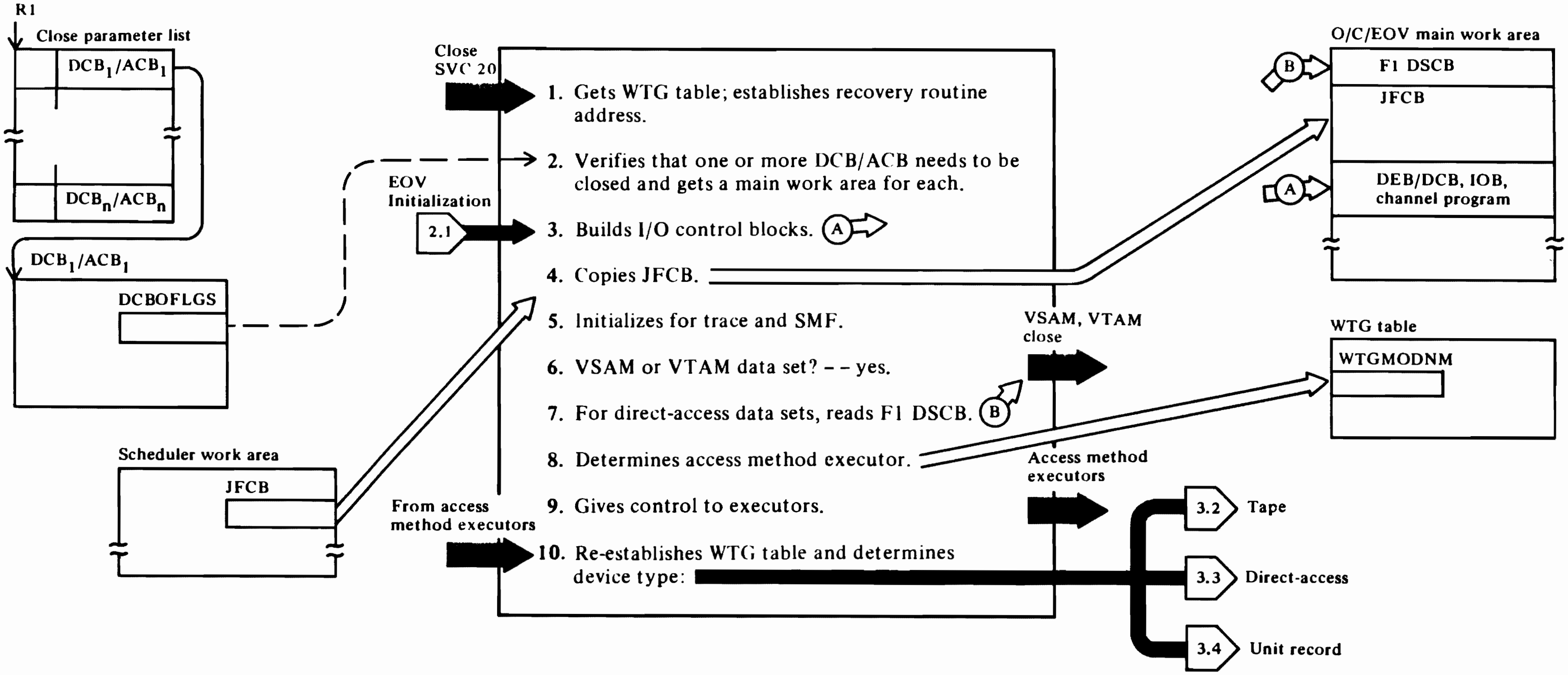
3.1 Initializes work areas, determines access method executors, and determines device-dependent processing.

3.2 Tape data sets: writes tape marks, processes labels, repositions tape.

3.3 Direct-access data sets: writes filemark, updates F1 DSCB, releases unused space, processes user labels.

3.4 Restores DCB/ACB to pre-open status, deletes DEB, access method routines, and appendages, frees work areas.

Diagram 3.1. Close Initialization



1. Gets WTG table; establishes recovery routine address.
2. Verifies that one or more DCB/ACB needs to be closed and gets a main work area for each.
3. Builds I/O control blocks.
4. Copies JFCB.
5. Initializes for trace and SMF.
6. VSAM or VTAM data set? – – yes.
7. For direct-access data sets, reads F1 DSCB.
8. Determines access method executor.
9. Gives control to executors.
10. Re-establishes WTG table and determines device type:

Diagram 3.1. Close Initialization

The Close initialization function is entered from the SVC second-level interrupt handler when SVC 20 is issued. Close initialization is also entered from EOV at Step 3, below, when EOD is reached on a data set being concatenated with another data set with different (unlike) characteristics.

IGC00020—CLOSE (SVC 20)

1 This routine gets a WTG table and associated work areas for this close request. See "Data Area Layouts" in the "Data Areas" section for an example of the manner in which work areas are obtained and chained during close processing. [IGC00020]

2 Checks the open, busy, and lock bits in the DCB/ACB OFLGS field to see if the DCB/ACB can be closed (lock and open bits on, busy bit off). An O/C/EOV main work area is needed for each ACB/DCB that can be closed. [CIN0010A]

Gets virtual storage for an O/C/EOV main work area for each ACB/DCB in the Close parameter list that passes the previous test. The address of each O/C/EOV work area is stored in the WTG table, the work area is initialized, and the DCB/ACB is copied into protected storage (the IECRES INIT macro is used to copy the DCB/ACB and, at the same time, turn on the busy bit in DCBOFLGS). [CIN00105]

If the open, busy, and lock bits are on and the access method is not VSAM or BTAM, this routine verifies that a DEB on the TCB-DEB chain points to the right DCB or ACB. [CIN00610]

3 This routine is entered from EOV (1) when volumes with unlike attributes are being concatenated for processing; and (2) when the user has specified in his DCB ABEND exit routine that he wants to ignore an error associated with EOV processing of a DCB.

This routine is entered to restore the DCB to closed status, before reopening the DCB to reflect the unlike attributes of the next data set to be processed in the concatenation. [IGC01020]

Builds a DEB/DCB in the O/C/EOV main work area. These control blocks will be used in subsequent I/O operations (reading DSCBs and labels, writing tapemarks, etc.). [CIN00860]

For data sets being processed using the EXCP access technique (MACRF=E) and for DCBs associated with a task being abnormally terminated , this routine purges any outstanding requests for input or output; the I/O halt option of the PURGE macro is specified (see *OS/VS2 System Programming Library: Data Management* for a description of the PURGE macro). [CIN00900]

IFG0200V

4 Copies the JFCB for the data set from the scheduler work area (SWA) to the O/C/EOV main work area. For catalog requests, control is passed to IFG0200N to move the JFCB (see *OS/VS2 Catalog Management Logic*). [CIN02300]

5 This routine checks to see if the system management facility (SMF) is being used to write type-14 and -15 records to the SMF data set; if so, sets an indicator that causes the appropriate record to be written whenever necessary in subsequent processing. [CIN02550]

6 This routine checks for an ACB representing a VTAM data set and when one is encountered, transfers control to the VTAM close function, whose entry point is IFG0200U (see *OS/VS2 VTAM Logic*). For ACBs representing VSAM data sets, control is transferred to the VSAM close function (entry point is IFG0200T), and for DCBs associated with ISAM programs that are to process VSAM data sets, control is passed to IFG0200S, the VSAM-ISAM compatibility interface. [CIN02600]

7 For direct-access data sets with standard user labels, this routine reads the format-1 DSCB into the O/C/EOV main work area. [CIN03600]

For VIO data sets, instead of reading the format-1 DSCB from a VTOC, this routine copies the format-1 DSCB maintained in the scheduler work area (SWA) to the O/C/EOV main work area. [CIN03900]

IFG0200W

8 The access method executor routine that is to receive control is determined. The access method executor to receive control for data sets on unit record and tape devices is determined based on (1) the data set organization, whether it is a TSO task closing the data set, or whether the task is being abnormally terminated; and (2) the type of device the data set resides on. An access method executor ID is entered in the WTG table for each entry in the Close parameter list. module names for these executors are shown as exits in the module flowchart for IFG0200W (see Chart 3.1). No executors are required to close dummy data sets or data sets being processed using the EXCP access technique. [CIN04400]

For data sets on direct access devices, this routine determines the access method executor that should receive control. This is done for each entry in the Close parameter list. The executor to receive control depends on data set organization (TCAM, SAM, ISAM, BDAM, etc.) [CIN05100]

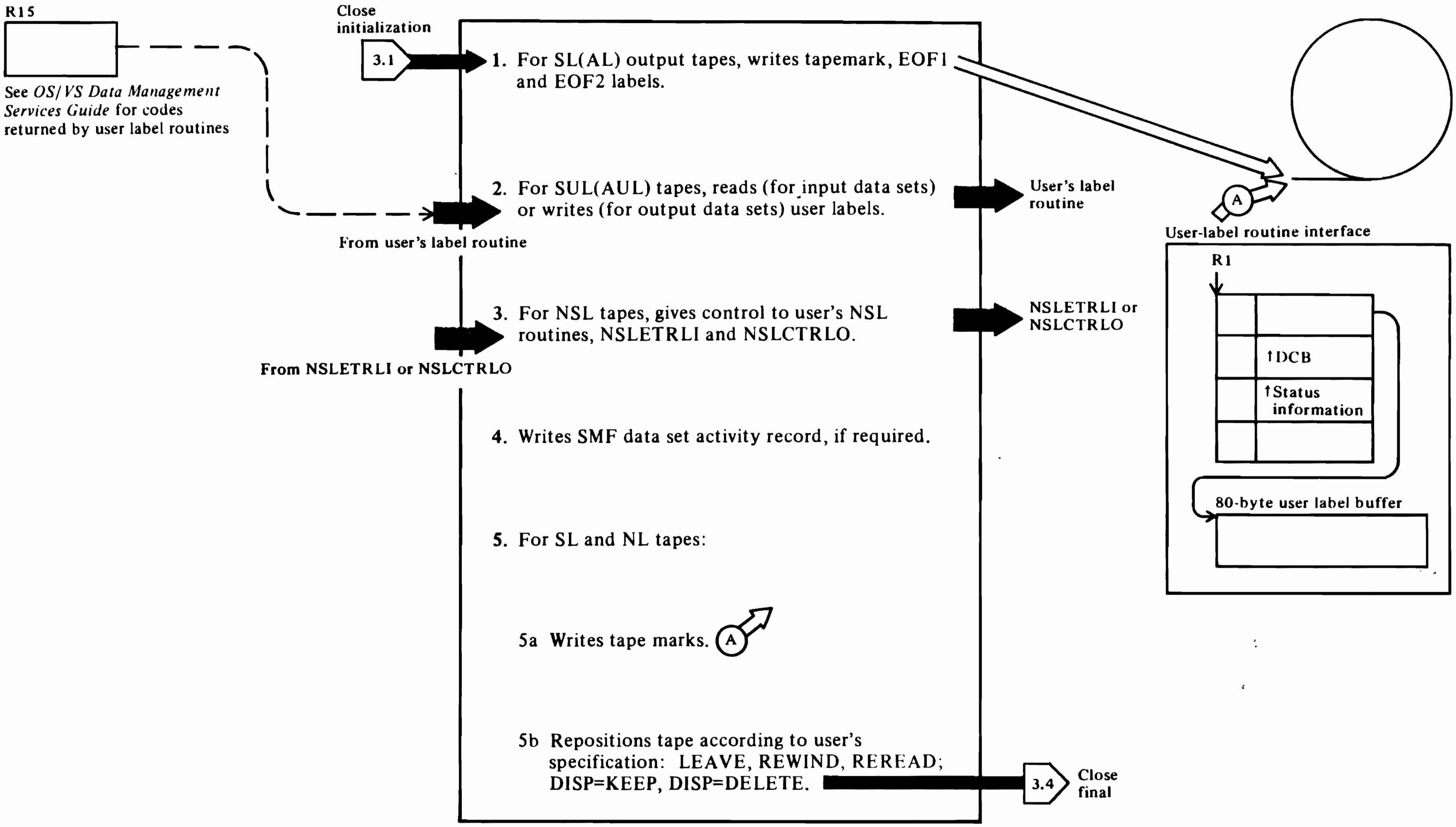
For SYSIN and SYSOUT data sets, this routine determines whether the request is represented by an ACB or a DCB. DCBs are processed by passing control to a routine that translates the DCB request to an ACB request and issues a second CLOSE macro against the ACB it constructed during Open processing, causing the Close routines to be reentered to close the ACB (see *OS/VS2 SAM Logic* for a description of the SYSIN-SYSOUT DCB executor, IGG0201W). [CIN05410]

9 When an entry in the WTG table has been specified for each member of the Close parameter list, this routine gives control to the first access method executor identified in the first WTG table entry. A complete list of the Close access method executors that can receive control is shown in Chart 3.1. [CIN05600]

IFG0200Y (alias IGG0200F and IGG0200G)

10 When control is returned from the access method close executors, this routine determines which Close function is to receive control, based primarily on the type of device on which the data set resides. Other considerations include whether the data set was opened for input or output and the type of labels used for direct-access and tape data sets. When the next Close function has been determined, a module ID is again put in the WTG table. This is done for each entry in the parameter list, which concludes the Close initialization function. The next function to receive control can be Close Tape (Diagram 2.2), Close DA (Diagram 2.3), or Close Final (Diagram 2.4). [CIN06100]

Diagram 3.2. Close Tape



1. For SL(AL) output tapes, writes tapemark, EOF1 and EOF2 labels.

2. For SUL(AUL) tapes, reads (for input data sets) or writes (for output data sets) user labels.

3. For NSL tapes, gives control to user's NSL routines, NSLETRLI and NSLCTRLO.

4. Writes SMF data set activity record, if required.

5. For SL and NL tapes:

   5a Writes tape marks.

   5b Repositions tape according to user's specification: LEAVE, REWIND, REREAD; DISP=KEEP, DISP=DELETE.

R15
See *OS/VS Data Management Services Guide* for codes returned by user label routines
Close initialization
3.1
From user's label routine
From NSLETRLI or NSLCTRLO
User's label routine
NSLETRLI or NSLCTRLO
A
User-label routine interface
R1
↑DCB
↑Status information
80-byte user label buffer
A
3.4
Close final

Diagram 3.2. Close Tape

This function is entered from the Close initialization function when a data set on tape is being closed. All label types are processed in this function, with exits to the user's label routine exits.

IFG0200Z

1 This routine writes a tapemark and EOF1 and EOF2 labels for output data sets. Information for the labels is gathered from the JFCB, TIOT, DCB, and UCB. If AL is specified, the labels are translated to ASCII before they are written. Additional information on standard and user labels can be found in *OS/VS Tape Labels*. [CTA00200]

IFG0202A

2 For both input (label processing deferred from EOV to EOF) and output data sets on tape, this routine processes user labels (SUL and AUL) if (1) the JFCBLTYP field indicates their presence and (2) the user's DCB exit list contains an active entry for user labels. If an output data set is being closed, a cleared buffer is passed to the user's program, in which he can construct a user label. For deferred label processing, the user label is read into the buffer before it is passed to the user's program. [CTA03600]

Before passing control to the user's label routine, this routine copies the DCB from protected storage into the user's storage. [CTA06100]

When control is returned from the user's label processing routine, the DCBOFLGS field is updated from the user's storage to protected storage. Then the contents of register 15 are examined to determine whether the user wants to continue to process labels. [CTA03800]

IFG0202B

3 This routine is entered when the JFCB for a data set on tape indicates the presence of nonstandard labels (NSL).

If NSL processing was deferred from end-of-data until the data set is closed this routine transfers control to NSLETRLI, the user installation's EOV trailer label routine, so that he can process the labels and position to another data set on the tape or dispose of the volume. The DCB in protected storage is copied to the user's storage before control is passed. [CCM00500]

This routine transfers control to NSLCTRLO so the user can write nonstandard trailer labels on output data sets, and position the tape or dispose of the volume. **Note:** no positioning is done by the system control program for NSL tapes. [CCM00700]

For NSL data sets, error statistics by volume (ESV) are written. [CCM01200]

IFG0202H

4 If a System Management Facility (SMF) type 14 or 15 record is required (data set activity is being monitored by SMF), this routine:

Builds JFCB and device independent DCB/DEB segment of the SMF record. [CCM02100]

Completes DCB/DEB segment and builds UCB segment. [CCM05050]

Issues SMFWTM macro instruction to write the record to the SMF data set. [CCM04700]

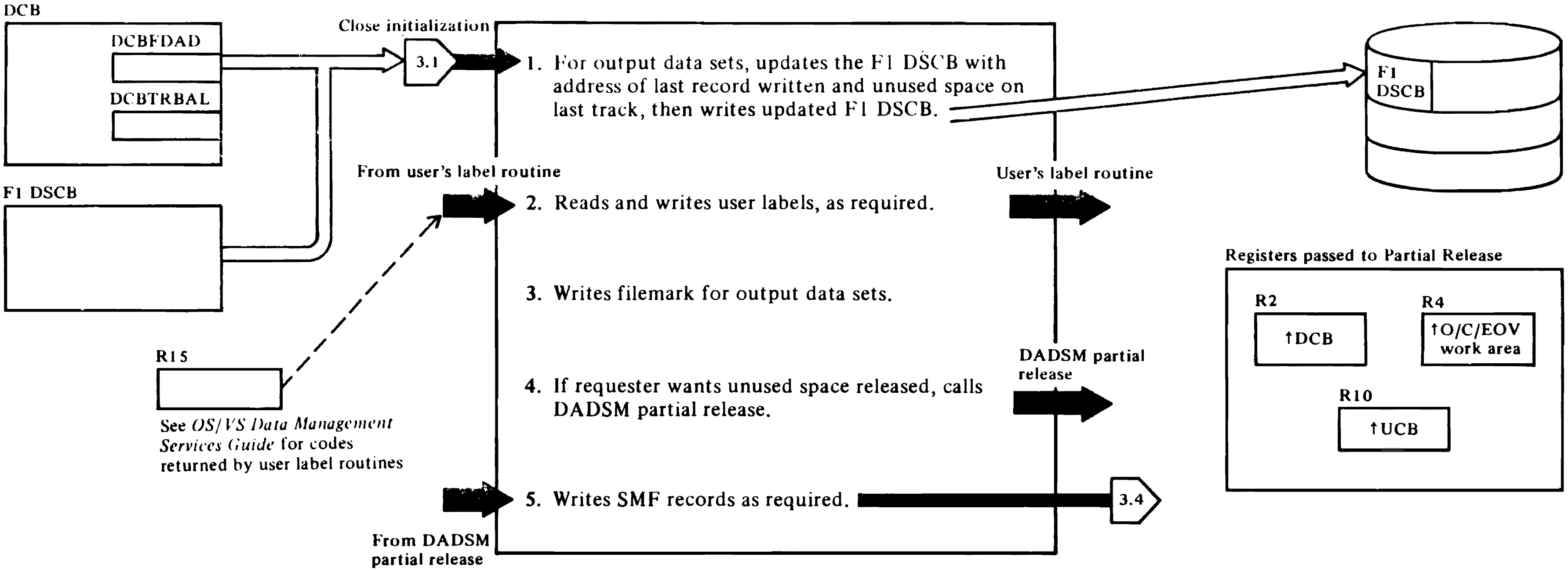
IFG0202F

5a This routine positions tape volumes according to the close option specified by the user and updates the UCB accordingly. For unlabeled tape (NL specified in the DD statement), tape marks are written after the data. For standard labeled tapes, tapemarks are written after the trailer labels. If REREAD is specified and the data set is the first one on the volume, the tape is rewound. If the data set is not the first one on the volume, four backspace files (BSF) and one forward space file (FSF) are issued to position to the beginning of a standard labeled tape; if the tape is not labeled, only two BSFs are required. [CTA06850]

5b If none of the Close options (LEAVE, REREAD, REWIND, or DISP) are specified, DISP is assumed, and the TIOESTTA field (byte 2 of the TIOT DD entry for the data set) is examined to determine the disposition action required. [CTA08900]

If KEEP is indicated for a reserved data set or if REWIND is indicated, the tape is rewound and the rewind bit (bit 3 in the UCBWGT field) is set to indicate to the scheduler component that no rewind is necessary. If the tape is to be unloaded, a demount message will be issued. [CTA09100]

Diagram 3.3. Close Direct-Access



1. For output data sets, updates the F1 DSCB with address of last record written and unused space on last track, then writes updated F1 DSCB.
2. Reads and writes user labels, as required.
3. Writes filemark for output data sets.
4. If requester wants unused space released, calls DADSM partial release.
5. Writes SMF records as required.

See *OS/VS Data Management Services Guide* for codes returned by user label routines

Diagram 3.3. Close Direct Access

This function is entered from the Close initialization function when a data set on a direct-access device is being closed. Steps 1, 3, and 4 are performed only for output data sets, while steps 2 and 5 are always performed, whether an input or output data set is being closed.

IFG0200Y (alias IGG0200F and IGG0200G)

1 This routine is entered to close output data sets on a direct-access device. If the format-1 DSCB has not yet been modified during Close processing, it is read in again. The actual device address in the DCBFDAD field is converted to TTR format and put in the format-1 DSCB. The track balance field (DSCTRBAL), the last-track-written field (DSCLSTAR), and the last-volume indicator are updated. For partitioned data sets, the directory count (DSCBCDBL) is also updated. This routine then determines the next module ID and puts the ID in the WTG table entry. [CIN07700]

IFG0201R

This routine writes the updated format-1 DSCB to the VTOC; if the user has requested that the unused portion of an output data set be released, the updated format-1 DSCB will be written by the DADSM Partial Release routine (see step 4, below). [CDA00215]

For VIO data sets, this routine copies the format-1 DSCB from the O/C/EOV main work area back to the scheduler work area (SWA). [CDA00215]

Only the first volume of multivolume data sets contain user labels. Therefore, if the current volume of the data set is not the first volume, this routine must find the first volume and read in the format-1 DSCB from that volume. If the first volume is not mounted, no user labels will be written. [CDA01300]

IFG0202C

2 This routine is entered if user labels are required for a data set opened for input or update. Trailer labels are read and passed to the user's label processing routine which is pointed to by the exit list in the DCB. If updating is necessary, this routine writes the user's updated labels when control is returned from the user's label routine. [CDA02500]

IFG0202D

This routine is entered if user labels are required for data sets opened for output. Control is passed to the user's label routine to format the labels. When control is returned, the labels are written by this routine on the user-label track associated with the data set [CDA05200]

IFG0202E

3 This routine is entered to write an end-of-file mark to mark the end of a physical-sequential, output data set. If no space remains in the last extent allocated to the data set, no file mark is written. [CDA06600]

4 If unused space is to be released from an output data set before the data set is closed, this routine transfers control to the DADSM Partial Release routine, IGG020P1 (see *OS/VS2 DADSM Logic*). When control is returned from the DADSM Partial Release routine, a check is made for errors encountered during release processing. [CDA07840]

IFG0202H

5 If an SMF type-14 or -15 record is required (data set activity is being monitored by SMF), this routine builds JFCB and device independent DCB/DEB segment of the SMF record. [CCM02100]

IFG0202I

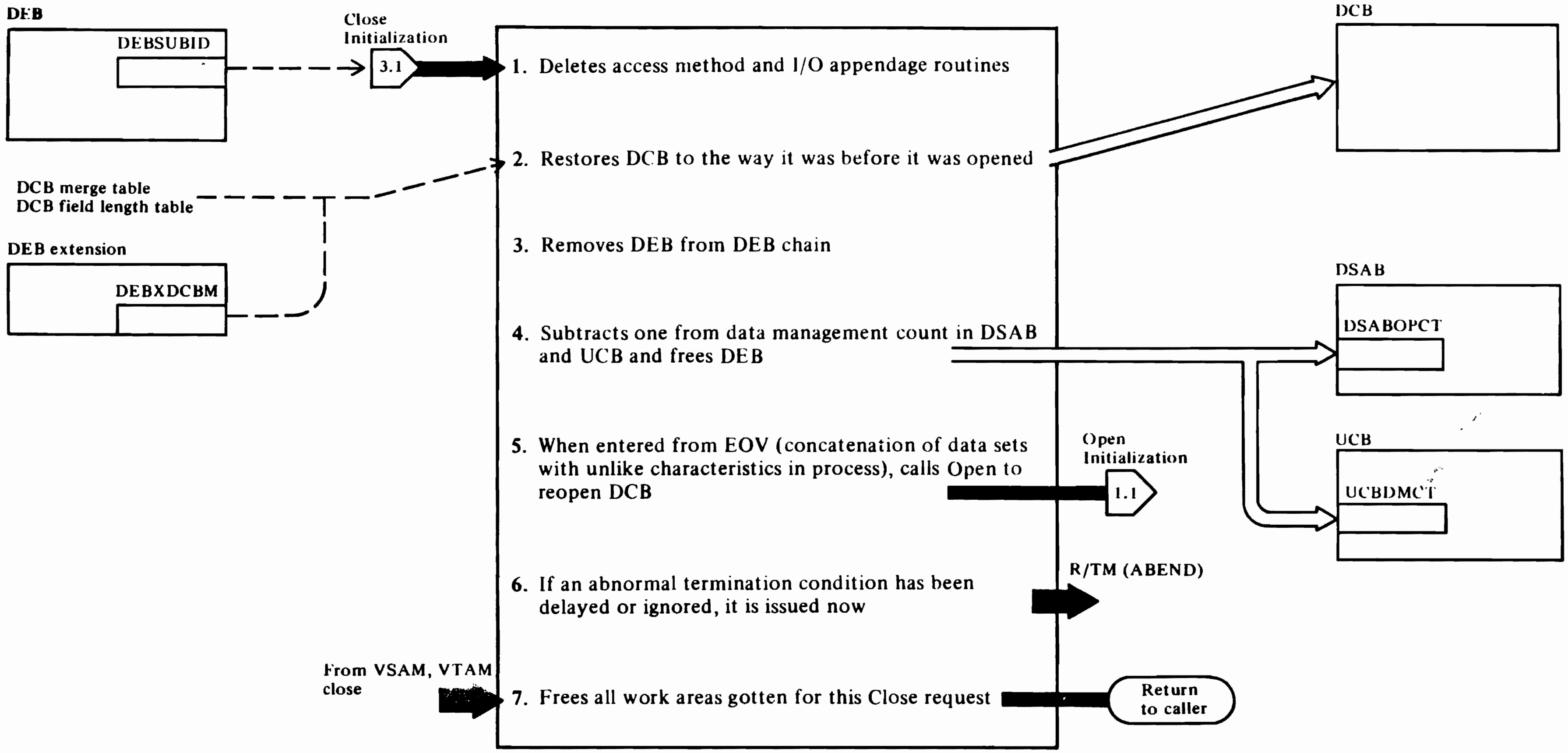
Completes DCB/DEB segment. [CCM02520]

Builds UCB segment. [CCM02770]

Builds ISAM extension. [CCM03650]

Issues the SMFWTM macro to write the record to the SMF data set. [CCM05300]

Diagram 3.4. Close Final



1. Deletes access method and I/O appendage routines
2. Restores DCB to the way it was before it was opened
3. Removes DEB from DEB chain
4. Subtracts one from data management count in DSAB and UCB and frees DEB
5. When entered from EOV (concatenation of data sets with unlike characteristics in process), calls Open to reopen DCB
6. If an abnormal termination condition has been delayed or ignored, it is issued now
7. Frees all work areas gotten for this Close request

Diagram 3.4. Close Final

This function is entered at the conclusion of the device dependent processing when data sets on tape and direct-access volumes are being closed. It is also entered from the Close initialization function when SYSIN, SYSOUT, and dummy data sets are being closed. The VTAM and VSAM Close components also return to this function at step 7.

IFG0202J

Frees the error recovery procedure (ERP) work areas and user totalling save area. [CFN00500]

Using the list of access method processing routines and their appendages that appears in the DEBSUBID field of the DEB, this routine deletes the routines to indicate that one or more of the tasks associated with each DCB are no longer using these routines. [CFN01000]

2 This routine restores some of the fields of the DCB to the setting they had before the JFCB was merged into the DCB during Open processing. The OPTCD, BFTEK, BFALN (HIARC bits 0 and 5 are not restored), DEVT and DVTBL fields for direct-access DCBs are restored. For TCAM DCBs with DSORG=TX or TQ, the BUFMAX, PCI, RESERVE, BUFSIZE, and THRESH fields are restored to 0, if they had been merged from the JFCB during Open processing. [CFN01200]

IFG0202K

3 Removes the DEB for this data set from the TCB-DEB chain for the task.

This routine searches the DSAB(s) associated with the data set and decrements the count of the number of DCBs or ACBs currently open to the data set. [CFN06172]

4 The data management count (UCBDMCT field) is the count of DCBs open to a data set on a particular unit. This count is decremented by one in each UCB that represents a unit with a volume of the data set mounted. [CFN01300]

Turns off the busy bit and open bit (unless errors have occurred while processing an ACB) in the DCBOFLGS filed. [CFN06610]

IFG0202L

5 Checks for concatenation of data sets with unlike attributes; this condition indicates that EOV has requested this close operation. This routine transfers control to Open to reopen the DCB with the appropriate attributes. An O/C/EOV main work area is passed to Open. [CFN07900]

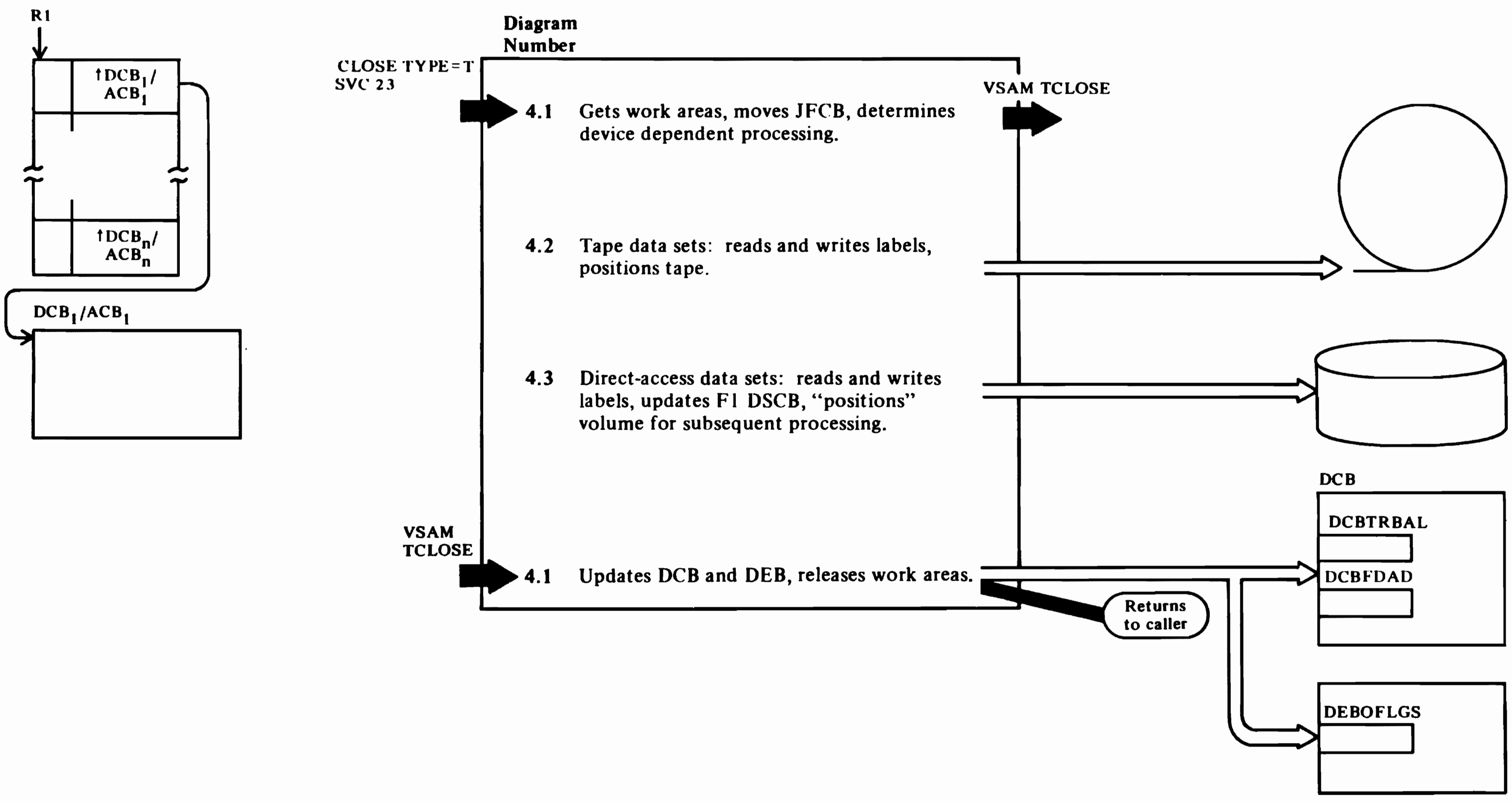
6 If the user has specified in his DCB ABEND exit routine that an abnormal termination condition that arose during Close processing be delayed or ignored, the ABEND is issued now. [CFN08080]

Sets the return code in register 15 for return to the routine that issued the CLOSE macro. [CFN07600]

If the user specified that his data set be freed at the time it is closed (rather than at task termination), this routine calls the dynamic allocation component so that the data set, volume, and unit can be deallocated. They will be deallocated if the data set, volume, or unit is not otherwise in use. [CFN08015]

7 Issues the IECRES FREE macro, which frees the work areas obtained to process the request. Returns control to the caller. [CFN08100]

Diagram 4.0. TCLOSE Overview



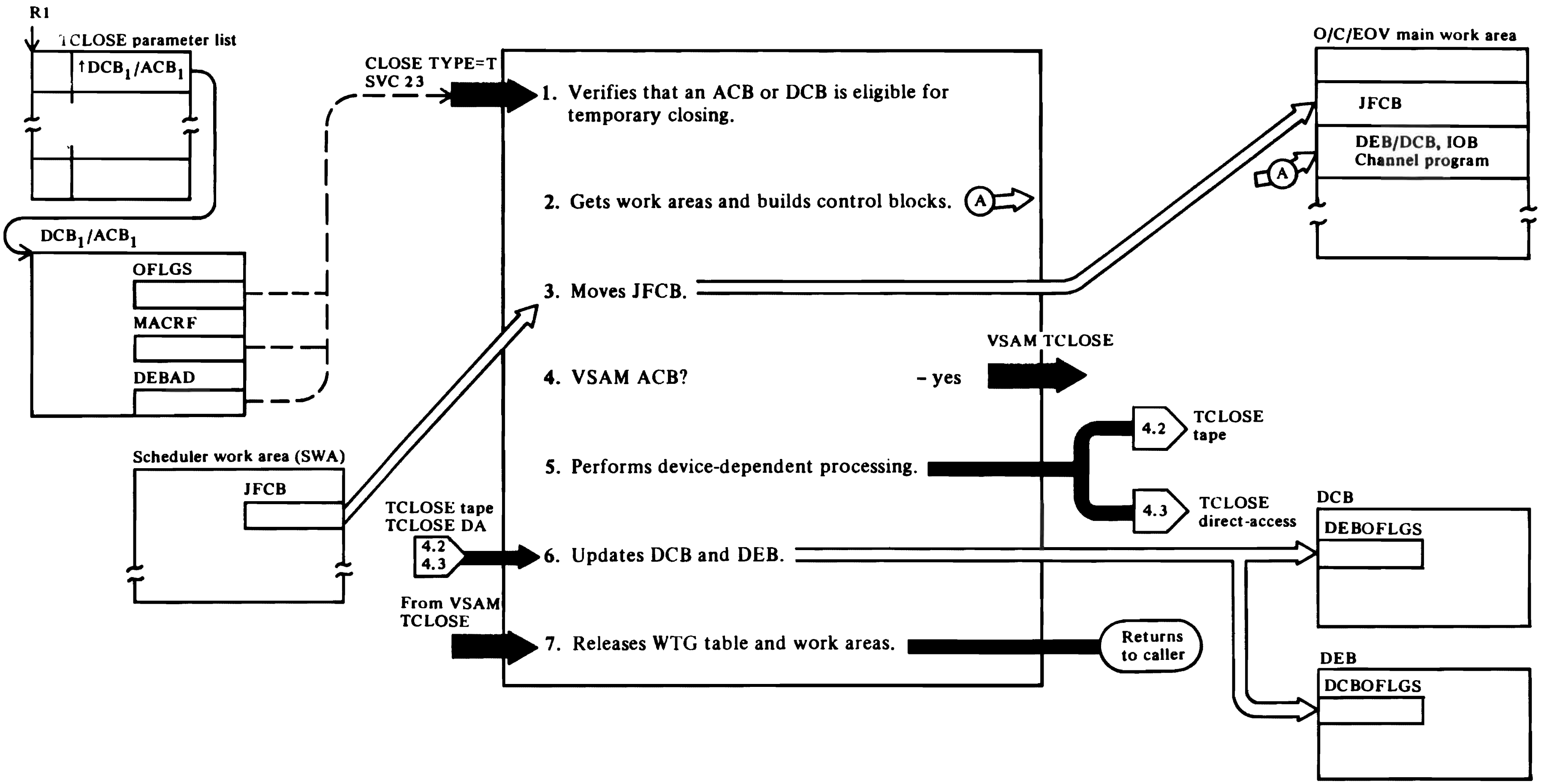
4.1 Gets work areas, moves JFCB, determines device dependent processing.

4.2 Tape data sets: reads and writes labels, positions tape.

4.3 Direct-access data sets: reads and writes labels, updates F1 DSCB, "positions" volume for subsequent processing.

4.1 Updates DCB and DEB, releases work areas.

Diagram 4.1. TCLOSE Initialization and Final Processing



1. Verifies that an ACB or DCB is eligible for temporary closing.

2. Gets work areas and builds control blocks.

3. Moves JFCB.

4. VSAM ACB? - yes

5. Performs device-dependent processing.

6. Updates DCB and DEB.

7. Releases WTG table and work areas.

Diagram 4.1. TCLOSE Initialization

This function is entered from the SVC second-level interrupt handler when SVC 23 is issued. SVC 23 results from the expansion of a CLOSE macro instruction when ,,TYPE=T is coded. TYPE=T indicates that a data set being processed using BSAM or VSAM is to retain its open status while its labels are being processed, and it is to be repositioned for the next processing phase. Temporary close processing is not supported for SYSIN or SYSOUT data sets or for any access method except BSAM or VSAM. This function checks for ACBs representing VSAM data sets and transfers control to the VSAM TCLOSE component to process these ACBs (see *OS/VS2 Virtual Storage Access Method (VSAM) Logic*). The VSAM TCLOSE component returns to the TCLOSE final function (IFG0232Z), to ensure parallel processing of the DCBs included in a multiple TCLOSE that consists of both VSAM ACBs and BSAM DCBs.

IGC0002C—CLOSE, TYPE=T (SVC 23)

1 This routine, the first load of TCLOSE, gets virtual storage for a WTG table and associated work areas. An entry of the WTG table is used for each DCB and ACB in the parameter list. See "Data Area Layouts" for a description of the work areas obtained and how the work areas are chained. [IGC0002C]

For each parameter list entry, this routine checks the lock and open bits of the OFLGS field of the DCB or ACB. If both bits are set to 1, it is assumed that the DCB/ACB is to be temporarily closed and processing is continued. If no DCB/ACB in the parameter list has both bits on, control is immediately returned to the caller. [TCM00500]

2 When all the tests in step 1 are successful, this routine gets an O/C/EOV main work area for each valid parameter list entry and copies the user's DCB into protected storage. [TCM00575]

If a checkpoint data set is being temporarily closed, frees the copy of the user's DCB and copies the protected checkpoint DCB into protected storage. [TCB00580]

Initializes the work area by building a DEB, DCB, IOB, and channel program in it. These control blocks are needed to read DSCBs, process labels, and to position data sets. [TCM06500]

Issues a PURGE macro instruction specifying the quiesce I/O option to purge outstanding I/O requests. (See *OS/VS2 System Programming Library: Data Management* for a description of the use of the PURGE macro.) The purge is not necessary for VSAM data sets. [TCM07000]

3 This routine moves the JFCB associated with each DCB from the scheduler work area (SWA) to the O/C/EOV main work area for that DCB. It is not necessary to move the JFCB for VSAM ACBs. [TCM07600]

4 If the data set being temporarily closed is a VSAM data set, this step sets the WTGMODNM entry for this ACB to IFG0231T to transfer control to VSAM TCLOSE (see *OS/VS2 Virtual Storage Access Method (VSAM) Logic*). [TCM09500]

5 Checks the UCBTBYT3 field to determine which device-dependent TCLOSE function to transfer control to, TCLOSE tape or TCLOSE direct-access. [TCM09570]

IFG0232Z

6 This routine is entered at conclusion of device-dependent and VSAM TCLOSE processing. It is verified that each DCB/ACB represented in the parameter list has been processed by each of the preceding TCLOSE functions and is now ready for final processing. [TCM50000]

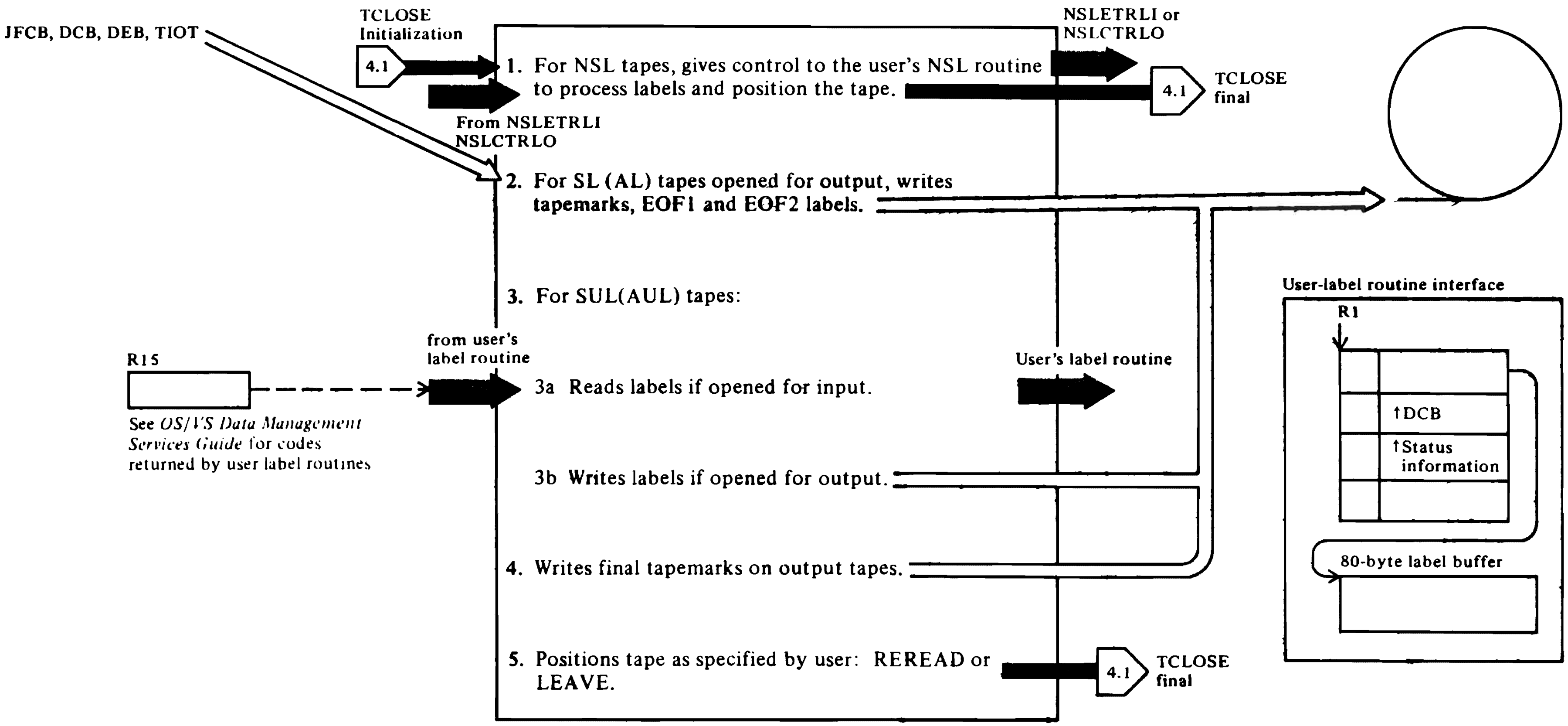
Determines for which parameter list entries abnormal termination has been deferred by the user's DCB ABEND exit routine, if any. [TCM52500]

If processing was terminated unsuccessfully on a DCB or ACB, only the busy bit (OBUSY field) is reset in the O/C/EOV main work area. If TCLOSE processing of an ACB/DCB was successful, the OWRIT, ORDBK and OEOF bits are reset to zero in the main work area. [TCM53600]

Copies the DCB in protected storage into the user's storage. [TCM53800]

7 When all DCBs represented in the TCLOSE parameter list have been processed, this routine frees the WTG table and all work areas associated with the request and returns to the caller. [TCM54500]

Diagram 4.2. TCLOSE Tape



1. For NSL tapes, gives control to the user's NSL routine to process labels and position the tape.

2. For SL (AL) tapes opened for output, writes tapemarks, EOF1 and EOF2 labels.

3. For SUL(AUL) tapes:

3a Reads labels if opened for input.

3b Writes labels if opened for output.

4. Writes final tapemarks on output tapes.

5. Positions tape as specified by user: REREAD or LEAVE.

See *OS/VS Data Management Services Guide* for codes returned by user label routines

Diagram 4.2. TCLOSE Tape

This function is entered from the TCLOSE intialization routine when a data set on tape is being temporarily closed. The processing performed in this function depends on the type of label on the tape and on whether the data set was opened for input or output.

IFG0202B

1 Before transferring control to one of the user's NSL routines, this routine copies the DCB in protected storage to the user's storage. [CCM00250]

This routine transfers control to the user's NSL Close routine, NSLCTRLO, so that the user can process NSL output labels and position the tape to continue processing. [CCM00700]

When NSL processing was deferred at end-of-volume, to be performed at end-of-data, this routine passes control to the user's EOV NSL routine, NSLETRLI, so that the user can process the labels and position the volume. [CCM00500]

Note: No positioning is done for NSL tapes by the system control program. The user's NSL routine is responsible for volume positioning during Close and TCLOSE processing. When control is returned to TCLOSE, only the TCLOSE final function is performed.

IFG0232G

2 This routine processes standard-labeled tapes for output data sets. One tapemark is written to delimit the data.

This routine can also be reentered after FEOV processing (see next paragraph). [TCT00400]

When end-of-tape was sensed when a tapemark was written, this routine makes a test to determine whether LEAVE was specified as the positioning option; if so, the tape is backspaced past the tape mark and an FEOV macro is issued so that trailer labels will be written, a new volume mounted, and HDR1 and HDR2 labels written. When control is returned to this routine from EOV, the user's DCB is again copied into protected storage. [TCT00600]

Builds and writes an EOF1 label following the tapemark (data delimiter) using information from the DCB, DEB, UCB, and JFCB. [TCT03000]

Builds an EOF2 label using information from the JFCB and UCB. [TCT03200]

IFG0232M

Completes EOF2 label using information from the TIOT, JFCB, DSAB, and UCB and writes it after the EOF1 label. [TCT06200]

3 This routine is entered when a tape with standard user labels (SUL or AUL) is encountered. Obtains storage for a user-label work area. [TCT07200]

If the data set was opened for output, this function passes the user's program a buffer in which the user builds his label. The user returns control to this routine to write the label. Register 15 is used by the user's label routine to specify whether label processing is to continue and what the next action is to be. [TCT08200]

If the data set was opened for input, this routine reads the label into the user's label buffer and passes control to the user's label routine to verify the label. [TCT09200]

Before passing control to the user's label routine, the DCB in protected storage is copied into the user's storage. [TCT10800]

IFG0232S

This routine is entered after user-label processing of AUL and SUL tapes and after TCLOSE initialization for NL, SL, and AL tapes. [TCT14200]

4 For output data sets, the first of two tapemarks is written following either the data (NL tape) or the data set labels (SL, AL, SUL, or AUL tape). [TCT14400]

Writes the second tapemark. [TCT14800]

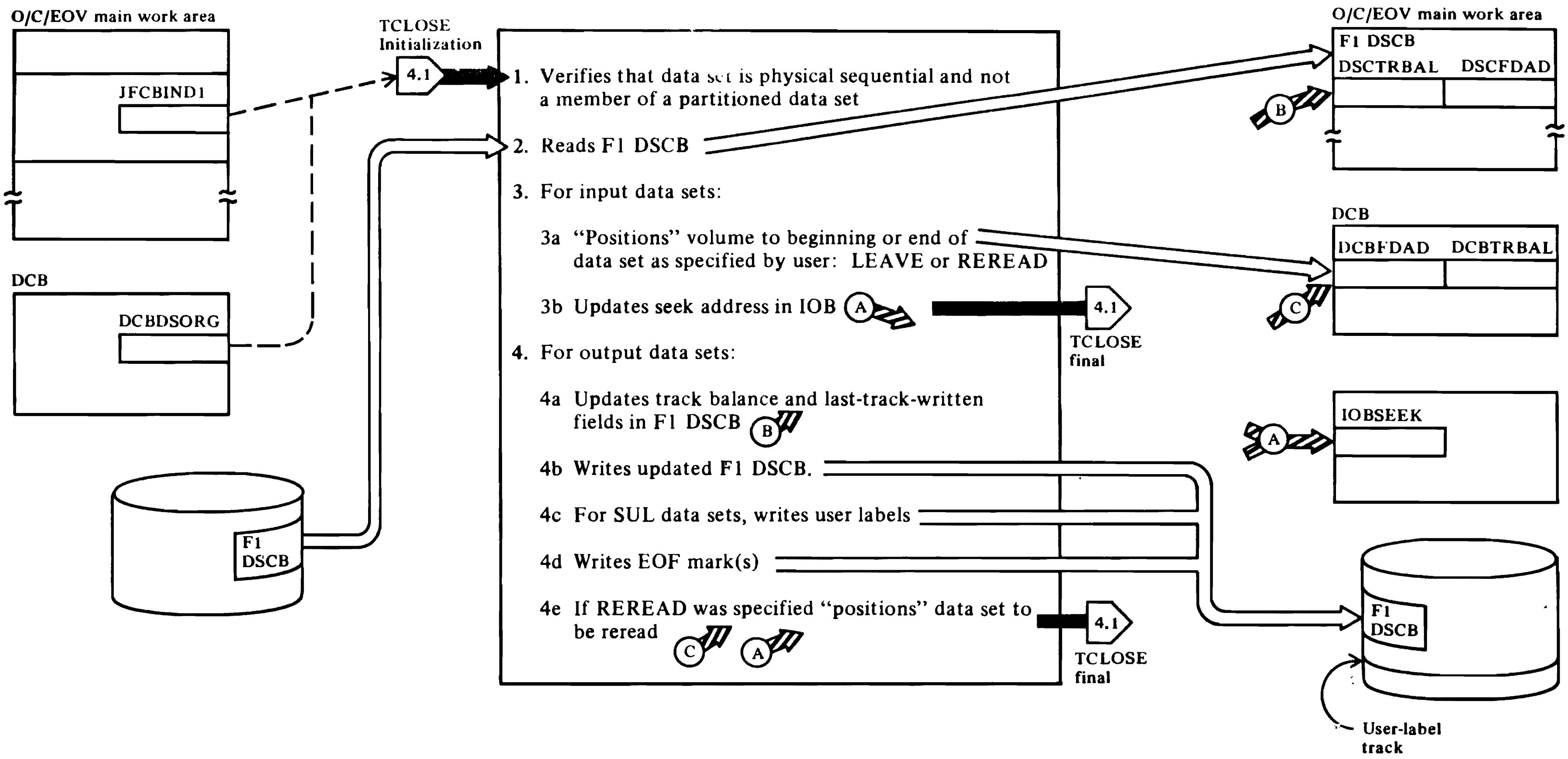
For NL tape data sets, a test is made to determine whether end-of-tape was sensed and, if so, whether LEAVE was specified; if both conditions were met, the tape is backspaced past the tapemark and an FEOV macro instruction is issued to pass control to EOV to mount a new volume. When the new volume has been mounted, normal TCLOSE processing is resumed. [TCT14600]

Writes the second tapemark. [TCT14800]

5 This routine positions labeled (SL or AL) tape according to (1) how the tape was being read (forward or backward) and (2) the positioning specified in the CLOSE ,,,TYPE=T macro (LEAVE or REREAD). [TCT16000]

This routine positions unlabeled (NL) tape using the same criteria as labeled tape. [TCT16800]

Diagram 4.3. TCLOSE Direct-Access



1. Verifies that data set is physical sequential and not a member of a partitioned data set
2. Reads F1 DSCB
3. For input data sets:
   - 3a "Positions" volume to beginning or end of data set as specified by user: LEAVE or REREAD
   - 3b Updates seek address in IOB (A)
4. For output data sets:
   - 4a Updates track balance and last-track-written fields in F1 DSCB (B)
   - 4b Writes updated F1 DSCB.
   - 4c For SUL data sets, writes user labels
   - 4d Writes EOF mark(s)
   - 4e If REREAD was specified "positions" data set to be reread (C) (A)

Diagram 4.3. TCLOSE Direct Access

This function is entered from TCLOSE initialization when a data set on a direct-access device is to be temporarily closed. The processing performed by this function depends on whether an input or an output data set is being temporarily closed. Steps 1 and 2 below are entered for both input and output data sets. Note that user labels are not processed for input data sets during TCLOSE processing.

TCLOSE DA processing support is intended for sequential data sets, both physical sequential and partitioned. However, data sets with other organizations are also processed sequentially, namely direct and indexed-sequential data sets. Direct and indexed-sequential data sets are not delimited with file marks, which are needed by TCLOSE to position to the end of data; therefore, TCLOSE can only reposition direct and indexed-sequential data sets that specify repositioning to the beginning of the data set. For partitioned data sets, TCLOSE only repositions to the beginning (the directory) or the end (following the last member) of the data set. No repositioning to individual members is done.

To ensure proper processing, the TCLOSE DA routines check the DCB, JFCB, and format-1 DSCB to determine data set organization.

IFG0232D

1 This routine is entered from TCLOSE initialization for both input and output data sets. The DSORG field of the DCB is tested for physical sequential (PS) and the JFCBIND1 field is tested to make sure that the DCB is not open to a member of a partitioned data set. If either test fails, TCLOSE processing of this DCB is terminated. [TCD00100]

2 For VIO data sets, this routine copies the format-1 DSCB maintained in the scheduler work area (SWA) to the O/C/EOV main work area. [TCD00110]

This routine reads the format-1 DSCB into the O/C/EOV main work area. [TCD00120]

3 If LEAVE positioning is to be performed for an input data set (indicating that the user wants to be at the end of the data set), the data set organization in the format-1 DSCB is tested for PS or PO. If it is neither, processing is terminated for this DCB. [TCD00150]

3a If the disposition of an input data set is LEAVE, the volume is positioned to the end of the data set by updating (1) the DCBFDAD field to point to the end-of-file mark following the data and (2) the track balance field (DCBTRBAL) to reflect the number of bytes remaining on the track. [TCD00150]

If the disposition of an input data set is REREAD (indicating that the user wants to be positioned to the beginning of the data set when he resumes processing), the volume is positioned to the beginning of the data set by updating (1) the DCBFDAD field to point to the beginning of the first extent of the data set and (2) the DCBTRBAL field to reflect the capacity of a full track. [TCD00200]

3b The full disk address in the IOBSEEK field is updated to correspond to the updated DCBFDAD, i.e., to point to the beginning of the first extent of the data set. [TCD00300]

4 When a data set opened for output is being temporarily closed, the DSORG field in the format-1 DSCB is tested for PS; if the DSORG is not PS, TCLOSE processing for the DCB is terminated. [TCD00800]

4a This routine updates the track balance field (DSCTRBAL), the last-track-written field (DSCLSTAR) of the format-1 DSCB for the data set, and sets the last-volume indicator (DSCDSIND). [TCD00900]

4b When the format-1 DSCB has been updated, this routine writes it back to the VTOC. If a VIO data set is being processed, this routine will move the updated format-1 DSCB back to the scheduler work area (SWA). Format-1 DSCBs for VIO data sets are maintained in the SWA, rather than in the VTOC of the volume on which they reside. [TCD00930]

IFG0232J

4c This routine processes user labels: builds the SUL interface, transfers contol to the user's label routine, and when control is returned, writes the user label to the user-label extent. [IFG0232J]

Before transferring control to the user's label routine, the DCB in the O/C/EOV main work area is copied into the user's storage. [TCD04100]

IFG0232Z

4d Writes an end-of-file mark to delimit the data set after the last record written for data sets using RECFM=FS. Writes end-of-file mark on the next track for all data sets. [TCD01700]

4e If a disposition of REREAD was specified, the data set is positioned to be reread. This is done by setting the DCBFDAD and IOBSEEK fields to the first track of the first extent of the data set. [TCD02500]

Diagram 5.0. Common Routines—Overview

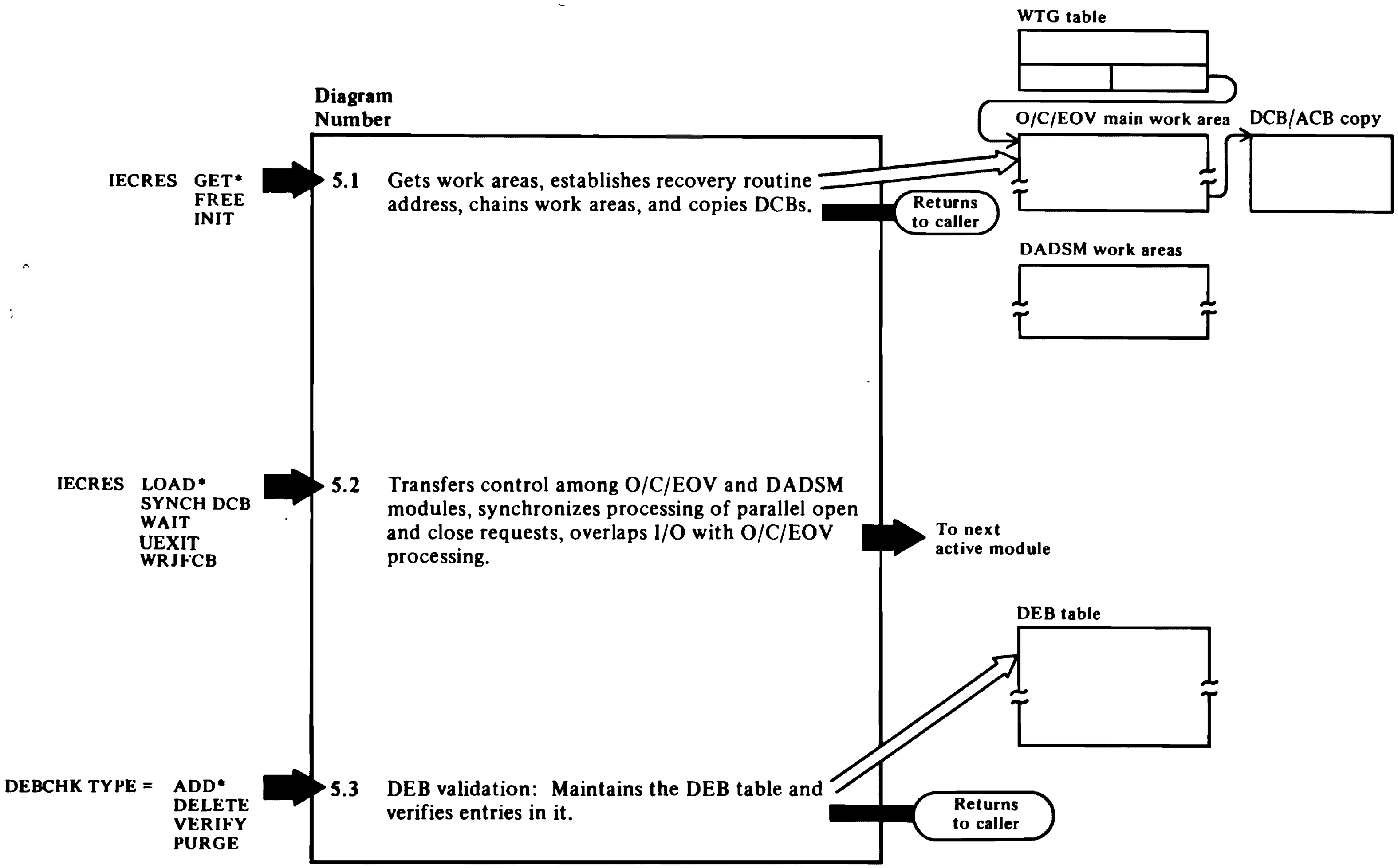


Diagram Number

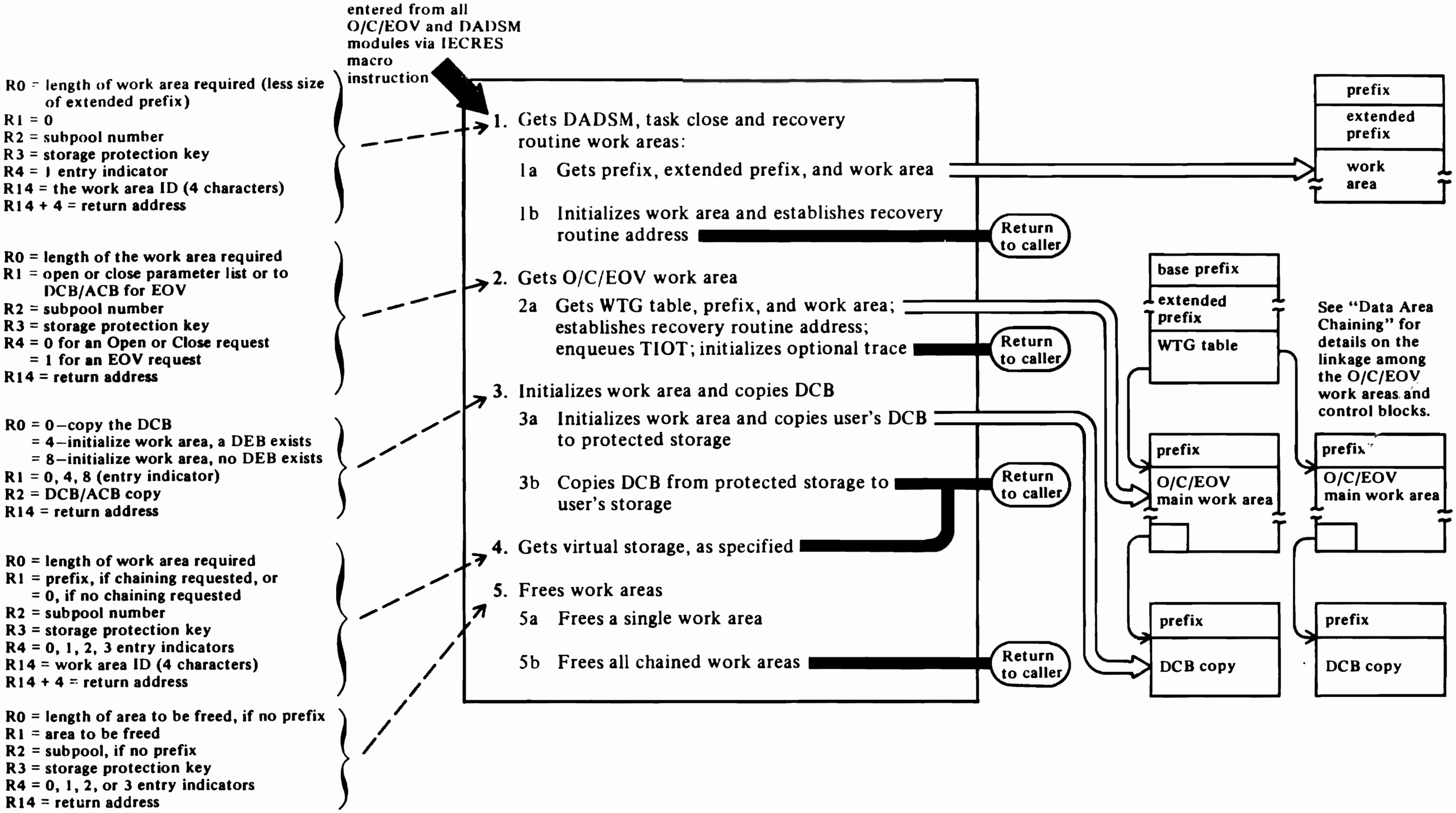
5.1 Gets work areas, establishes recovery routine address, chains work areas, and copies DCBs.

5.2 Transfers control among O/C/EOV and DADSM modules, synchronizes processing of parallel open and close requests, overlaps I/O with O/C/EOV processing.

5.3 DEB validation: Maintains the DEB table and verifies entries in it.

*See Diagrams 5.1, 5.2, and 5.3 for input (register contents) that results from the expansion of these macro instructions.

Diagram 5.1. Get and Initialize Work Areas, Free Work Areas



entered from all O/C/EOV and DADSM modules via IECRES macro instruction

R0 = length of work area required (less size of extended prefix)
R1 = 0
R2 = subpool number
R3 = storage protection key
R4 = 1 entry indicator
R14 = the work area ID (4 characters)
R14 + 4 = return address

R0 = length of the work area required
R1 = open or close parameter list or to DCB/ACB for EOV
R2 = subpool number
R3 = storage protection key
R4 = 0 for an Open or Close request
= 1 for an EOV request
R14 = return address

R0 = 0–copy the DCB
= 4–initialize work area, a DEB exists
= 8–initialize work area, no DEB exists
R1 = 0, 4, 8 (entry indicator)
R2 = DCB/ACB copy
R14 = return address

R0 = length of work area required
R1 = prefix, if chaining requested, or
= 0, if no chaining requested
R2 = subpool number
R3 = storage protection key
R4 = 0, 1, 2, 3 entry indicators
R14 = work area ID (4 characters)
R14 + 4 = return address

R0 = length of area to be freed, if no prefix
R1 = area to be freed
R2 = subpool, if no prefix
R3 = storage protection key
R4 = 0, 1, 2, or 3 entry indicators
R14 = return address

1. Gets DADSM, task close and recovery routine work areas:
   - 1a Gets prefix, extended prefix, and work area
   - 1b Initializes work area and establishes recovery routine address
2. Gets O/C/EOV work area
   - 2a Gets WTG table, prefix, and work area; establishes recovery routine address; enqueues TIOT; initializes optional trace
3. Initializes work area and copies DCB
   - 3a Initializes work area and copies user's DCB to protected storage
   - 3b Copies DCB from protected storage to user's storage
4. Gets virtual storage, as specified
5. Frees work areas
   - 5a Frees a single work area
   - 5b Frees all chained work areas

See "Data Area Chaining" for details on the linkage among the O/C/EOV work areas and control blocks.

Diagram 5.1. Get and Initialize Work Areas, Free Work Areas

These subroutines of the O/C/EOV-DADSM service routine are entered via the IECRES macro issued by the Open, Close, End-of-volume, and DADSM functions; the subroutines get, initialize, and free work areas, work area prefixes, the WTG table, and buffers; they also enqueue and dequeue the TIOT of data sets that may be released before the task they are allocated to is terminated; and they establish the address of a recovery routine.

IFG019RA

1 IECRES GET,PREFIX=FIRST: The DADSM and Task Close functions obtain a work area (that includes a modified WTG table for use by the DADSM routines), a prefix, and an extended prefix using this macro. The macro causes two entries to IFG019RA, the first to get the storage, the second to initialize it and to establish the address of the Task Close or O/C/EOV-DADSM recovery routine that is to get control if an error occurs during processing (also called an *ESTAE routine*).

1a First entry: calculates the amount of space based on the length of a prefix, extended prefix, and the recovery routine parameter list, as well as the amount of storage requested. [RES01750]

Calls the GETMAIN routines via branch entry and sets the area obtained to zeros. Builds the prefix. [RES01830]

1b Second entry: initializes the write-to-operator (WTO) length field used by the module trace routine and saves the address of the recovery routine parameter list in the work area just obtained. [RES01760]

Loads the Task Close recovery routine or the first load of the O/C/EOV-DADSM recovery routines and issues the ESTAE macro to establish the address that is to get control if the Task Close or DADSM routine fails. Diagrams 6.2 through 6.4 describe the O/C/EOV-DADSM recovery routines. [RES06000]

2 IECRES GET,PREFIX=WTG | EOV: this routine is called by the O/C/EOV routines. For requests from EOV, it gets only one main work area and a single-entry WTG table, which is part of the main work area, while for requests from Open and Close, a variable-length WTG table and more than one work area can be obtained. Storage for the other data areas associated with the request is also obtained: prefix for every work area, extended prefix for each initial work area, the recovery routine parameter list, an area for copying the O/C/EOV parameter lists into protected storage, a work area used by IFG019RA, and a trace buffer, if needed. These storage areas are illustrated in "Data Area Layouts."

2a Calculates the length of the WTG table and the number of work areas needed. [RES01810]

Calls the GETMAIN routines via branch entry and sets the area obtained to zeros. Builds the prefix. [RES01830]

When the DD statement contains DCB=DIAGNS=TRACE this routine gets a buffer into which control blocks and work areas being traced are put and loads the routine (IFG019TR) that selects and moves the control blocks. [RES01827]

Enqueues the TIOT before updating the DSAB, which ensures the unit will not be dynamically deallocated during O/C/EOV processing. Initializes GTF buffer if the optional module trace is active. [RES01850]

Loads the O/C/EOV-DADSM recovery routine and issues the ESATE macro to establish the address that is to get control if an abnormal termination condition is encountered. [RES06000]

3a IECRES INIT: If work area initialization is required, fills in the following fields in the work area: DXWDSORG, DSASCBAD, DXTCBADR, DXJSCBAD, DXDSABQD, DXTIOTAD, DXWMCR. [RES14000]

If the data set associated with the current DCB/ACB is not open (DEB=NO operand used in the macro to indicate that no DEB exists), this routine fills in the DXDSABAD, DXTIOTAD, DXUCBADR, and DXDEBXAD fields. [RES14030]

If the data set is open (DEB=YES used in IECRES macro), this routine fills in the DXDSABAD, DXTIOTAD, DXUCBADR, and DXDEBXAD fields, using pointers from the DEB. [RES14200]

Copies the user's DCB to protected storage. (DCBCOPY=TOWKAR operand used in IECRES macro.) [RES14800]

3b Copies the DCB in the protected storage to the user's storage. (DCBCOPY=FRWKAR operand used in IECRES macro.) [RES14820]

4 IECRES GET, PREFIX=YES | NO: entered by O/C/EOV and DADSM modules to get storage other than the initial O/C/EOV or DADSM work areas or the WTG table. Calls GETMAIN routines via branch entry and sets the storage obtained to zeros. If there is another prefix to which this work area is to be chained, this routine backward and forward chains the work area using pointers in the prefixes. [RES01830]

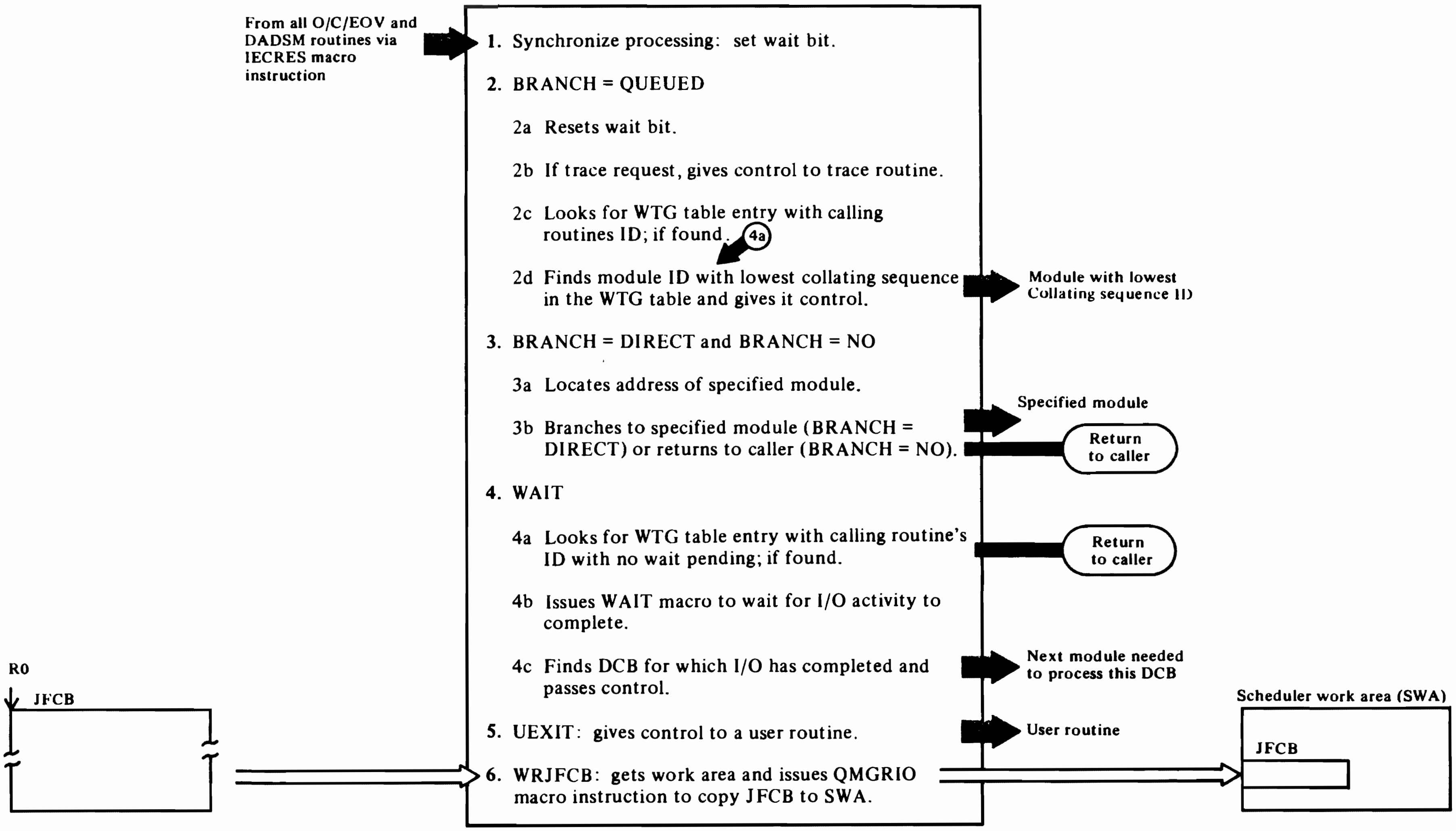
5 IECRES FREE: this routine is used (1) to free one work area or (2) to free the first work area obtained for an O/C/EOV or DADSM request and all the associated work areas (for example, to free a WTG table and all the work areas associated with the WTG table for an open request). The address of the associated work areas to be freed is obtained from the work area prefix.

5a IECRES FREE: entered by Open, Close, EOV, and DADSM this routine unchains the work area being released, and rechains the remaining work areas, if necessary. Calls the FREEMAIN routines via branch entry. [RES01860]

5b IECRES FREE,PREFIX=WTG | EOV | FIRST: entered by EOV, DADSM, etc., this routine dequeues the TIOT if it was enqueued during the IECRES GET routine (step 2a), deletes the recovery routine loaded in step 1b, and deletes the ESTAE address. [RES01880]

Releases the initial work area and all work areas chained from the initial work area. [RES01860]

Diagram 5.2. Transfer Control, Synchronize Processing, Write a JFCB


From all O/C/EOV and DADSM routines via IECRES macro instruction
R0
JFCB
4a
Module with lowest Collating sequence ID
Specified module
Return to caller
Return to caller
Next module needed to process this DCB
User routine
Scheduler work area (SWA)
JFCB

1. Synchronize processing: set wait bit.
2. BRANCH = QUEUED
   - 2a Resets wait bit.
   - 2b If trace request, gives control to trace routine.
   - 2c Looks for WTG table entry with calling routines ID; if found.
   - 2d Finds module ID with lowest collating sequence in the WTG table and gives it control.
3. BRANCH = DIRECT and BRANCH = NO
   - 3a Locates address of specified module.
   - 3b Branches to specified module (BRANCH = DIRECT) or returns to caller (BRANCH = NO).
4. WAIT
   - 4a Looks for WTG table entry with calling routine's ID with no wait pending; if found.
   - 4b Issues WAIT macro to wait for I/O activity to complete.
   - 4c Finds DCB for which I/O has completed and passes control.
5. UEXIT: gives control to a user routine.
6. WRJFCB: gets work area and issues QMGRIO macro instruction to copy JFCB to SWA.

Diagram 5.2. Transfer Control and Synchronize Processing

These subroutines of IFG019RA (also called the *O/C/EOV-DADSM service routine*) are called by the Open and Close modules to synchronize the parallel processing of multiple DCB/ACB Open and Close requests and to wait for events to complete. They are also called by the EOV and DADSM components to transfer control, and by IFG019RA itself during work area initialization to find the entry point address of a recovery routine.

IFG019RA

1 IECRES SYNCHDCB: before transferring control to access method executors and user routines (label processors, exit routines), the Open and Close modules issue this form of the macro instruction to bring all of the DCB/ACBs in the parameter list to the same stage of processing. This is necessary because more than one of the DCB/ACBs may require processing by an access method executor or a user routine. This routine sets the *psuedo-wait* or *synch* bit in the DXRESIND field of the O/C/EOV main work area. [RES0400]

2 IECRES LOAD,BRANCH=QUEUED or IECRES XCTL: This subroutine is called by O/C/EOV routines when control is to be passed to anoed to an O/C/EOV module, determined by the collating sequence of the module IDs in the WTG table entries. Control can also be returned to the calling routine if another DCB/ACB is to be processed by that module.

2a This routine turns off the wait bit in the DXRESIND field. [RES01900]

2b If the optional work area trace has been requested (DCB=DIAGNS=TRACE or AMP='TRACE' was coded in the DD statement) and the trace routine has been loaded, this routine passes control to the trace routine (IFG019TR). At each subsequent entry into the service routine (IFG019RA) in order to transfer control to another module, IFG019TR is given control. IFG019TR gathers control blocks and writes them using the GTRACE macro. [RES02000]

2c Examines each module ID in the WTG table to see if the calling routine's ID is there. If it is, the BRANCH=QUEUED request is treated like a WAIT request (step 4a). [RES02300]

2d If the calling routine's ID does not appear in the WTG table, this routine determines the ID with the lowest collating sequence, moves its name into the WTGMODNM field and its entry point address to WTGMODEP, and gives control to that routine via step 3. [RES02400]

3 IECRES LOAD,BRANCH=DIRECT: used when an O/C/EOV module must transfer control immediatel to another module (e.g., to the access method executors). [RES02800]

3a If the WTGMODEP field of the WTG table is not zero, the address of the module to receive control is already known. If WTGMODEP is zero the following procedure is used to get the entry-point address of the module:

(1) Calls IEAQCDSR to search the active LPA for a copy of the module

(2) Calls IEAVVMSR to search the pageable LPA

(3) Issues LOAD macro to search SYS1.LINKLIB [RES02840]

3b When the address of the module to be given control is obtained, it is given control using a branch instruction. If IECRES LOAD,BRANCH=NO is coded, control is returned to the calling routine after the address of the new module is obtained and moved to the WTGMODEP field. [RES02810+8]

4 IECRES WAIT is issued by the Open and Close routines to overlap I/O with processing on another DCB or ACB in the same request.

4a Looks for calling module's ID in the WTG table entries. If the ID is found and there is no wait pending, the calling routine is given control. [RES00600]

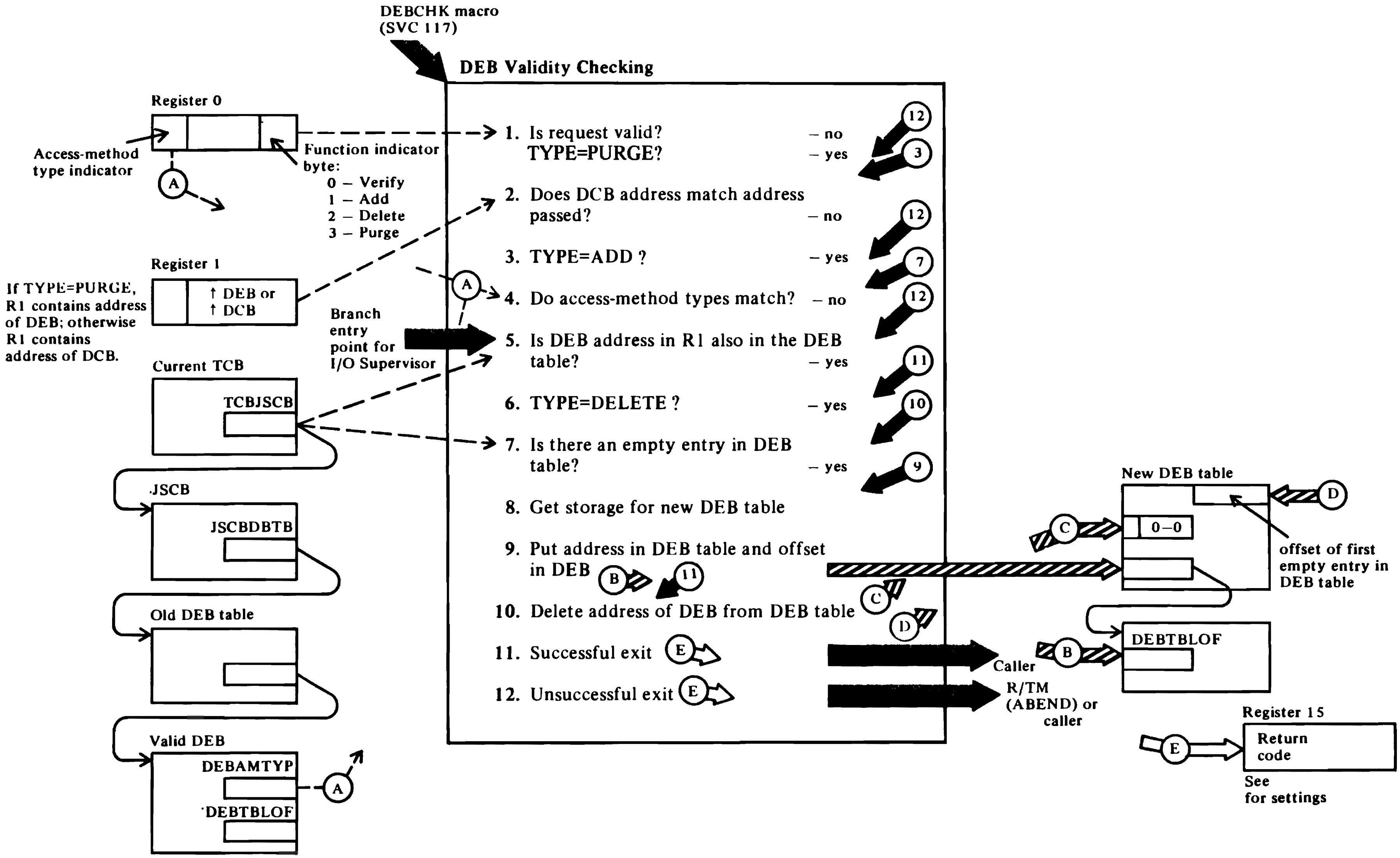
4b If a wait is pending for all DCBs (this is determined via the list of ECB pointers in the service routine work area), a WAIT macro with the LONG option is issued to suspend activity for this open request. [RES00800]

4c When an event is posted complete and control is returned following the WAIT macro, this routine determines which DCB is again ready to be processed. [RES00600]

5 IECRES UEXIT: issued to (1) branch directly to the caller if the caller is executing in a system storage protection key (key 0 through 7) or (2) to SYNCH (SVC 12) to the caller if the storage protection key is greater than 7 (user key). [RES09000]

6 IECRES WRJFCB: issued to update a JFCB in the scheduler work area (SWA). This routine gets a work area, copies the JFCB to the work area, and issues the QMNGRIO macro. [RES10000]

Diagram 5.3. DEB Validity Checking



DEBCHK macro (SVC 117)

DEB Validity Checking

1. Is request valid? – no
   TYPE=PURGE? – yes
2. Does DCB address match address passed? – no
3. TYPE=ADD ? – yes
4. Do access-method types match? – no
5. Is DEB address in R1 also in the DEB table? – yes
6. TYPE=DELETE ? – yes
7. Is there an empty entry in DEB table? – yes
8. Get storage for new DEB table
9. Put address in DEB table and offset in DEB
10. Delete address of DEB from DEB table
11. Successful exit
12. Unsuccessful exit

If TYPE=PURGE, R1 contains address of DEB; otherwise R1 contains address of DCB.

Diagram 5.3. DEB Validity Checking

This routine is entered from the first-level interrupt handler as a result of the issuance of the DEBCHK macro, the expansion for which terminates with an SVC 117. The DEBCHK macro is issued by programs in program state and in supervisor state. See *OS/VS2 System Programming Library: Data Management* for a description of the use of the DEBCHK macro.

IFGDEBCK

1 At entry to this routine, the TESTAUTH macro is issued to determine whether the caller is executing in a system storage protection key (key 0-7), in supervisor state, or is authorized under APF. If any one of these conditions is true, this routine initializes a set of authorized BXLE masks that are used to control the functions that are entered. Otherwise, this routine establishes a set of unauthorized BXLE masks which allow only the VERIFY option to be performed. [DCK00000]

If a valid TYPE parameter was specified, a check is made to determine whether the caller is authorized to use it. The caller must be executing in a system storage protection key (key 0 through 7), in supervisor state, or be authorized under APF in order to use the TYPE=ADD, DELETE, or PURGE. [DCK00100]

2 This routine is not executed if TYPE=PURGE. If TYPE is not PURGE, the DEB address is loaded into a register from the DCB pointed to by register 1. The DCB address is loaded from the DEBDCBAD field of the DEB. The DCB address passed is compared to the DCB address loaded. [DCK00110]

3 If TYPE=ADD is specified, the access method check is bypassed and control is passed to the add routine (step 7).

4 The high-order byte of register 0 contains a code that identifies the access method of the data set associated with the DEB being verified. (See *OS/VS2 System Programming Library: Data Management* for a list of the valid codes. [DCK00200]

If the access method value in the high-order byte of input register 0 is not specified (X'00'), the access method check is bypassed and control is passed to the verify routine (step 5). If the access method value is specified, the value is compared to the access method value in the DEBAMTYP field. [DCK00210]

5 When a program other than the I/O supervisor requests that a DEB address be verified, this routine branches and links to the verify routine of IFGDEBVR. [DCK00300]

The EXCP component of the I/O supervisor branches directly to the entry point to verify DEB addresses before issuing a start I/O (SIO) for an EXCP request. The conditions that will result in an error return are: (1) the DEB table does not exist, (2) the DEBTBLOF field is less than 1, (3) DEBTBLOF is larger than the DEB table, or (4) the DEB pointer and DEB table entry are not equal. [IFGDEBVR]

7 Before adding a DEB address to the DEB table, this routine verifies that the DEB whose address was passed in register 1 is on the DEB chain of the current TCB or, for a VSAM data set, on the DEB chain of the jobstep TCB. A test is made to make sure that the DEBAMTYP field and the DEBTBLOF field are set to zero. The second halfword of the DEB table is tested to determine whether there are any empty entries in the DEB table. [DCK00400]

8 If no DEB table exists for the job step or if there are no empty entries in the existing table, space for a new table is obtained. (The table cannot exceed 32,760 bytes.) If the old table was too small, the old table is moved to the new table, which is 80 bytes longer, and the old table is freed. [DCK00440]

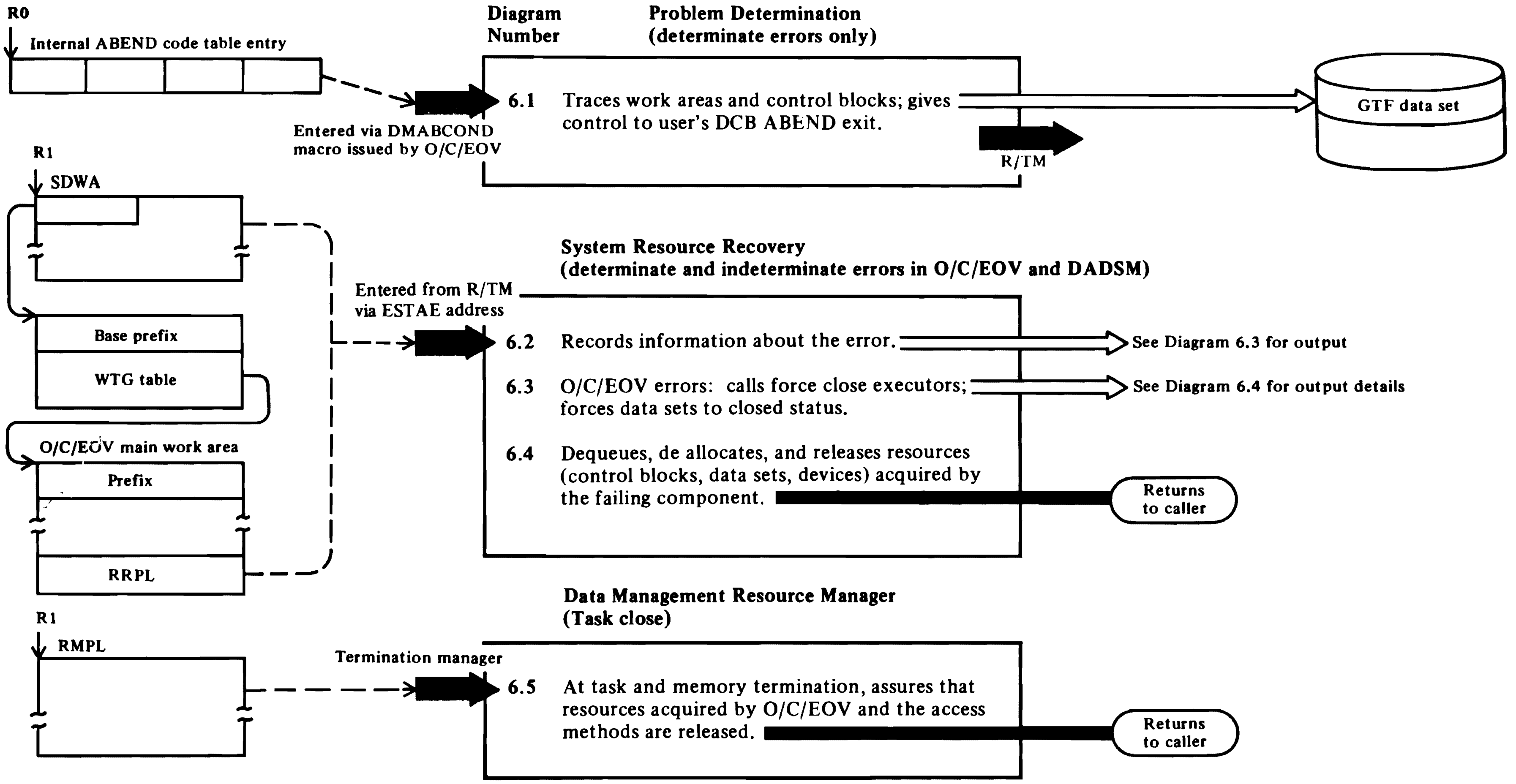
9 Enters the DEB address in the DEB table and updates the control field, if necessary. [DCK00420]

10 For TYPE=PURGE or DELETE, deletes DEB address from the DEB table and updates control field, if necessary. [DCK00500]

11 Register 15 is set to zero if no errors were encountered. [DCK02000]

12 If the caller is authorized and TYPE=ADD, DELETE, or VERIFY was specified, an ABEND macro is issued (16E system code). If TYPE=PURGE was specified, or if the caller is not authorized, returns to caller. In either case a return code is set in register 15. See the "Diagnostic Aids" section or *OS/VS2 System Programming Library: Data Management* for a description of the return codes issued. [DCK01000]

Diagram 6.0. Problem Determination, Recovery, and Termination—Overview


R0
Internal ABEND code table entry
Entered via DMABCOND macro issued by O/C/EOV
R1
SDWA
Base prefix
WTG table
O/C/EOV main work area
Prefix
RRPL
Entered from R/TM via ESTAE address
R1
RMPL
Termination manager
Diagram Number
R/TM
GTF data set
See Diagram 6.3 for output
See Diagram 6.4 for output details
Returns to caller
Returns to caller

Problem Determination (determinate errors only)

6.1 Traces work areas and control blocks; gives control to user's DCB ABEND exit.

System Resource Recovery (determinate and indeterminate errors in O/C/EOV and DADSM)

6.2 Records information about the error.

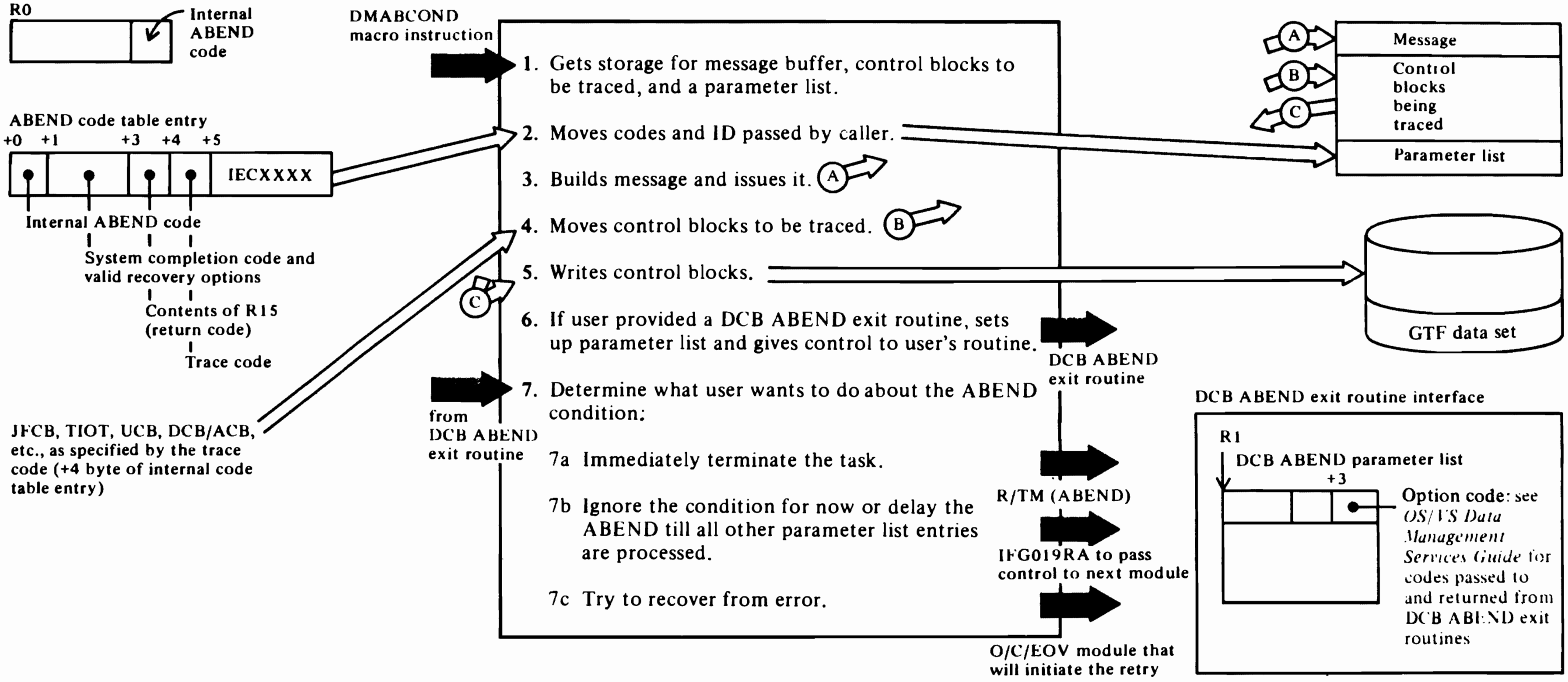
6.3 O/C/EOV errors: calls force close executors; forces data sets to closed status.

6.4 Dequeues, de allocates, and releases resources (control blocks, data sets, devices) acquired by the failing component.

Data Management Resource Manager (Task close)

6.5 At task and memory termination, assures that resources acquired by O/C/EOV and the access methods are released.

Diagram 6.1. Problem Determination—ABEND Interpretation and DCB ABEND Exit



1. Gets storage for message buffer, control blocks to be traced, and a parameter list.
2. Moves codes and ID passed by caller.
3. Builds message and issues it. (A)
4. Moves control blocks to be traced. (B)
5. Writes control blocks.
6. If user provided a DCB ABEND exit routine, sets up parameter list and gives control to user's routine.
7. Determine what user wants to do about the ABEND condition:

7a Immediately terminate the task.

7b Ignore the condition for now or delay the ABEND till all other parameter list entries are processed.

7c Try to recover from error.

R0
Internal ABEND code
ABEND code table entry
+0 +1 +3 +4 +5
IECXXXX
Internal ABEND code
System completion code and valid recovery options
Contents of R15 (return code)
Trace code
JFCB, TIOT, UCB, DCB/ACB, etc., as specified by the trace code (+4 byte of internal code table entry)
DMABCOND macro instruction
from DCB ABEND exit routine
C
A
B
Message
Control blocks being traced
Parameter list
GTF data set
DCB ABEND exit routine
R/TM (ABEND)
IFG019RA to pass control to next module
O/C/EOV module that will initiate the retry
DCB ABEND exit routine interface
R1
DCB ABEND parameter list
+3

Option code: see *OS/VS Data Management Services Guide* for codes passed to and returned from DCB ABEND exit routines

Diagram 6.1. Problem Determination—ABEND Interpretation and DCB ABEND Exit Recovery

If an abnormal termination (ABEND) condition is detected during O/C/EOV processing, the routine that detected the abnormal termination condition issues the DMABCOND macro instruction to call this function. The DMABCOND macro instruction passes the internal ABEND code, using which, this function writes a message and traces control blocks, blocks, and then, if the user has provided a DCB ABEND exit routine, passes control to the user's routine to allow him to attempt to recover. See *OS/VS Data Management Services Guide* for a description of the ABEND conditions from which the user can attempt to recover.

IFG0190P from Open
IFG0200P from Close
IFG0230P from TCLOSE
IFG0550P from EOV
IGG0196M from RDJFCB or the Access Method Open Executors
IGG0206M from the Access Method Close Executors

1 Registers 2 and 4 through 8 are saved in the SVRB extended save area. This routine uses the internal ABEND code passed in register 0 to refer to tables that contain the system ABEND and return code, a trace code that indicates which control blocks are to be written to the GTF trace data set for each ABEND condition, and the ID of the message that is to be issued to the programmer. See "Internal ABEND Codes" for a description of the codes and their meanings. [PDF01010]

2 This routine gets a work area, sets it to all zeros, and moves the internal code, return code, trace code and message ID into the work area. [PDF01040]

IFG0199B

3 Even if the user recovers successfully in his DCB ABEND exit routine (see step 6), a message is written to indicate the nature of the problem from which he recovered. This routine formats the message that is to be issued, completing all the fields: job name, step name, concatenation number, data set name, etc. The routing and descriptor codes are also moved into the message, then the message is issued. [PDF02000]

IFG0199D

4 Based on the internal ABEND code passed by the routine that issued the DMABCOND macro, this routine traces significant control blocks by writing them to the GTF data set. [PDF03040]

IFG0199E

6 Determines whether the user provided an active DCB ABEND exit routine address in his DCB exit list. If it was not provided, the ABEND is delayed, if possible; otherwise the task is immediately abnormally terminated by issuing the ABEND macro instruction. [PDF04010]

Copies the DCB from protected storage to the user's storage, sets the lock bit, and saves the contents of register 1 in the SVRB extended save area. Moves valid options to the user's parameter list and passes control to the user's DCB/ACB ABEND exit routine. [PDF04040]

7 When the user returns from his DCB ABEND exit routine, this routine examines the option mask (byte 3 of the parameter list) to determine what the user wants to do and if what he wants to do is valid under the conditions. If the user specified an invalid option, the ABEND is delayed, if possible. [PDF04060]

7a Issues the ABEND macro. [PDF04480]

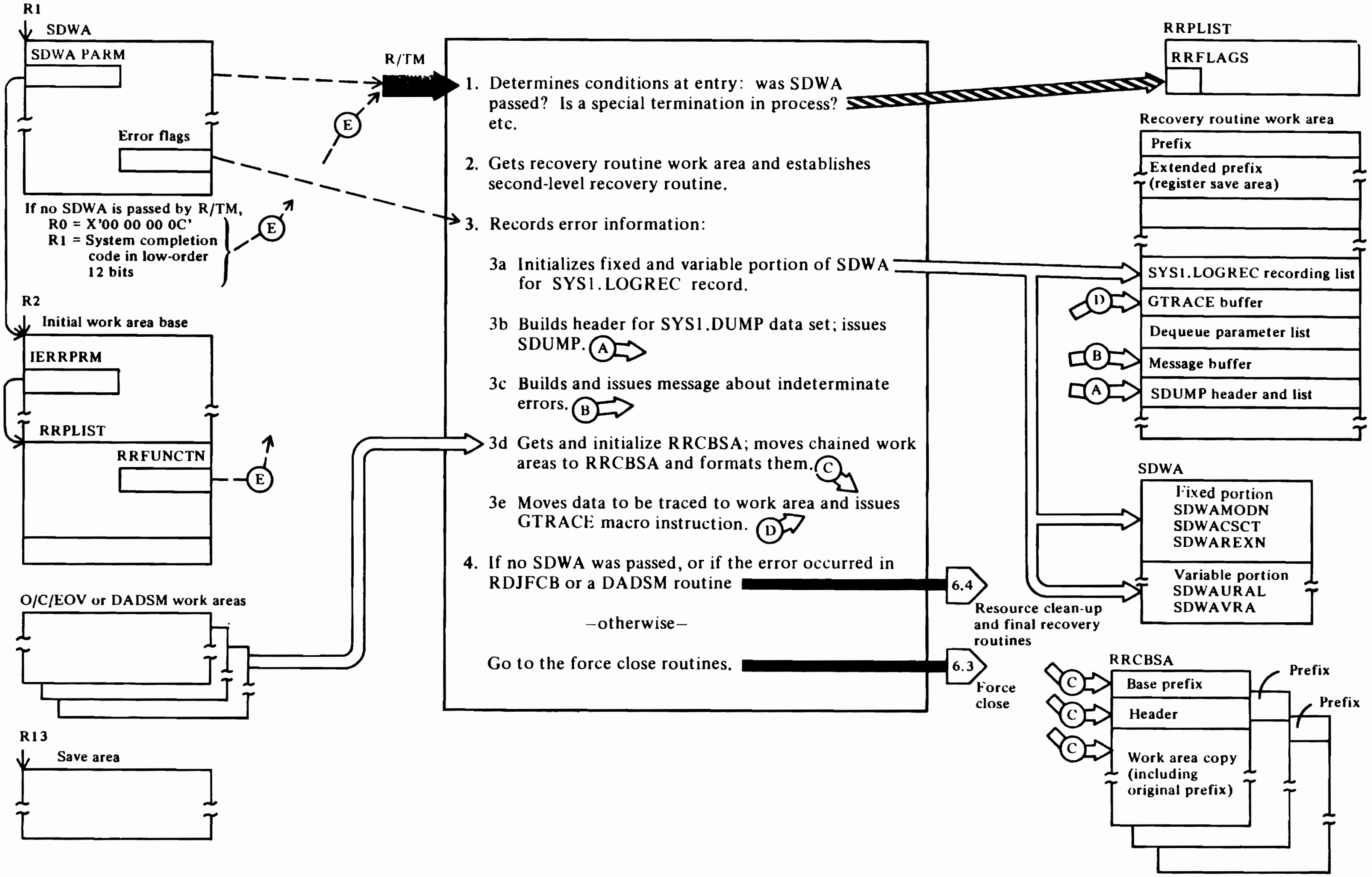
7b If the ABEND can be delayed or ignored under the conditions under which the error occurred, this routine turns off the busy bit in DCBOFLGS, sets the delay-ABEND indicator in the JFCB or the ignore indicator in the RRPL, and issues the IECRES LOAD,BRANCH=QUEUED macro to allow IFG019RA to use the WTG table to determine which O/C/EOV module is to get control next. [PDF04450]

Note that if the error occurred in an open or close access method executor, IFG019RA will give control to the access method executor determination function (module IFG0196V or IFG0200W) so that any other ACB/DCB being processed in parallel can be processed by the access method executors. [PDF04420]

7c If the user provided new volume serial numbers to be used in trying to recover, this routine moves them to the JFCB or JFCB extension. [PDF04080]

Frees the work area obtained in step 1 and the user's work area, if requested, then returns to the O/C/EOV routine that will attempt to recover. [PDF04520]

Diagram 6.2. Recovery—Initialization and Error Recording



1. Determines conditions at entry: was SDWA passed? Is a special termination in process? etc.

2. Gets recovery routine work area and establishes second-level recovery routine.

3. Records error information:

   3a Initializes fixed and variable portion of SDWA for SYS1.LOGREC record.

   3b Builds header for SYS1.DUMP data set; issues SDUMP. (A)

   3c Builds and issues message about indeterminate errors. (B)

   3d Gets and initialize RRCBSA; moves chained work areas to RRCBSA and formats them. (C)

   3e Moves data to be traced to work area and issues GTRACE macro instruction. (D)

4. If no SDWA was passed, or if the error occurred in RDJFCB or a DADSM routine → 6.4 Resource clean-up and final recovery routines

   —otherwise—

   Go to the force close routines. → 6.3 Force close

If no SDWA is passed by R/TM,
R0 = X'00 00 00 0C'
R1 = System completion code in low-order 12 bits

Diagram 6.2. O/C/EOV and DADSM Recovery—Initialization and Error Recording

The recovery initialization function is entered via the ESTAE interface from the recovery/termination manager (R/TM) when an abnormal termination condition or an indeterminate error (a program check or a machine check) is encountered during O/C/EOV or DADSM processing. The address of the first load of these recovery routines is established at the time the O/C/EOV or DADSM function gets a work area during initialization (see Diagram 5.1, IECRES GET).

IFG0RR0A

1 Internal flags are set that are used to determine subsequent recovery processing when this routine is entered under the following conditions: (1) no system diagnostic work area (SDWA) was obtained and (2) a special termination (e.g., the operator cancelled the job) is in process; this means that the recovery routine is not allowed to retry the function that failed. [START0A]

2 A work area is obtained from subpool 230 and the address of a second-level recovery routine, IFG0RR0F, is established. This routine is given control if an error occurs during the execution of the first-level recovery routines (IFG0RR0A, IFG0RR0B, or IFG0RR0E). [LVL20A]

3 The nature of the error information to be recorded is determined from the following conditions: (1) whether an SDWA was passed by R/TM; (2) whether the entry was via percolation from R/TM, the result of an abnormal termination condition, the result of an indeterminate error (a program check or a machine check); or (3) whether a special termination is in process. [ERRREC0A]

One or more of the following error recording subroutines is called:

3a Information is set up in the SDWA for inclusion of a record to be written to the SYS1.LOG data set on return to R/TM. [SUBLOG]

3b An SDUMP macro is issued to write the information to be dumped to the SYS1.DUMP data set. [SUBDUMP]

3c A WTO macro is issued to issue the IEC999I indeterminate error message. [SUBWTP]

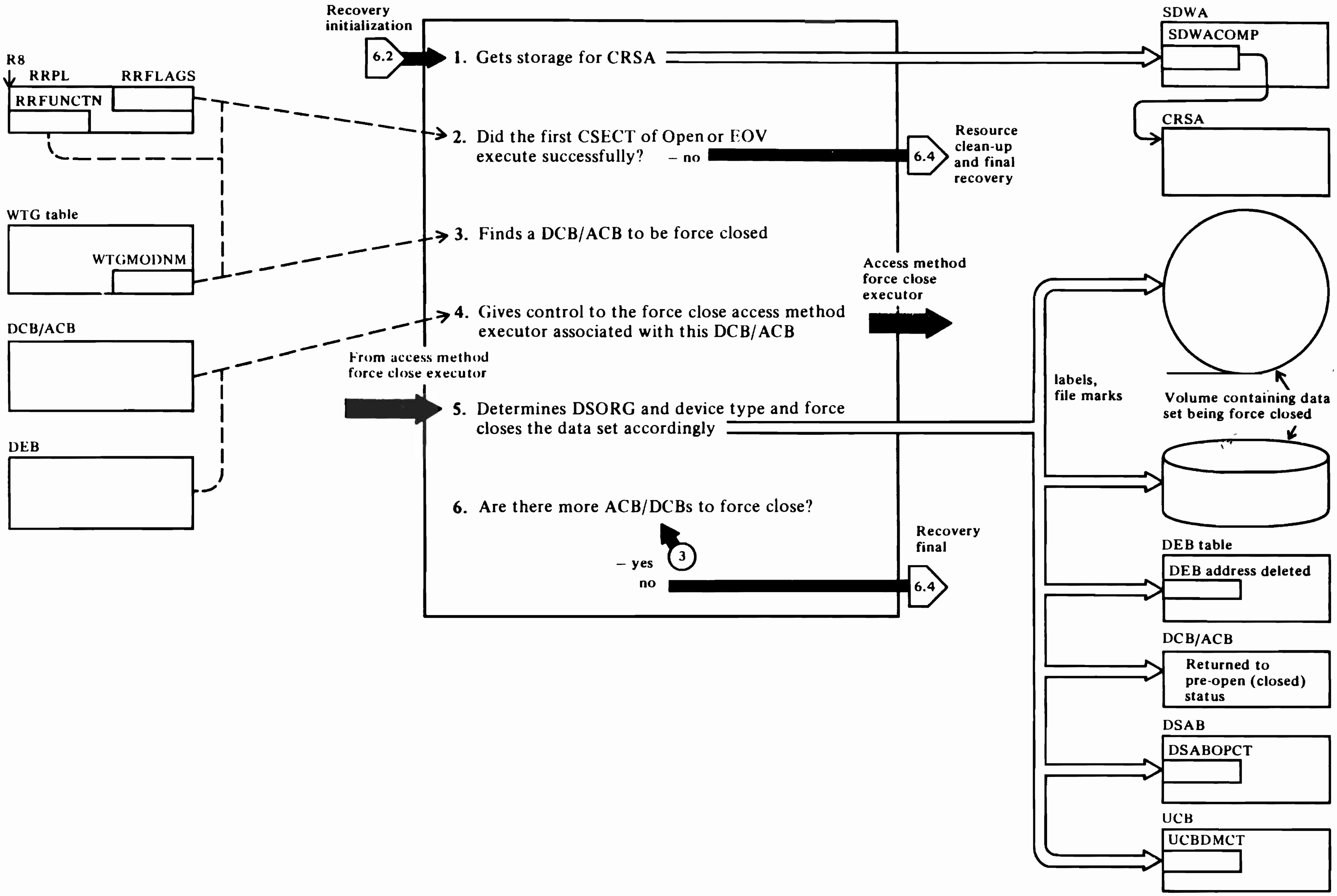
3d This subroutine obtains a work area in subpool 230 for building the recovery routine control block save area (RRCBSA). The chained work areas, including their prefixes, obtained by the O/C/EOV routines during normal processing are validated and copied into the RRCBSA segment. An RRCBSA will be built for a DADSM error only when an ISAM or VIO allocation is in progress when the error occurred. See the section "ABEND Error Recording from the System Recovery Routines" for a description of the RRCBSA. [SUB230]

3e This subroutine issues the GTRACE macro instruction to write the RRCBSA to the GTF data set.

4 This routine determines the next function to receive control. The recovery routine clean-up and termination function (Diagram 6.4) is given control if (1) no SDWA was passed, or (2) the failure occurred in a DADSM routine or in RDJFCB. The Force Close routine, IFG0RR0B (see Diagram 6.3), is given control if the failure occurred in an O/C/EOV routine. [NEXTMOD]

This routine reestablishes register contents necessary to continue processing after the second-level recovery routine, IFG0RR0F, returns control. [RETRY0A]

Diagram 6.3. Recovery—Force Close



1. Gets storage for CRSA
2. Did the first CSECT of Open or EOV execute successfully? – no
3. Finds a DCB/ACB to be force closed
4. Gives control to the force close access method executor associated with this DCB/ACB
5. Determines DSORG and device type and force closes the data set accordingly
6. Are there more ACB/DCBs to force close? – yes 3 no

Recovery initialization
6.2
R8
RRPL
RRFLAGS
RRFUNCTN
WTG table
WTGMODNM
DCB/ACB
DEB
Resource clean-up and final recovery
6.4
Access method force close executor
From access method force close executor
Recovery final
6.4
SDWA
SDWACOMP
CRSA
labels, file marks
Volume containing data set being force closed
DEB table
DEB address deleted
DCB/ACB
Returned to pre-open (closed) status
DSAB
DSABOPCT
UCB
UCBDMCT

Diagram 6.3. Recovery—Force Close

IFG0RR0B

This function is entered from the initialization and error recording function when a determinate or indeterminate error occurs during Open, Close, or EOV processing. This function forces data sets to closed status in order to preserve the integrity of the data sets and to release all the resources that were obtained during O/C/EOV processing for the task that failed. Force closing involves performing some or all of the functions provided by the Close component. The stage of processing at which the failure occurred determines how many of the Close functions must be performed.

1 The CRSA is used by the issuer of the OPEN, CLOSE, FEOV, EOV macro in his ESTAE exit, which can receive control from R/TM after Force Close has completed. When the CRSA has been obtained, this routine puts its address in the SDWACOMP field of the SDWA. [GETCRSA]

2 If the error occurred during Open or EOV processing, none of the Force Close functions will be done unless the first CSECT (IGC00011, IGC0003A, or IGC0005E) was successfully executed. [CHKFIN1]

3 If entry to the recovery routine resulted from an indeterminate error (e.g., a program check), all of the DCB/ACBs that have not yet been closed will be force closed.

If entry to the recovery routine resulted from an ABEND condition detected in the O/C/EOV components, for which a DMABCOND macro was issued, followed by an ABEND macro, this routine will determine which of the DCB/ACBs in the parameter list needs to be force closed: (1) if the error was detected by an Open routine, this routine will select any DCB/ACB not yet marked as open in the DCB/ACB OFLGS field; (2) if the error was detected by a Close routine, this routine will select any DCB/ACB not yet marked as not open in the DCB/ACB OFLGS field; and (3) if the error was detected by an EOV routine, only one DCB/ACB will be in the parameter list, and it will be force closed, if normal EOV processing has gone beyond the point where a normal close could occur. Since a multiple OPEN or CLOSE request may be in progress, it is important to note that all the DCB/ACBs that can be successfully opened or closed will have already been processed. Only the DCB/ACBs for which errors occurred must be forced to closed status. [SELTCALL]

4 This routine determines whether an access method executor should be given control during Force Close processing. This depends on which O/C/EOV functions had been completed when the error occurred. The chronological order in which the O/C/EOV components are called, relative to the access method executors follows:

1. SVC 19 receives control from the SLIH
2. Open access method executors or open strings (e.g., VSAM or VTAM) given control
3. Open access method executors or open strings complete successfully
4. Open completes successfully
5. EOV (SVC 55 or 31) gets control from SLIH
6. EOV completes successfully
7. Close (SVC 20) gets control from the SLIH
8. Close access method executors or close strings given control
9. Close executors or close strings complete successfully
10. Close completes successfully

If an error is encountered during the phase between items 1 and 2 or between items 9 and 10, only the access method string force close executors (e.g., VSAM force close executor) could be given control. If the error occurs at any point after the Open access method executor receives control (item 2) and before the Close access method executors return control (item 9) a force close executor will be called by this routine, if one exists for the particular data set organization. [CTRCALL]

Using the MACRF and DSORG field of the DCB, the DSORG and AMETH fields of the ACB, the TIOELINK field of the TIOT (to test for spooled SYSIN and SYSOUT data sets), this routine determines which force close executor is to be given control. [FCXCHK]

This routine passes control and the following registers to the force close executors: R2 ▸copy of DCB/ACB being force closed; R4 ▸O/C/EOV main work area associated with the DCB/ACB to be force closed; R5recovery routine work area; R6 ▸WTG table for this request; R7 ▸SDWA; R8 ▸RRPLIST; R9 ▸DD entry or is set to zero; and R11 ▸DEB or is set to zero. [FCXIFACE]

When the force close executors return, register 15 contains one of the following codes: 00—OPEN macro may be reissued for this DCB/ACB; 04—OPEN macro should not be reissued for this DCB/ACB; and 08—the DCB/ACB is not to be restored to pre-open status and OPEN should not be reissued. [FCXRETRN]

5 When the force close executors return control, this routine determines which of the force close subroutines are to be executed. For a tape or direct access data set, selection of the force close subroutines depends on (1) for data sets on tape on the type of label being used (SL, AL or NSL) and on whether the data set was opened for input or output, and (2) for data sets on direct-access devices, whether the EXCP access technique is being used, whether the data set was opened for input or output, and whether a physical-sequential data set is being processed (DSORG = PS or PO). [POSTFCX]

Several force close strings, consisting of subroutine calls, are defined in the following paragraphs. The calling routine that calls each of the force close subroutines is defined below. The label identifying the calling routines follows their description. The numbers of each of the subroutines called are associated with the labels. For example, the calling routine that handles force close processing for ACBs representing SYSIN and SYSOUT data sets can be found at label SPACB. The routine at label SPACB calls subroutines numbered 1, 12, 13, 15, 18, and 19. This is shown as:

[SPACB1,12,13,15,18,19]

Descriptions of the functions performed by the numbered subroutines follow the calling routine descriptions.

A VSAM data set or data space is to be force closed. [VSAM13,19]

A SYSIN or SYSOUT (spooled) data set represented by an ACB is to be forced to closed status. [SPACB1,12,13,15,18,19]

A SYSIN or SYSOUT (spooled) data set represented by a DCB is to be forced to closed status. [SPDCB11,18,19]

A VTAM data set is to be forced to closed status. [VTAM18,19]

A dummy data set is to be forced to closed status. [DUMMY1,9,11,12,13,16,18,19]

A data set on a tape unit that is not ready is to be forced to closed status. [TAPNR1,9,10,11,12,13,14,16,17,18,19]

An input data set on tape with standard labels (SL/AL or SUL/AUL) is to be forced to closed status. [TAPSLIN1,6,7,9,10,11,12,13,14,16,17,18,19]

An output data set on tape, with standard labels (SL/AL or SUL/AUL), is to be forced to close status. [TAPSLOUT1,5,6,7,9,10,11,12,13,14,16,17,18,19]

A data set on tape with non-standard labels (NSL), with no labels, or for which label processing was bypassed (NL or BLP) is to be forced to closed status. [TAPNL1,4,6,7,9,10,11,12,13,14,16,17,18,19]

An input data set on a direct access device or a data set on a direct access device being accessed using the EXCP access technique (MACRF=E), where a device dependent section is not present, or a data set on a direct access device for which the DSORG was not specified as PS or PO (physical sequential or partitioned) is to be forced to closed status. [DA11,6,9,10,11,12,13,14,16,17,18,19]

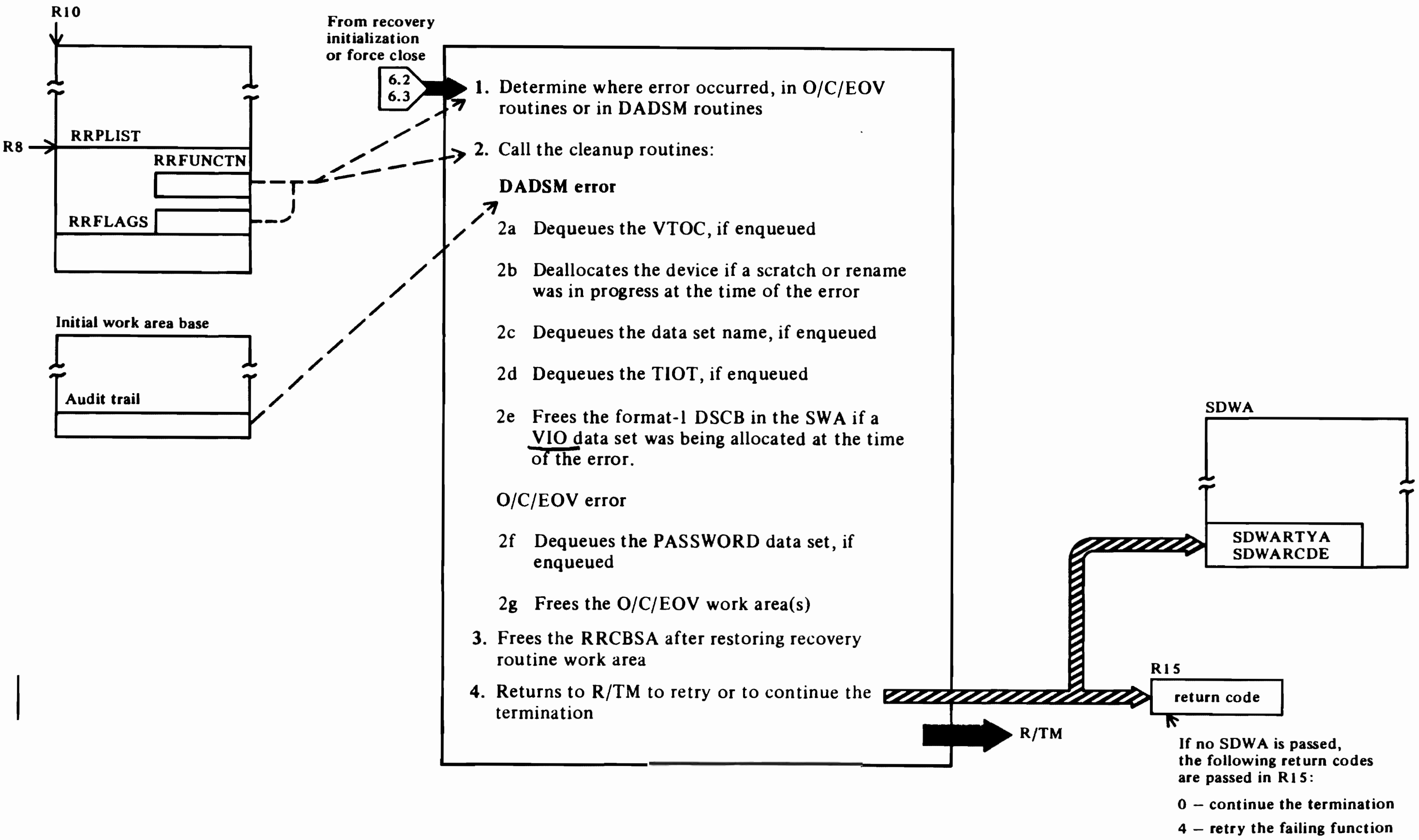
An output data set on a direct access device for which a DSORG or PS or PO (physical sequential or partitioned) was specified. [DA31,2,3,6,9,10,11,12,13,14,16,17,18,19]

When none of the previous conditions apply (that is, the data set is not VSAM, VTAM, or SYSIN or SYSOUT, not on tape or direct access), this routine calls subroutines that force the data set to closed status. [OTHER1,8,9,10,11,12,13,16,17,18,19]

The functions performed by each of the subroutines called by the previous routines are:

1. Deletes the address of the DEB associated with the data set from the DEB table. [DEBCKDEL]
2. Writes an updated format-1 DSCB for the data set being force closed to the VTOC. [WRDSCB]
3. Writes a file mark at the end of the direct access data set. [WRFMARK]
4. Writes a tape mark at the end of the data. [WRTM]
5. Writes a tapemark, labels, and another tapemark. [WTMLBLTM]
6. Builds and writes an SMF record to record data set activity. [CHKSMF]
7. Rewinds the tape volume currently being processed. [VOLDISP]
8. Frees the error recovery procedure (ERP) work area used for 3211 printer. [CHKUNREC]
9. Deletes the access method processing routines and their associated I/O appendages. [CHKSUBNM]
10. Frees the user-totalling save area. [CHKUTOT]
11. Restores the DCB to pre-open status. [DCBREST]
12. Removes the DEB associated with the data set from the TCB-DEB chain. [DCHNDEB]
13. Decrements the count of number of data sets in the DSAB (the DSABOPCT field). [DECRDSAB]
14. Decrements the count of the number of open data sets in the UCB (UCBDMCT field). [DECRDMCT]
15. Restores the ACB to pre-open status. [ACBREST]
16. Frees any IRBs associated with the data set. [IRBCHK]
17. Frees any RQEs associated with the data set. [RRQCHK]
18. Performs final DCB/ACB restoration. [FINREST]
19. Copies the DCB/ACB in protected storage to the user's storage. [COPYBACK]

Diagram 6.4. Recovery—Resource Clean-Up and Final Processing



1. Determine where error occurred, in O/C/EOV routines or in DADSM routines

2. Call the cleanup routines:

   DADSM error

   2a Dequeues the VTOC, if enqueued

   2b Deallocates the device if a scratch or rename was in progress at the time of the error

   2c Dequeues the data set name, if enqueued

   2d Dequeues the TIOT, if enqueued

   2e Frees the format-1 DSCB in the SWA if a VIO data set was being allocated at the time of the error.

   O/C/EOV error

   2f Dequeues the PASSWORD data set, if enqueued

   2g Frees the O/C/EOV work area(s)

3. Frees the RRCBSA after restoring recovery routine work area

4. Returns to R/TM to retry or to continue the termination

If no SDWA is passed, the following return codes are passed in R15:

0 – continue the termination

4 – retry the failing function

Diagram 6.4. Resource Clean-up and Final Recovery

IFG0RR0E

1 This routine determines whether the failure occurred in O/C/EOV or DADSM.

- The following subroutines are executed when the DADSM functions fail:

2a The DEQ macro is issued with major name of "SYSVTOC" and minor name of the volume serial number of the volume being processed by the DADSM routine that failed. [SUBVTOC]

2b If the UCB associated with a data set being scratched or renamed was dynamically allocated, this routine issues the DYNALLOC macro to deallocate the unit. [SUBALLOC]

2c Issues the DEQ macro with major name of "SYSDSN" and minor name of the data set name that was being processed by the DADSM routine that failed. [SUBDSN]

2d This routine issues the DEQ macro with major name of "SYSZTIOT" and minor name of the ASID/DSAB address. [SUBTIOT]

2e If a VIO data set was being allocated when the DADSM routine failed, this routine frees the virtual DSCB that was being used to make the allocation. [SUBSWA]

- The following subroutines are called when O/C/EOV routines fail:

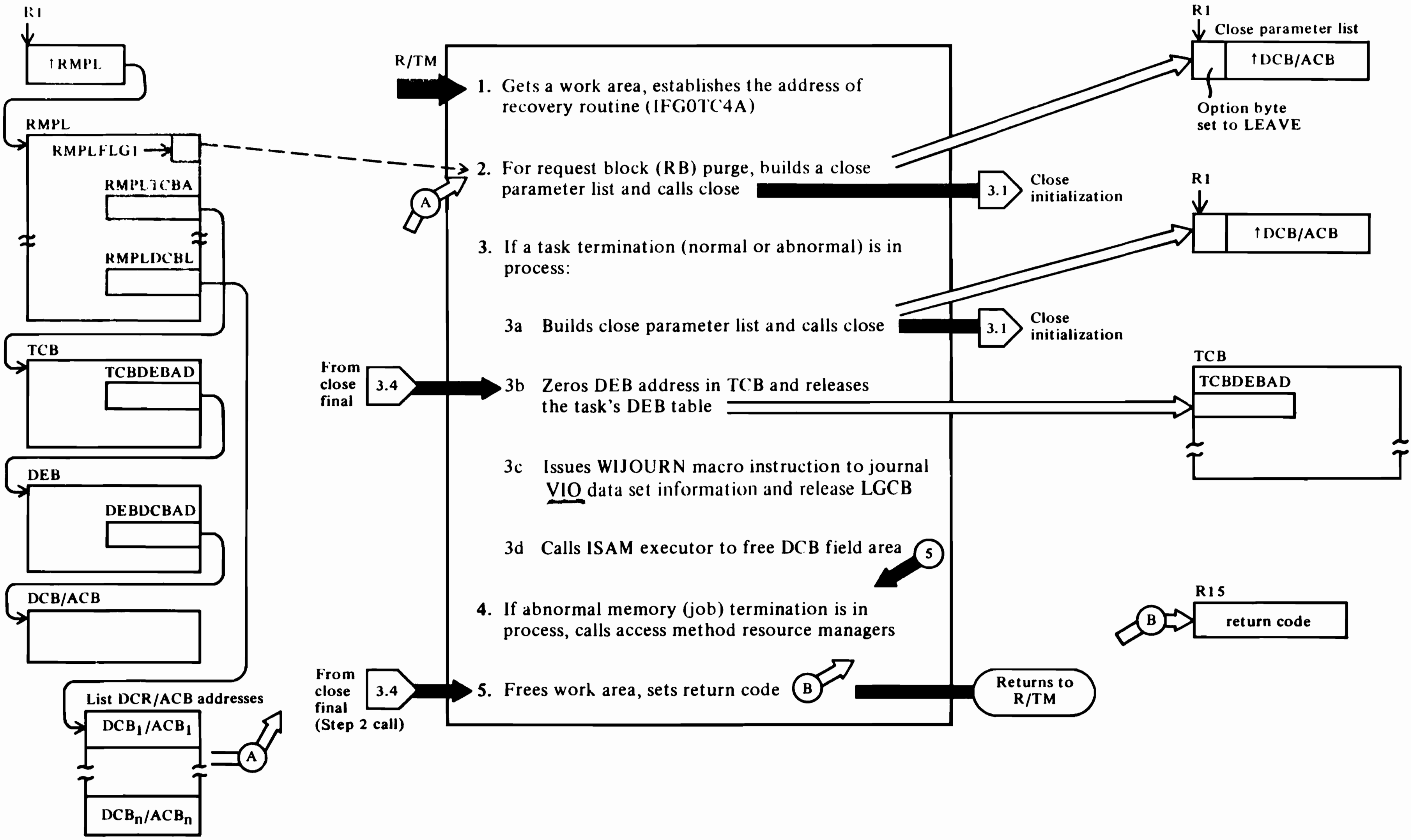
2f Issues the DEQ macro with major name "SYSZPSWD" and minor name of the data set. [SUBPASS]

2g Frees the RRCBSA (built in IFG0RR0A) after restoring the O/C/EOV chained areas. The IECRES FREE,PREFIX=YES,A=(1) macro is used to free all of the RRCBSA. [SUBFREE]

3 Releases the work area obtained in IFG0RR0A (see Diagram 6.2). [TRRWAFRE]

4 This routine returns to R/TM. A direct branch is used to return control if no SDWA was passed. The SETRP macro is used if an SDWA was passed. If no SDWA is available, register 15 is used to communicate the request to continue the abnormal termination. If an SDWA is available, the request to continue the termination or to retry the failing function is communicated by the SETRP macro instruction. A retry is attempted only if (1) a special termination is not in progress, (2) an SDWA was passed, and (3) a retry address of the Open or Close routine has been provided. A retry address will be provided only when an ABEND condition has been detected and the IGNORE option (see Diagram 6.1) is allowed and chosen by the user's DCB ABEND exit routine. [PERCBR] if direct branch to R/TM, [PRECSET] if SETRP is used to return to R/TM, or [MLRETRY] if a retry is to be attempted.

Diagram 6.5. Task Close (Data Management Resource Manager)



1. Gets a work area, establishes the address of recovery routine (IFG0TC4A)

2. For request block (RB) purge, builds a close parameter list and calls close

3. If a task termination (normal or abnormal) is in process:

   3a Builds close parameter list and calls close

   3b Zeros DEB address in TCB and releases the task's DEB table

   3c Issues WIJOURN macro instruction to journal VIO data set information and release LGCB

   3d Calls ISAM executor to free DCB field area

4. If abnormal memory (job) termination is in process, calls access method resource managers

5. Frees work area, sets return code

Diagram 6.5 Task Close (Data Management Resource Manager)

The Task Close function is called by the termination component of the recovery/termination manager (R/TM). It is called to release the resources that were obtained during open processing by the Open component. Five conditions are handled by Task Close: request block (RB) purge in step 2; normal and abnormal task termination in step 3; normal memory termination; and abnormal memory termination in step 4. The condition to be handled is determined based on the contents of the RMPLFLG1 field, established by R/TM.

IFG0TC0A

1 At entry, this routine obtains a work area and establishes the address of the recovery routine (IFG0TC4A) that is to be called by R/TM if Task Close fails. [IFG0TC0A]

2 When a request block purge is requested, this routine builds a Close parameter list containing the address of only one DCB/ACB. The LEAVE option is specified in the high-order byte of the entry. The LEAVE option tells Close to position the data set at its logical end. The DCB/ACB address is obtained from the list of ACB/DCB addresses pointed to from the RMPL (RMPLDCBL). Close is called once for each entry in the list of ACB/DCBs addresses. [IFG0TC0B(TC0BRBP)]

This routine issues the CLOSE macro, once for each entry in the list of ACB/DCBs. [IFG0TC1B]

When each entry in the RMPLDCBL has been closed, this routine exits via step 5. [IFG0TC3A]

3 This routine is entered when R/TM indicates that a task is being terminated, either normally or abnormally. [IFG0TC0A]

3a This routine picks up the address of a DCB from each DEB in the TCB-DEB chain, points to it with register 1 (constituting a Close parameter list), and passes control to Close (Diagram 3.1) to close the data set. [IFG0TC0B(TC0BTSKT)]

3b When each data set represented by ACB/DCB in the TCB-DEB chain has been closed, this routine zeros the pointer to the DEB chain in the TCB and frees the DEB table. Note: the DEB table is released only if the terminating task is a jobstep task and the JSCB of the mother task and the terminating task are not the same. [IFG0TC0C]

3c When a DCB/ACB that has been closed represents a VIO data set, this routine issues the WIJOURN macro instruction to ensure that the format-1 DSCB and the data set page correspondence table (DSPCT) for VIO data sets are available to the restart or a jobstep restart component, in case a restart at a checkpoint is subsequently performed. [IFG0TC0D]

For normal/abnormal job step task termination, this routine issues the WIJOURN macro instruction to ensure that the format-1 DSCB and the data set page correspondence table (DSPCT) for VIO data sets are available to the restart or a jobstep restart component, in case a restart at a checkpoint or an automatic step restart is subsequently performed. [IFG0TC0D]

3d This routine calls the ISAM executor (IGG0202C) to ensure that the DCB field area (DCBFA) has been freed. See *OS/VS2 ISAM Logic* for additional details. [IFG0TC0F]

4 Task close is called during a normal memory termination, but no processing takes place; however, when called during abnormal memory termination, this routine calls several resource managers to ensure that any storage acquired by data management components is freed and that any resources enqueued upon are dequeued.

Calls the VIO resource manager, IDAVBPS2, to free logical group control blocks (LGCB), and release VIO data sets or logical groups that have not been journaled (that cannot be reactivated). [IFG0TC0E]

Calls the ISAM executor, IGG0202C, to ensure that the DCB field area is freed. [IFG0TC0F]

Calls the VSAM O/C/EOV executor, IDAOCEA2, to free resources. See *OS/VS2 Virtual Storage Access Method (VSAM) Logic*. [IFG0TC0G]

Calls the VSAM catalog O/C/EOV executor, IDACAT13, to free resources. See *OS/VS2 Catalog Management Logic*. [IFG0TC0I]

Calls GAM resource manager, IGG0203Z, to free resources. See *OS/VS Graphic Access Method Logic*. [IFG0TC0H]

5 Frees work area obtained in step 1, puts a return code in register 15, and returns to caller (R/TM). If register 15 is not zero an abnormal termination is issued by R/TM with system code C03. [IFG0TC3A]

PROGRAM ORGANIZATION

The charts that follow show flow of control among the O/C/EOV object modules. Each box represents an object module. Object modules are the same as control sections in the O/C/EOV routines. The module names on the left and above each box are the modules that enter the module represented by the box; the module name below and to the right of each box are the modules to which the module represented by the box can pass control.

This section can be used in conjunction with the method of operation diagrams in Section 2 to determine the function provided by each module. Section 4, the "Directory," which lists module names in alphameric sequence and tells on which method of operation diagrams and in which chart in this section an object module may be found.

There are several paths through the O/C/EOV modules that are most commonly used. Several factors determine which of these common paths are taken: the function being performed (for example, opening a data set); the kind of device the data set being processed resides on; and whether the data set is being opened for input or output. Other factors also affect which path through the O/C/EOV modules is taken, but not significantly. Many of the more common paths through the O/C/EOV modules appear in the module lists in in the "Module Listing Routine Examples" section. You may want to examine these lists in conjunction with the charts that follow.

Legend:

IFG0192G

The module names in italics that appear to the right or left of a module box in italics are modules that are documented in another program logic manual, usually in an access method PLM.

(R14=*multiple of 4*)

Many of the O/C/EOV modules have branch tables as their entry points. The function performed by these modules depends on the contents of register 14 at entry. The register 14 contents are provided here to indicate exit and entry conditions. When no register 14 contents are shown, you may assume that (1) the initial entry (R14 = 0) is being used, or (2)the module being entered does not have a branch table.

(1,2,3,...)

These numbers are the O/C/EOV internal ABEND codes passed by the module they appear with to the appropriate problem determination routine (IFG0190P, IFG0200P, IFG0230P, IFG0550P, or IGG0196M). Each internal code is associated with a system ABEND code and return code passed in register 15. A cross-reference table for these internal codes, system codes, and return codes, as well as the message associated with each, appears in the "Diagnostic Aids" section.

[1.2]

The numbers in brackets following the name and description of a module receiving control are off-page connectors. They tell you which chart to turn to next if you are following a particular path through the component. Some of the exit routines are not followed by brackets. You will find these modules (1) on the same chart or (2) if the module name is in italics—for example, *IDA0192G*—in another PLM.

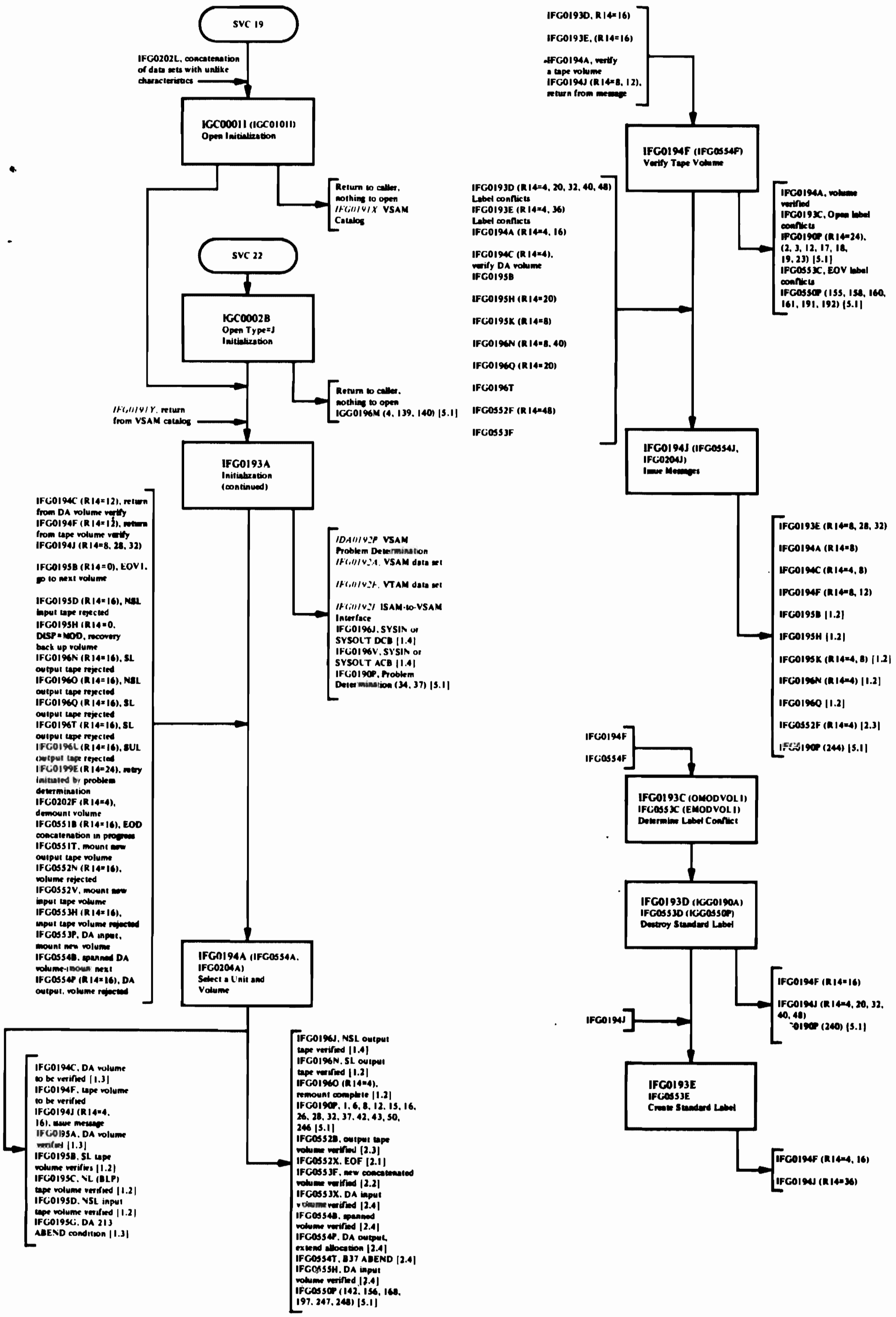

SVC 19
IFG0202L, concatenation of data sets with unlike characteristics
IGC00011 (IGC01011) Open Initialization
Return to caller, nothing to open
IFG0191X VSAM Catalog
SVC 22
IGC0002B Open Type=J Initialization
Return to caller, nothing to open
IGG0196M (4, 139, 140) [5.1]
IFG0191Y, return from VSAM catalog
IFG0193A Initialization (continued)
IFG0194C (R14=12), return from DA volume verify
IFG0194F (R14=12), return from tape volume verify
IFG0194J (R14=8, 28, 32)
IFG0195B (R14=0), EOV1, go to next volume
IFG0195D (R14=16), NSL input tape rejected
IFG0195H (R14=0, DISP=MOD, recovery back up volume
IFG0196N (R14=16), SL output tape rejected
IFG0196O (R14=16), NSL output tape rejected
IFG0196Q (R14=16), SL output tape rejected
IFG0196T (R14=16), SL output tape rejected
IFG0196U (R14=16), SUL output tape rejected
IFG0199E (R14=24), retry initiated by problem determination
IFG0202F (R14=4), demount volume
IFG0551B (R14=16), EOD concatenation in progress
IFG0551T, mount new output tape volume
IFG0552N (R14=16), volume rejected
IFG0552V, mount new input tape volume
IFG0553H (R14=16), input tape volume rejected
IFG0553P, DA input, mount new volume
IFG0554B, spanned DA volume-mount next
IFG0554P (R14=16), DA output, volume rejected
IDA0192P VSAM Problem Determination
IFG0192A, VSAM data set
IFG0192F, VTAM data set
IFG0192I ISAM-to-VSAM Interface
IFG0196J, SYSIN or SYSOUT DCB [1.4]
IFG0196V, SYSIN or SYSOUT ACB [1.4]
IFG0190P, Problem Determination (34, 37) [5.1]
IFG0194A (IFG0554A, IFG0204A) Select a Unit and Volume
IFG0194C, DA volume to be verified [1.3]
IFG0194F, tape volume to be verified
IFG0194J (R14=4, 16), issue message
IFG0195A, DA volume verified [1.3]
IFG0195B, SL tape volume verifies [1.2]
IFG0195C, NL (BLP) tape volume verified [1.2]
IFG0195D, NSL input tape volume verified [1.2]
IFG0195G, DA 213 ABEND condition [1.3]
IFG0196J, NSL output tape verified [1.4]
IFG0196N, SL output tape verified [1.2]
IFG0196O (R14=4), remount complete [1.2]
IFG0190P, 1, 6, 8, 12, 15, 16, 26, 28, 32, 37, 42, 43, 50, 246 [5.1]
IFG0552B, output tape volume verified [2.3]
IFG0552X, EOF [2.1]
IFG0553F, new concatenated volume verified [2.2]
IFG0553X, DA input volume verified [2.4]
IFG0554B, spanned volume verified [2.4]
IFG0554P, DA output, extend allocation [2.4]
IFG0554T, B37 ABEND [2.4]
IFG0555H, DA input volume verified [2.4]
IFG0550P (142, 156, 168, 197, 247, 248) [5.1]
IFG0193D, R14=16)
IFG0193E, (R14=16)
IFG0194A, verify a tape volume
IFG0194J (R14=8, 12), return from message
IFG0194F (IFG0554F) Verify Tape Volume
IFG0193D (R14=4, 20, 32, 40, 48) Label conflicts
IFG0193E (R14=4, 36) Label conflicts
IFG0194A (R14=4, 16)
IFG0194C (R14=4), verify DA volume
IFG0195B
IFG0195H (R14=20)
IFG0195K (R14=8)
IFG0196N (R14=8, 40)
IFG0196Q (R14=20)
IFG0196T
IFG0552F (R14=48)
IFG0553F
IFG0194A, volume verified
IFG0193C, Open label conflicts
IFG0190P (R14=24), (2, 3, 12, 17, 18, 19, 23) [5.1]
IFG0553C, EOV label conflicts
IFG0550P (155, 158, 160, 161, 191, 192) [5.1]
IFG0194J (IFG0554J, IFG0204J) Issue Messages
IFG0193E (R14=8, 28, 32)
IFG0194A (R14=8)
IFG0194C (R14=4, 8)
IFG0194F (R14=8, 12)
IFG0195B [1.2]
IFG0195H [1.2]
IFG0195K (R14=4, 8) [1.2]
IFG0196N (R14=4) [1.2]
IFG0196Q [1.2]
IFG0552F (R14=4) [2.3]
IFG0190P (244) [5.1]
IFG0194F
IFG0554F
IFG0193C (OMODVOL1)
IFG0553C (EMODVOL1)
Determine Label Conflict
IFG0193D (IGG0190A)
IFG0553D (IGG0550P)
Destroy Standard Label
IFG0194F (R14=16)
IFG0194J (R14=4, 20, 32, 40, 48)
IFG0190P (240) [5.1]
IFG0194J
IFG0193E
IFG0553E
Create Standard Label
IFG0194F (R14=4, 16)
IFG0194J (R14=36)

Chart 1.1. Open Initialization and Volume Mount and Verification

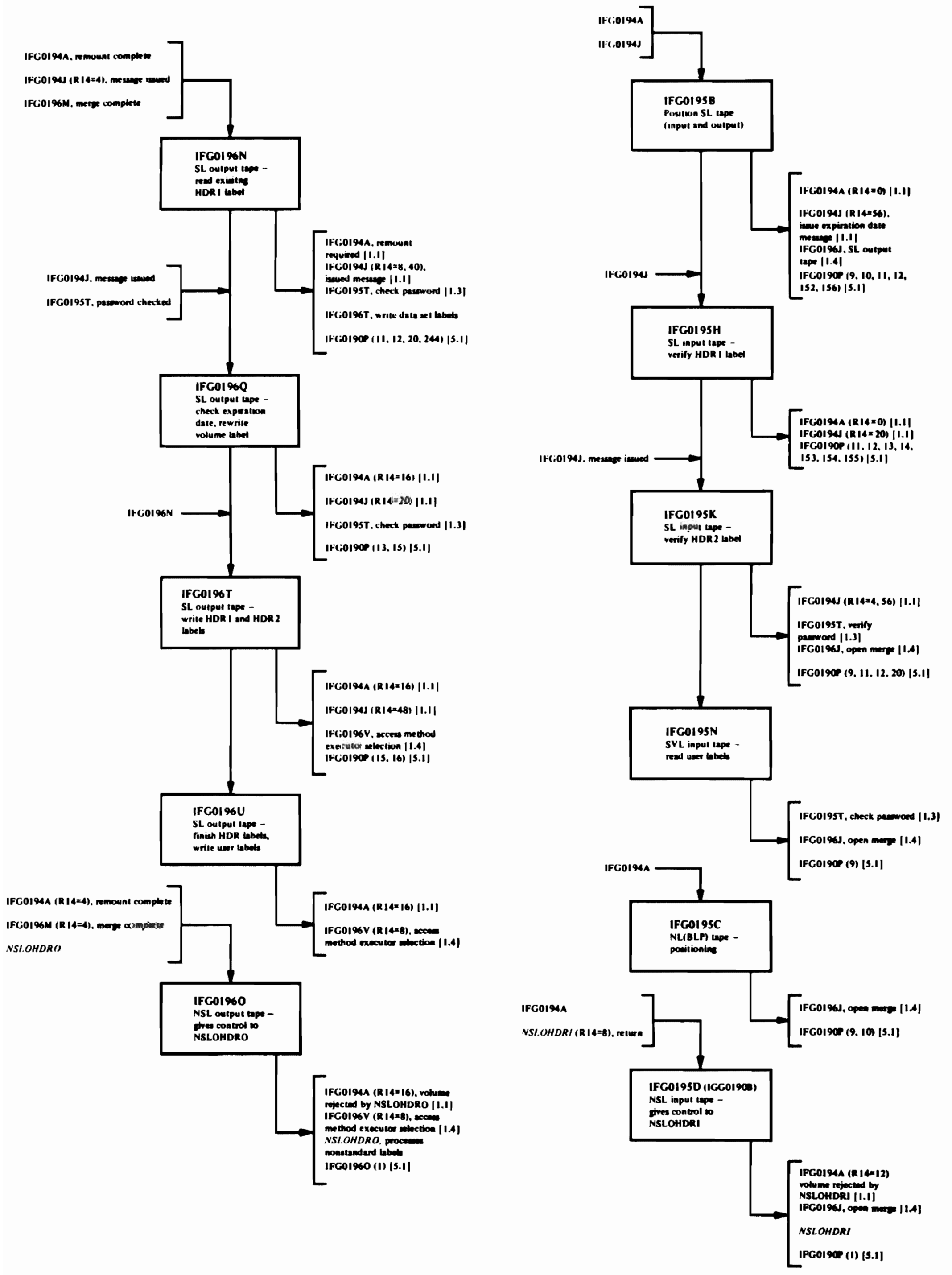

IFG0194A, remount complete
IFG0194J (R14=4), message issued
IFG0196M, merge complete
IFG0196N
SL output tape –
read exiitng
HDR1 label
IFG0194A, remount required [1.1]
IFG0194J (R14=8, 40), issued message [1.1]
IFG0195T, check password [1.3]
IFG0196T, write data set labels
IFG0190P (11, 12, 20, 244) [5.1]
IFG0194J, message issued
IFG0195T, password checked
IFG0196Q
SL output tape –
check expiration
date, rewrite
volume label
IFG0194A (R14=16) [1.1]
IFG0194J (R14=20) [1.1]
IFG0195T, check password [1.3]
IFG0190P (13, 15) [5.1]
IFG0196N
IFG0196T
SL output tape –
write HDR1 and HDR2
labels
IFG0194A (R14=16) [1.1]
IFG0194J (R14=48) [1.1]
IFG0196V, access method executor selection [1.4]
IFG0190P (15, 16) [5.1]
IFG0196U
SL output tape –
finish HDR labels,
write user labels
IFG0194A (R14=16) [1.1]
IFG0196V (R14=8), access method executor selection [1.4]
IFG0194A (R14=4), remount complete
IFG0196M (R14=4), merge complete
NSLOHDRO
IFG0196O
NSL output tape –
gives control to
NSLOHDRO
IFG0194A (R14=16), volume rejected by NSLOHDRO [1.1]
IFG0196V (R14=8), access method executor selection [1.4]
NSLOHDRO, processes nonstandard labels
IFG0196O (1) [5.1]
IFG0194A
IFG0194J
IFG0195B
Position SL tape
(input and output)
IFG0194A (R14=0) [1.1]
IFG0194J (R14=56), issue expiration date message [1.1]
IFG0196J, SL output tape [1.4]
IFG0190P (9, 10, 11, 12, 152, 156) [5.1]
IFG0194J
IFG0195H
SL input tape –
verify HDR1 label
IFG0194A (R14=0) [1.1]
IFG0194J (R14=20) [1.1]
IFG0190P (11, 12, 13, 14, 153, 154, 155) [5.1]
IFG0194J, message issued
IFG0195K
SL input tape –
verify HDR2 label
IFG0194J (R14=4, 56) [1.1]
IFG0195T, verify password [1.3]
IFG0196J, open merge [1.4]
IFG0190P (9, 11, 12, 20) [5.1]
IFG0195N
SVL input tape –
read user labels
IFG0195T, check password [1.3]
IFG0196J, open merge [1.4]
IFG0190P (9) [5.1]
IFG0194A
IFG0195C
NL(BLP) tape –
positioning
IFG0196J, open merge [1.4]
IFG0190P (9, 10) [5.1]
IFG0194A
NSLOHDRI (R14=8), return
IFG0195D (IGG0190B)
NSL input tape –
gives control to
NSLOHDRI
IFG0194A (R14=12) volume rejected by NSLOHDRI [1.1]
IFG0196J, open merge [1.4]
NSLOHDRI
IFG0190P (1) [5.1]

Chart 1.2. Open Tape

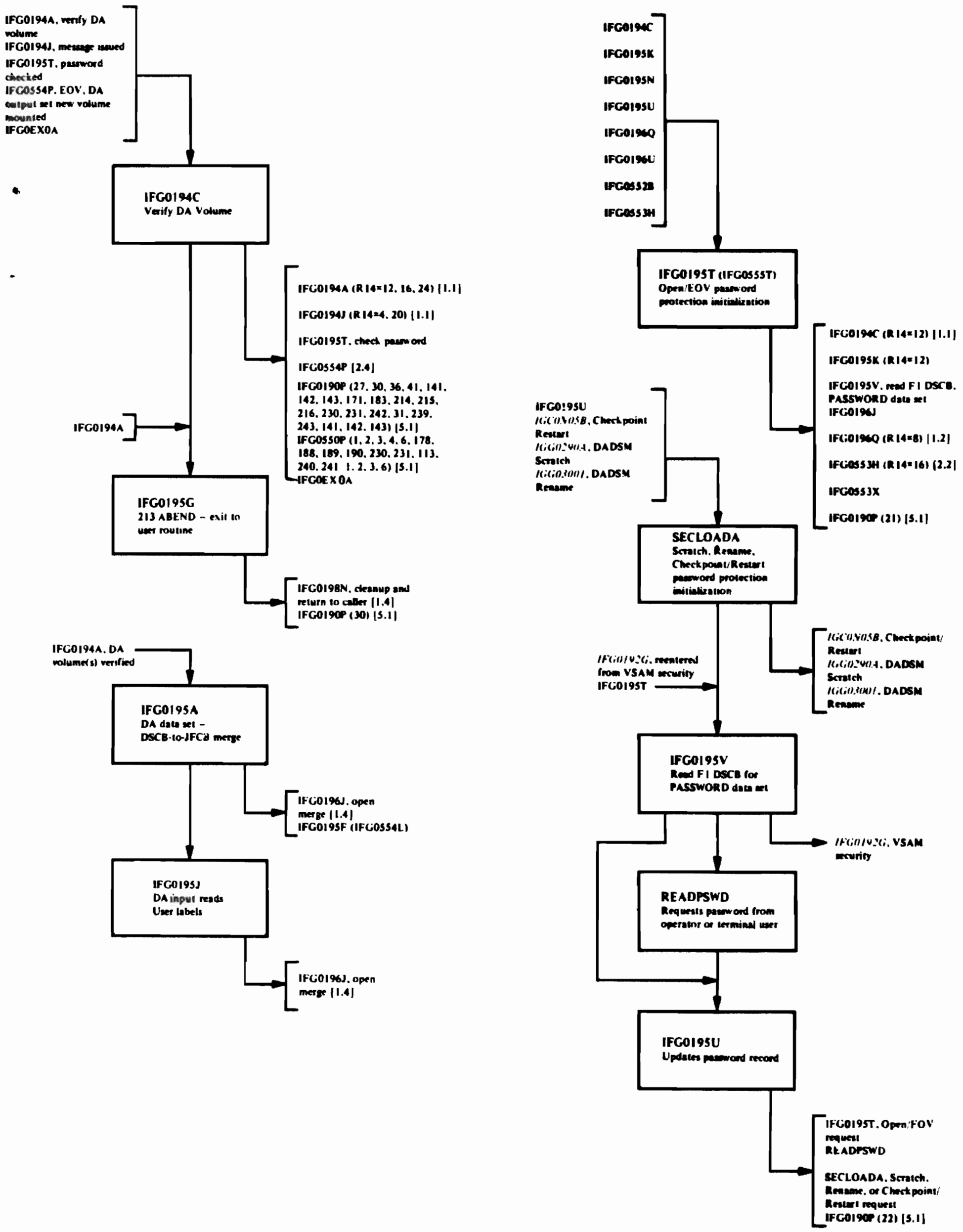

IFG0194A, verify DA volume
IFG0194J, message issued
IFG0195T, password checked
IFG0554P, EOV, DA output set new volume mounted
IFG0EX0A
IFG0194C
Verify DA Volume
IFG0194A (R14=12, 16, 24) [1.1]
IFG0194J (R14=4, 20) [1.1]
IFG0195T, check password
IFG0554P [2.4]
IFG0190P (27, 30, 36, 41, 141, 142, 143, 171, 183, 214, 215, 216, 230, 231, 242, 31, 239, 243, 141, 142, 143) [5.1]
IFG0550P (1, 2, 3, 4, 6, 178, 188, 189, 190, 230, 231, 113, 240, 241 1, 2, 3, 6) [5.1]
IFG0EX0A
IFG0194A
IFG0195G
213 ABEND – exit to user routine
IFG0198N, cleanup and return to caller [1.4]
IFG0190P (30) [5.1]
IFG0194A, DA volume(s) verified
IFG0195A
DA data set – DSCB-to-JFCB merge
IFG0196J, open merge [1.4]
IFG0195F (IFG0554L)
IFG0195J
DA input reads User labels
IFG0196J, open merge [1.4]
IFG0194C
IFG0195K
IFG0195N
IFG0195U
IFG0196Q
IFG0196U
IFG0552B
IFG0553H
IFG0195T (IFG0555T)
Open/EOV password protection initialization
IFG0194C (R14=12) [1.1]
IFG0195K (R14=12)
IFG0195V, read F1 DSCB, PASSWORD data set
IFG0196J
IFG0196Q (R14=8) [1.2]
IFG0553H (R14=16) [2.2]
IFG0553X
IFG0190P (21) [5.1]
IFG0195U
IGC0N05B, Checkpoint Restart
IGG0290A, DADSM Scratch
IGG0300I, DADSM Rename
SECLOADA
Scratch, Rename, Checkpoint/Restart password protection initialization
IGC0N05B, Checkpoint/Restart
IGG0290A, DADSM Scratch
IGG0300I, DADSM Rename
IFG0192G, reentered from VSAM security
IFG0195T
IFG0195V
Read F1 DSCB for PASSWORD data set
IFG0192G, VSAM security
READPSWD
Requests password from operator or terminal user
IFG0195U
Updates password record
IFG0195T, Open/FOV request
READPSWD
SECLOADA, Scratch, Rename, or Checkpoint/Restart request
IFG0190P (22) [5.1]

Chart 1.3. Open Direct Access and Open/EOV Password Protection

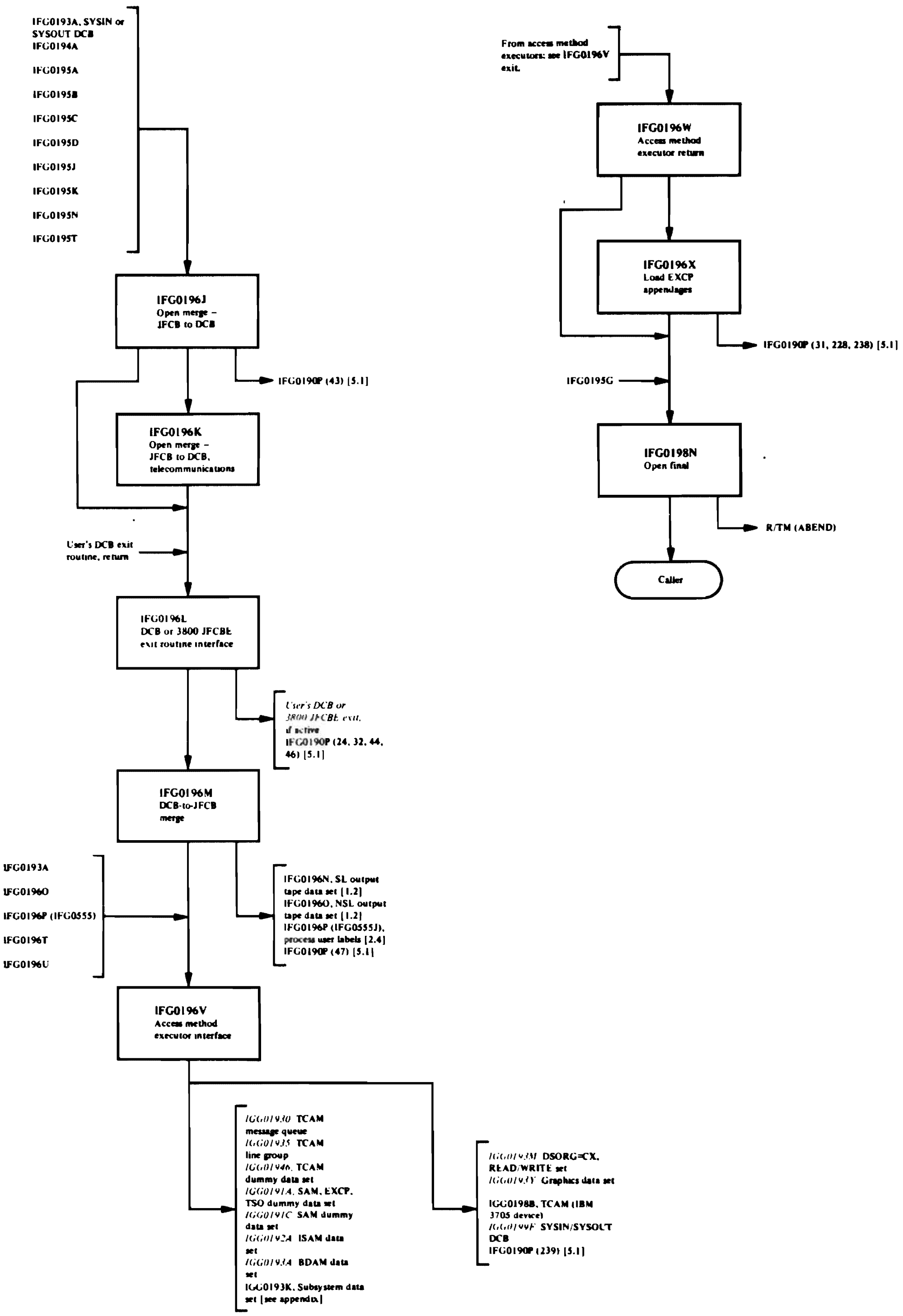

IFG0193A, SYSIN or SYSOUT DCB
IFG0194A
IFG0195A
IFG0195B
IFG0195C
IFG0195D
IFG0195J
IFG0195K
IFG0195N
IFG0195T
IFG0196J Open merge – JFCB to DCB
IFG0190P (43) [5.1]
IFG0196K Open merge – JFCB to DCB, telecommunications
User's DCB exit routine, return
IFG0196L DCB or 3800 JFCBE exit routine interface
User's DCB or 3800 JFCBE exit, if active
IFG0190P (24, 32, 44, 46) [5.1]
IFG0196M DCB-to-JFCB merge
IFG0193A
IFG0196O
IFG0196P (IFG0555)
IFG0196T
IFG0196U
IFG0196N, SL output tape data set [1.2]
IFG0196O, NSL output tape data set [1.2]
IFG0196P (IFG0555J), process user labels [2.4]
IFG0190P (47) [5.1]
IFG0196V Access method executor interface
IGG01930 TCAM message queue
IGG01935 TCAM line group
IGG01946, TCAM dummy data set
IGG0191A, SAM, EXCP, TSO dummy data set
IGG0191C SAM dummy data set
IGG0192A ISAM data set
IGG0193A BDAM data set
IGG0193K, Subsystem data set [see appendix]
IGG0193M DSORG=CX, READ/WRITE set
IGG0193Y Graphics data set
IGG0198B, TCAM (IBM 3705 device)
IGG0199F SYSIN/SYSOUT DCB
IFG0190P (239) [5.1]
From access method executors: see IFG0196V exit.
IFG0196W Access method executor return
IFG0196X Load EXCP appendages
IFG0190P (31, 228, 238) [5.1]
IFG0195G
IFG0198N Open final
R/TM (ABEND)
Caller

Chart 1.4. Open Merge and Open Final

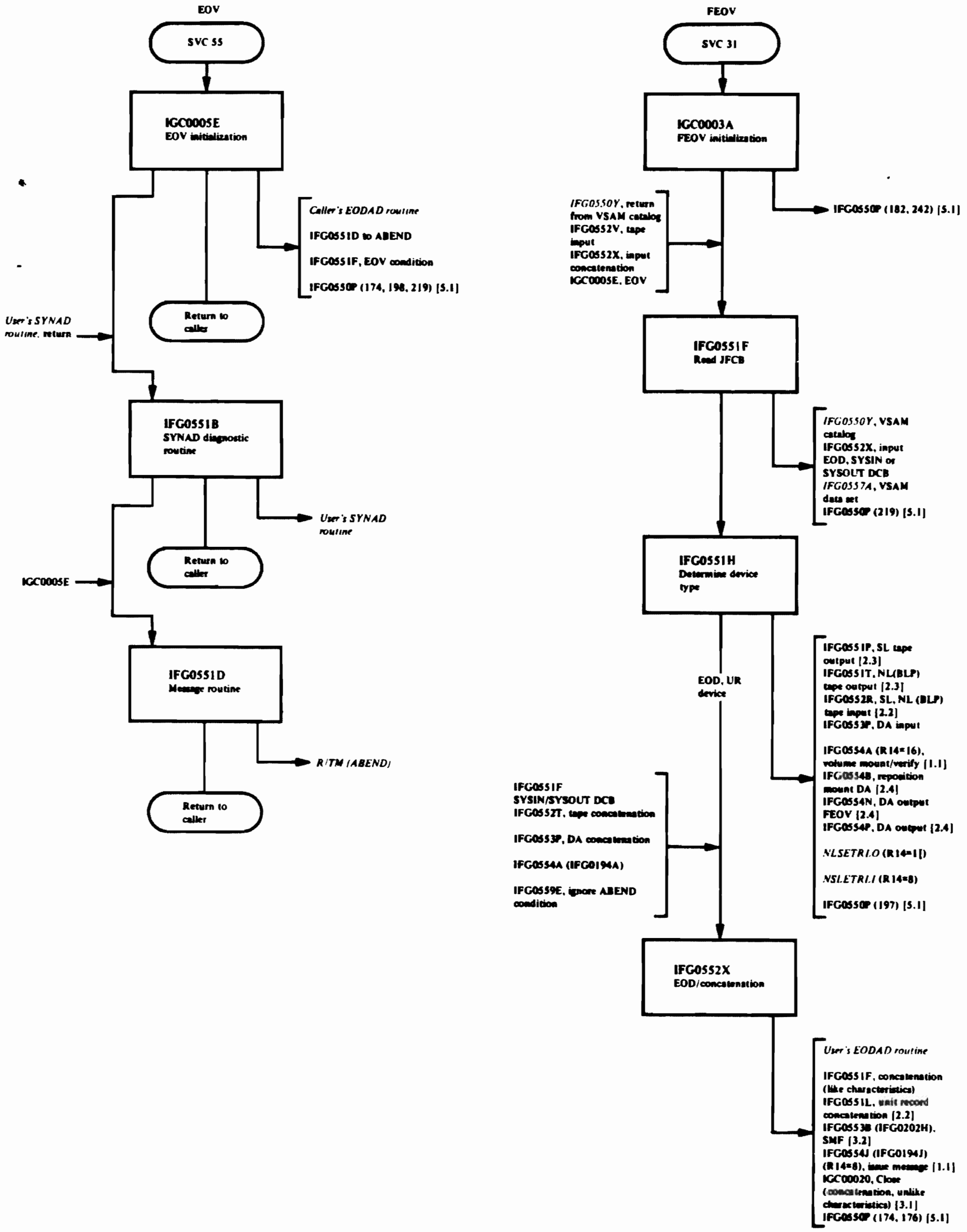

EOV
SVC 55
IGC0005E
EOV initialization
Caller's EODAD routine
IFG0551D to ABEND
IFG0551F, EOV condition
IFG0550P (174, 198, 219) [5.1]
Return to caller
User's SYNAD routine, return
IFG0551B
SYNAD diagnostic routine
User's SYNAD routine
Return to caller
IGC0005E
IFG0551D
Message routine
R/TM (ABEND)
Return to caller
FEOV
SVC 31
IGC0003A
FEOV initialization
IFG0550Y, return from VSAM catalog
IFG0552V, tape input
IFG0552X, input concatenation
IGC0005E, EOV
IFG0550P (182, 242) [5.1]
IFG0551F
Read JFCB
IFG0550Y, VSAM catalog
IFG0552X, input EOD, SYSIN or SYSOUT DCB
IFG0557A, VSAM data set
IFG0550P (219) [5.1]
IFG0551H
Determine device type
EOD, UR device
IFG0551P, SL tape output [2.3]
IFG0551T, NL(BLP) tape output [2.3]
IFG0552R, SL, NL (BLP) tape input [2.2]
IFG0553P, DA input
IFG0554A (R14=16), volume mount/verify [1.1]
IFG0554B, reposition mount DA [2.4]
IFG0554N, DA output FEOV [2.4]
IFG0554P, DA output [2.4]
NLSETRLO (R14=1[)
NSLETRLI (R14=8)
IFG0550P (197) [5.1]
IFG0551F SYSIN/SYSOUT DCB
IFG0552T, tape concatenation
IFG0553P, DA concatenation
IFG0554A (IFG0194A)
IFG0559E, ignore ABEND condition
IFG0552X
EOD/concatenation
User's EODAD routine
IFG0551F, concatenation (like characteristics)
IFG0551L, unit record concatenation [2.2]
IFG0553B (IFG0202H), SMF [3.2]
IFG0554J (IFG0194J) (R14=8), issue message [1.1]
IGC00020, Close (concatenation, unlike characteristics) [3.1]
IFG0550P (174, 176) [5.1]

Chart 2.1. EOV Initialization

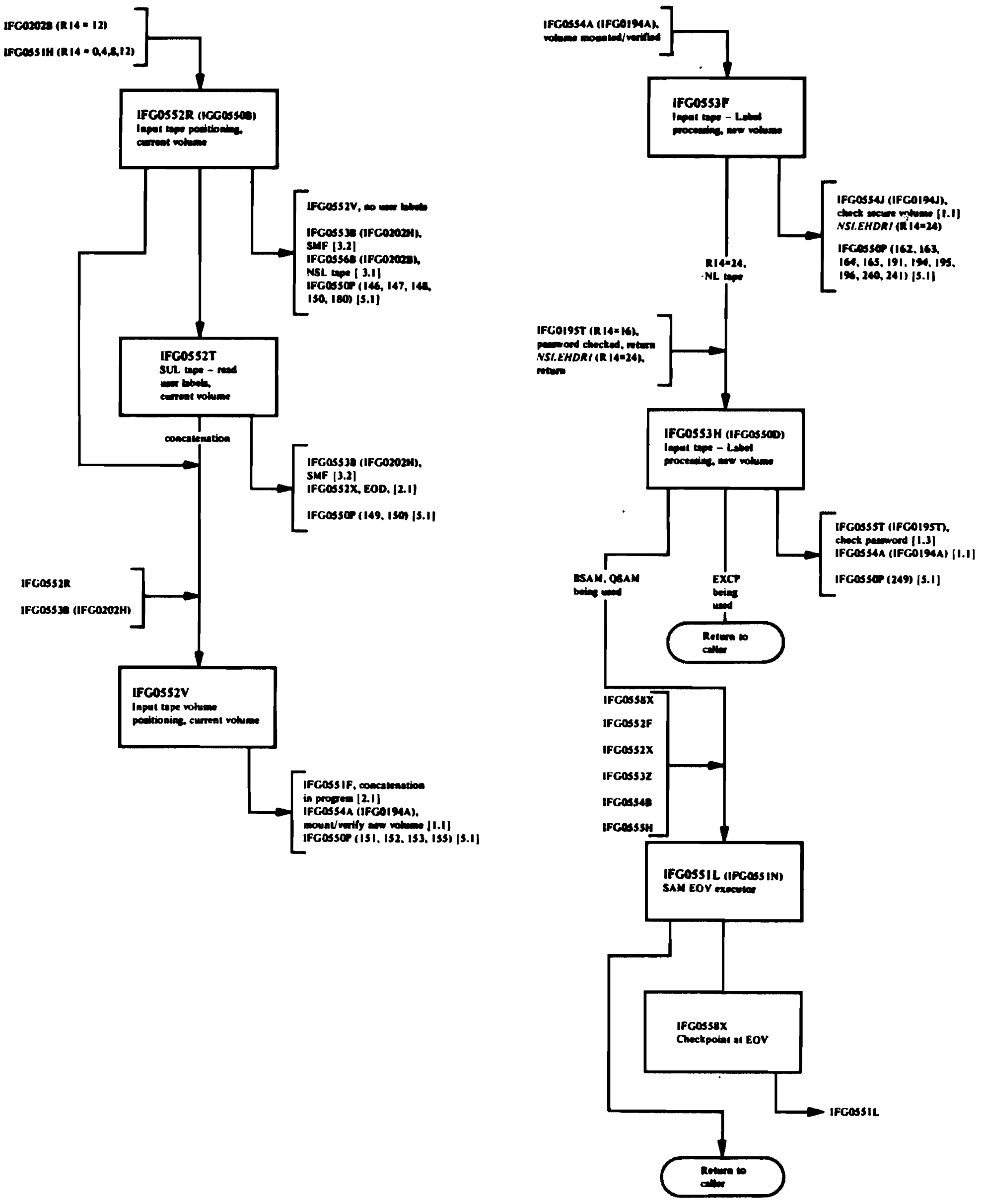

IFG0202B (R14 = 12)
IFG0551H (R14 = 0,4,8,12)
IFG0552R (IGG0550B)
Input tape positioning, current volume
IFG0552V, no user labels
IFG0553B (IFG0202H), SMF [3.2]
IFG0556B (IFG0202B), NSL tape [3.1]
IFG0550P (146, 147, 148, 150, 180) [5.1]
IFG0552T
SUL tape – read user labels, current volume
concatenation
IFG0553B (IFG0202H), SMF [3.2]
IFG0552X, EOD, [2.1]
IFG0550P (149, 150) [5.1]
IFG0552R
IFG0553B (IFG0202H)
IFG0552V
Input tape volume positioning, current volume
IFG0551F, concatenation in progress [2.1]
IFG0554A (IFG0194A), mount/verify new volume [1.1]
IFG0550P (151, 152, 153, 155) [5.1]
IFG0554A (IFG0194A), volume mounted/verified
IFG0553F
Input tape – Label processing, new volume
IFG0554J (IFG0194J), check secure volume [1.1]
NSLEHDRI (R14=24)
IFG0550P (162, 163, 164, 165, 191, 194, 195, 196, 240, 241) [5.1]
R14=24, NL tape
IFG0195T (R14=16), password checked, return
NSLEHDRI (R14=24), return
IFG0553H (IFG0550D)
Input tape – Label processing, new volume
IFG0555T (IFG0195T), check password [1.3]
IFG0554A (IFG0194A) [1.1]
IFG0550P (249) [5.1]
BSAM, QSAM being used
EXCP being used
Return to caller
IFG0558X
IFG0552F
IFG0552X
IFG0553Z
IFG0554B
IFG0555H
IFG0551L (IFG0551N)
SAM EOV executor
IFG0558X
Checkpoint at EOV
IFG0551L
Return to caller

Chart 2.2. EOV Tape Input

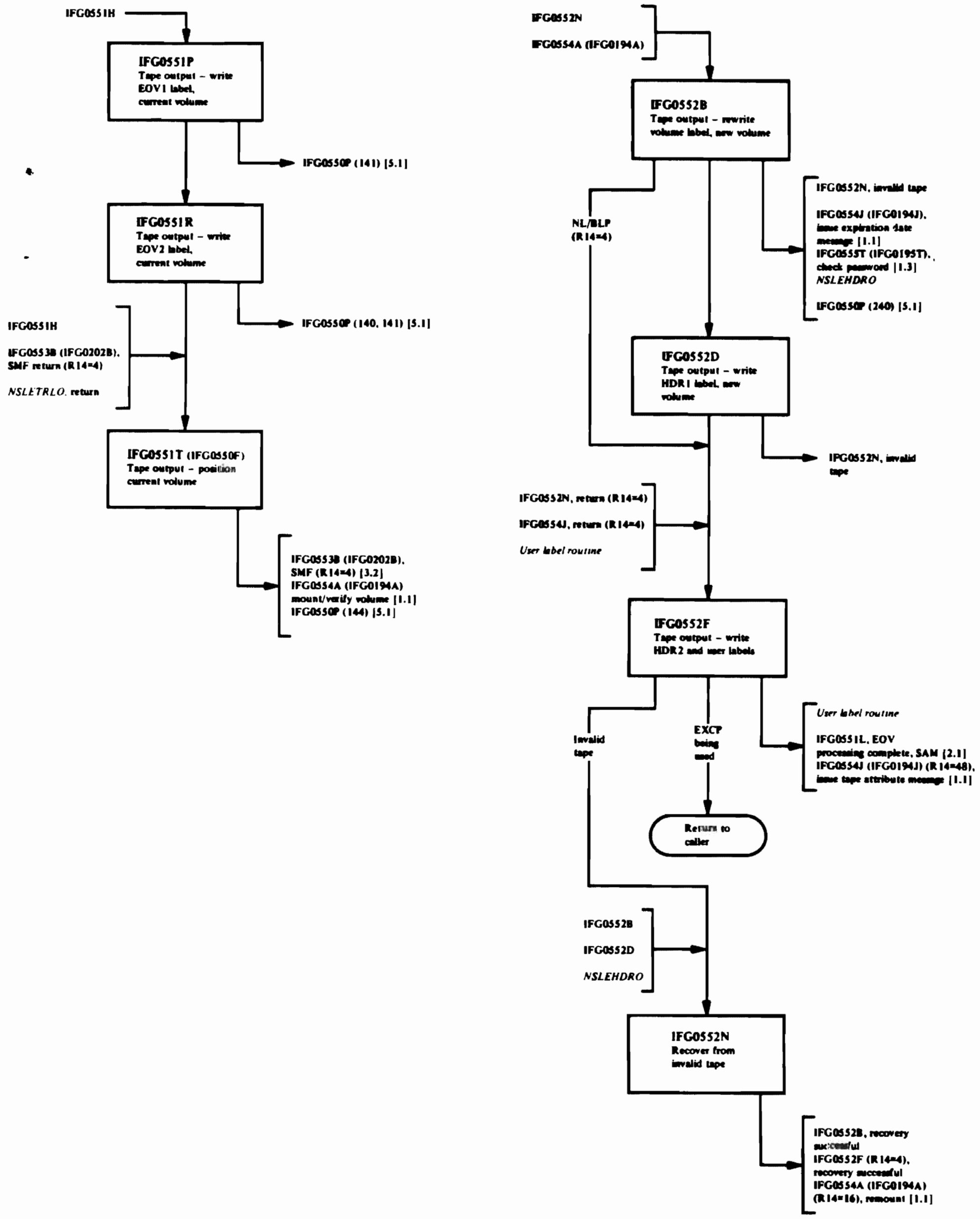


Chart 2.3. EOV Tape Output

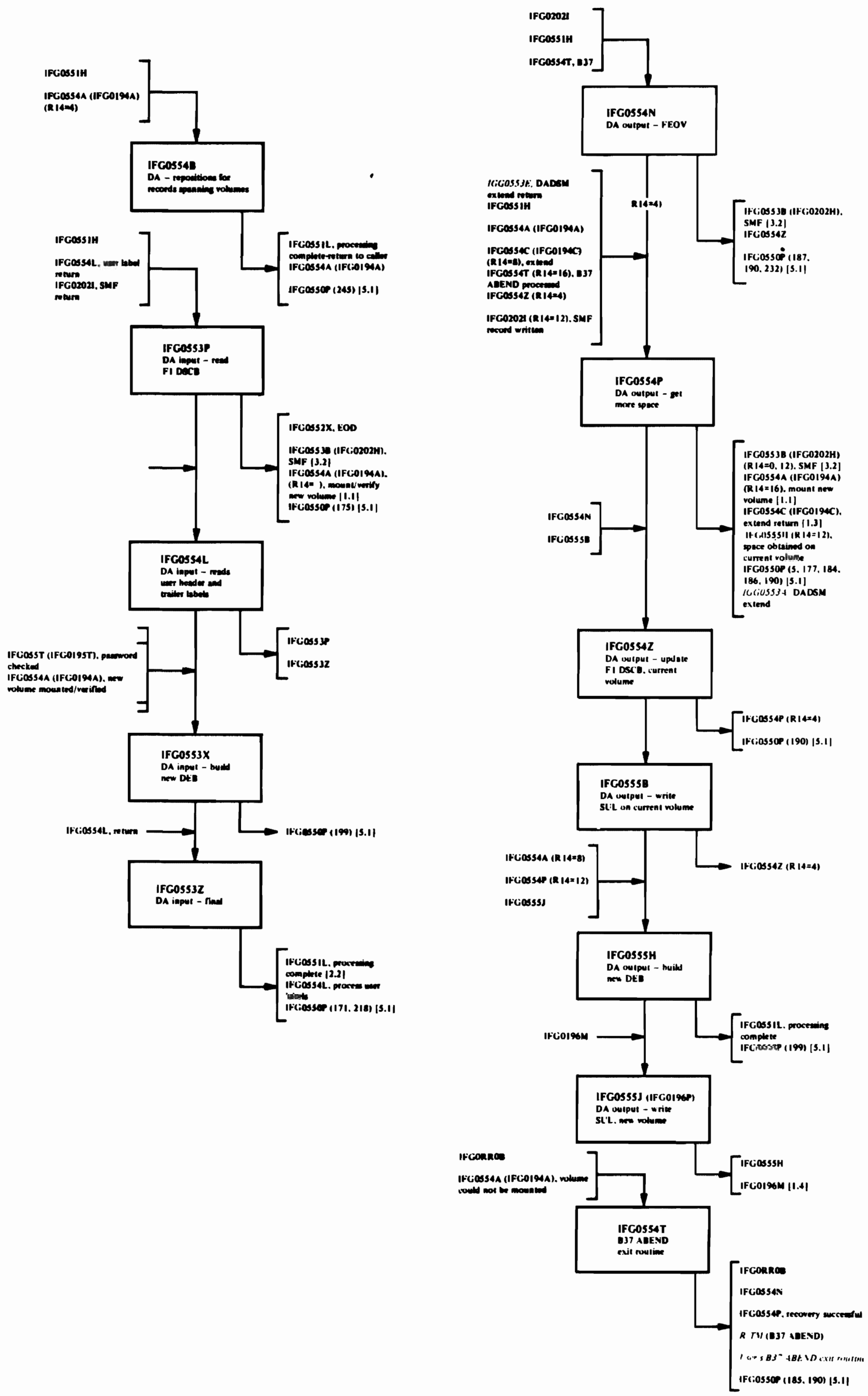

IFG0551H
IFG0554A (IFG0194A) (R14=4)
IFG0554B
DA – repositions for records spanning volumes
IFG0551L, processing complete-return to caller
IFG0554A (IFG0194A)
IFG0550P (245) [5.1]
IFG0551H
IFG0554L, user label return
IFG0202I, SMF return
IFG0553P
DA input – read F1 DSCB
IFG0552X, EOD
IFG0553B (IFG0202H), SMF [3.2]
IFG0554A (IFG0194A), (R14=), mount/verify new volume [1.1]
IFG0550P (175) [5.1]
IFG0554L
DA input – reads user header and trailer labels
IFG055T (IFG0195T), password checked
IFG0554A (IFG0194A), new volume mounted/verified
IFG0553P
IFG0553Z
IFG0553X
DA input – build new DEB
IFG0554L, return
IFG0550P (199) [5.1]
IFG0553Z
DA input – final
IFG0551L, processing complete [2.2]
IFG0554L, process user labels
IFG0550P (171, 218) [5.1]
IFG0202I
IFG0551H
IFG0554T, B37
IFG0554N
DA output – FEOV
IGG0553E, DADSM extend return
IFG0551H
IFG0554A (IFG0194A)
IFG0554C (IFG0194C) (R14=8), extend
IFG0554T (R14=16), B37 ABEND processed
IFG0554Z (R14=4)
IFG0202I (R14=12), SMF record written
R14=4)
IFG0553B (IFG0202H), SMF [3.2]
IFG0554Z
IFG0550P (187, 190, 232) [5.1]
IFG0554P
DA output – get more space
IFG0554N
IFG0555B
IFG0553B (IFG0202H) (R14=0, 12), SMF [3.2]
IFG0554A (IFG0194A) (R14=16), mount new volume [1.1]
IFG0554C (IFG0194C), extend return [1.3]
IFG0555H (R14=12), space obtained on current volume
IFG0550P (5, 177, 184, 186, 190) [5.1]
IGG0553A DADSM extend
IFG0554Z
DA output – update F1 DSCB, current volume
IFG0554P (R14=4)
IFG0550P (190) [5.1]
IFG0555B
DA output – write SUL on current volume
IFG0554A (R14=8)
IFG0554P (R14=12)
IFG0555J
IFG0554Z (R14=4)
IFG0555H
DA output – build new DEB
IFG0196M
IFG0551L, processing complete
IFG0550P (199) [5.1]
IFG0555J (IFG0196P)
DA output – write SUL, new volume
IFG0RR0B
IFG0554A (IFG0194A), volume could not be mounted
IFG0555H
IFG0196M [1.4]
IFG0554T
B37 ABEND exit routine
IFG0RR0B
IFG0554N
IFG0554P, recovery successful
RTM (B37 ABEND)
User's B37 ABEND exit routine
IFG0550P (185, 190) [5.1]

Chart 2.4. EOV Direct Access, Input and Output

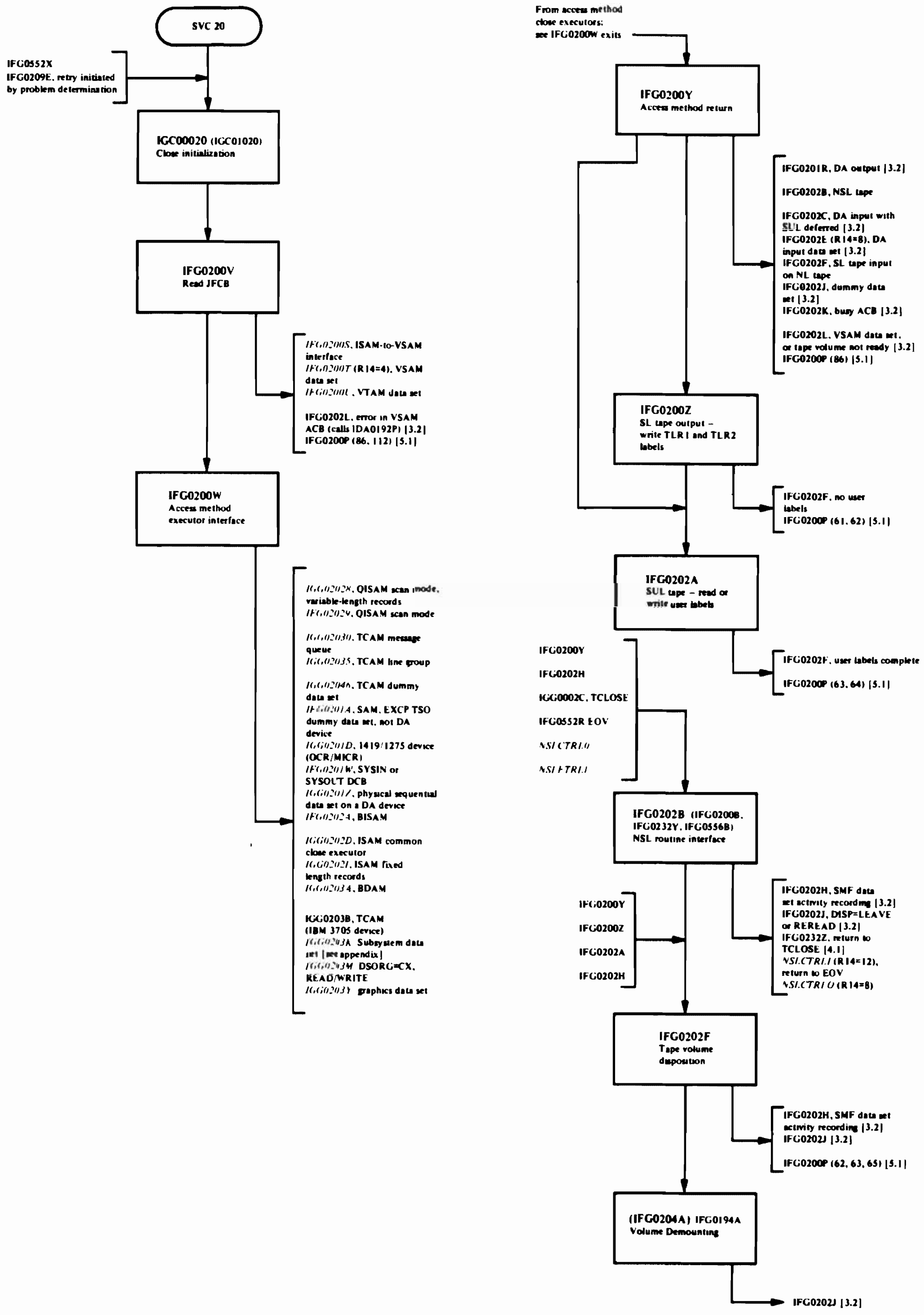

SVC 20
IFG0552X
IFG0209E, retry initiated by problem determination
IGC00020 (IGC01020) Close initialization
IFG0200V Read JFCB
IFG0200S, ISAM-to-VSAM interface
IFG0200T (R14=4), VSAM data set
IFG0200U, VTAM data set
IFG0202L, error in VSAM ACB (calls IDA0192P) [3.2]
IFG0200P (86, 112) [5.1]
IFG0200W Access method executor interface
IGG02028, QISAM scan mode, variable-length records
IFG02029, QISAM scan mode
IGG02030, TCAM message queue
IGG02035, TCAM line group
IGG02046, TCAM dummy data set
IFG0201A, SAM, EXCP TSO dummy data set, not DA device
IGG0201D, 1419/1275 device (OCR/MICR)
IFG0201W, SYSIN or SYSOUT DCB
IGG0201Z, physical sequential data set on a DA device
IFG02024, BISAM
IGG0202D, ISAM common close executor
IGG0202I, ISAM fixed length records
IGG02034, BDAM
IGG0203B, TCAM (IBM 3705 device)
IGG0203A Subsystem data set [see appendix]
IGG0203M DSORG=CX, READ/WRITE
IGG0203Y graphics data set
From access method close executors: see IFG0200W exits
IFG0200Y Access method return
IFG0201R, DA output [3.2]
IFG0202B, NSL tape
IFG0202C, DA input with SUL deferred [3.2]
IFG0202E (R14=8), DA input data set [3.2]
IFG0202F, SL tape input on NL tape
IFG0202J, dummy data set [3.2]
IFG0202K, busy ACB [3.2]
IFG0202L, VSAM data set, or tape volume not ready [3.2]
IFG0200P (86) [5.1]
IFG0200Z SL tape output – write TLR1 and TLR2 labels
IFG0202F, no user labels
IFG0200P (61, 62) [5.1]
IFG0202A SUL tape – read or write user labels
IFG0202F, user labels complete
IFG0200P (63, 64) [5.1]
IFG0200Y
IFG0202H
IGG0002C, TCLOSE
IFG0552R EOV
NSLCTRLO
NSLETRLI
IFG0202B (IFG0200B, IFG0232Y, IFG0556B) NSL routine interface
IFG0200Y
IFG0200Z
IFG0202A
IFG0202H
IFG0202H, SMF data set activity recording [3.2]
IFG0202J, DISP=LEAVE or REREAD [3.2]
IFG0232Z, return to TCLOSE [4.1]
NSLCTRLI (R14=12), return to EOV
NSLCTRLO (R14=8)
IFG0202F Tape volume disposition
IFG0202H, SMF data set activity recording [3.2]
IFG0202J [3.2]
IFG0200P (62, 63, 65) [5.1]
(IFG0204A) IFG0194A Volume Demounting
IFG0202J [3.2]

Chart 3.1. Close Initialization and Close Tape

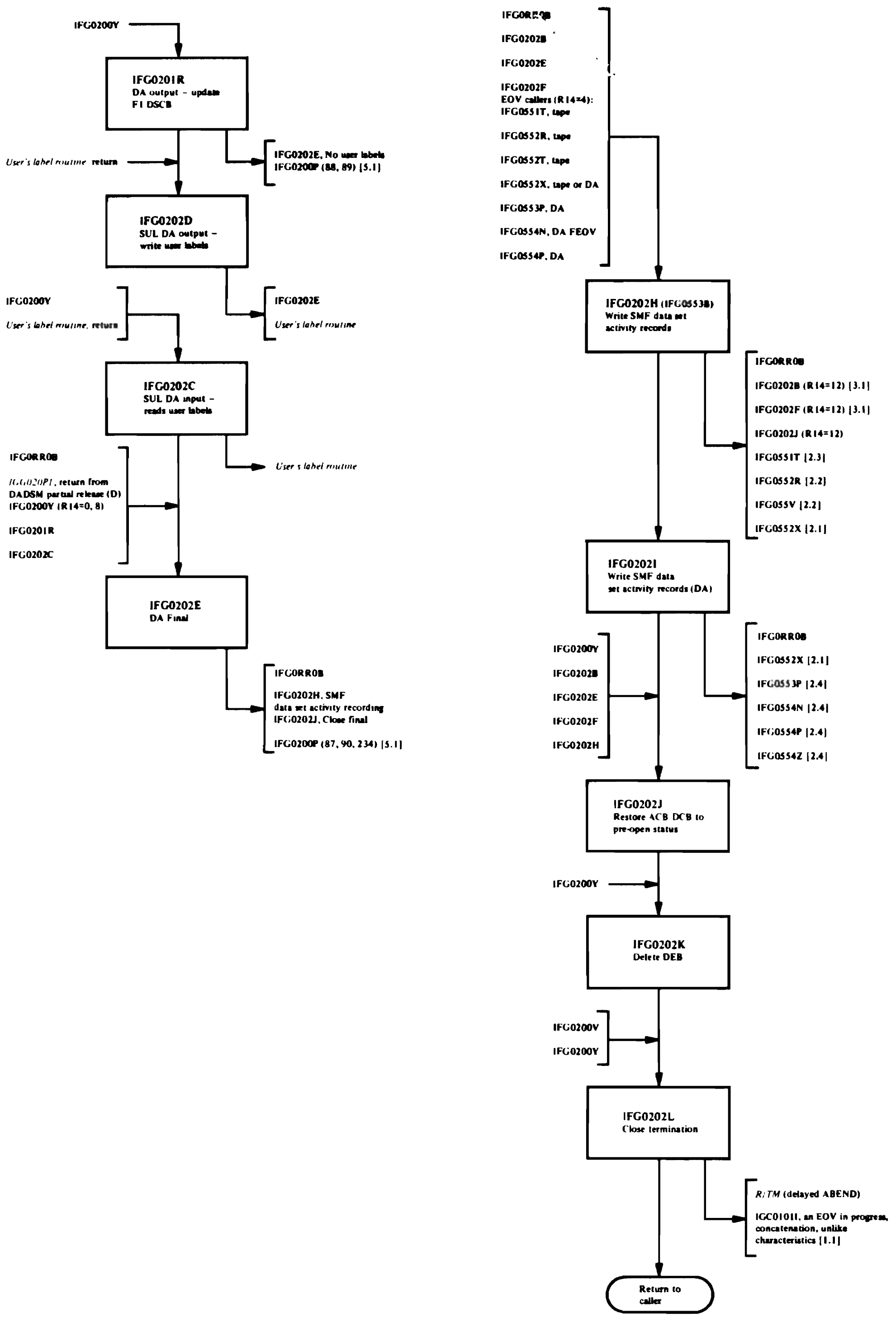


Chart 3.2. Close Direct Access, Close SMF, and Close Final

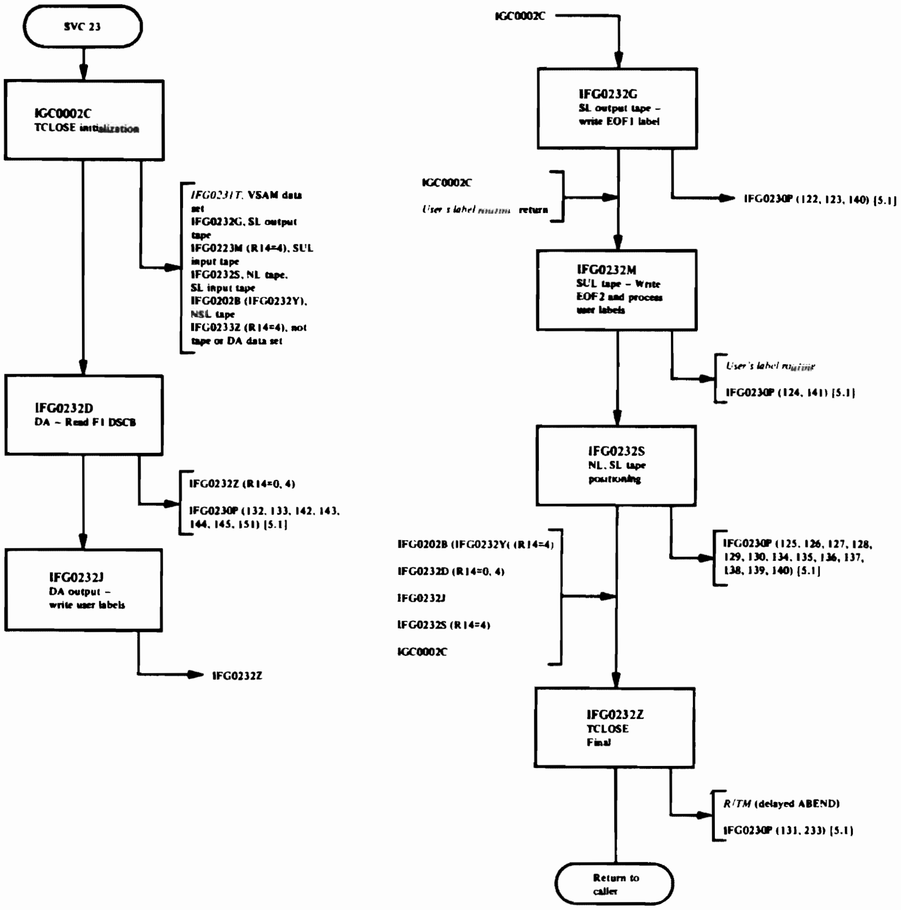

SVC 23
IGC0002C
TCLOSE initialization
IFG0231T, VSAM data set
IFG0232G, SL output tape
IFG0223M (R14=4), SUL input tape
IFG0232S, NL tape, SL input tape
IFG0202B (IFG0232Y), NSL tape
IFG02332 (R14=4), not tape or DA data set
IFG0232D
DA – Read F1 DSCB
IFG0232Z (R14=0, 4)
IFG0230P (132, 133, 142, 143, 144, 145, 151) [5.1]
IFG0232J
DA output – write user labels
IFG0232Z
IGC0002C
IFG0232G
SL output tape – write EOF1 label
IGC0002C
User's label routine return
IFG0230P (122, 123, 140) [5.1]
IFG0232M
SUL tape – Write EOF2 and process user labels
User's label routine
IFG0230P (124, 141) [5.1]
IFG0232S
NL, SL tape positioning
IFG0230P (125, 126, 127, 128, 129, 130, 134, 135, 136, 137, 138, 139, 140) [5.1]
IFG0202B (IFG0232Y((R14=4)
IFG0232D (R14=0, 4)
IFG0232J
IFG0232S (R14=4)
IGC0002C
IFG0232Z
TCLOSE
Final
R/TM (delayed ABEND)
IFG0230P (131, 233) [5.1]
Return to caller

Chart 4.1. TCLOSE

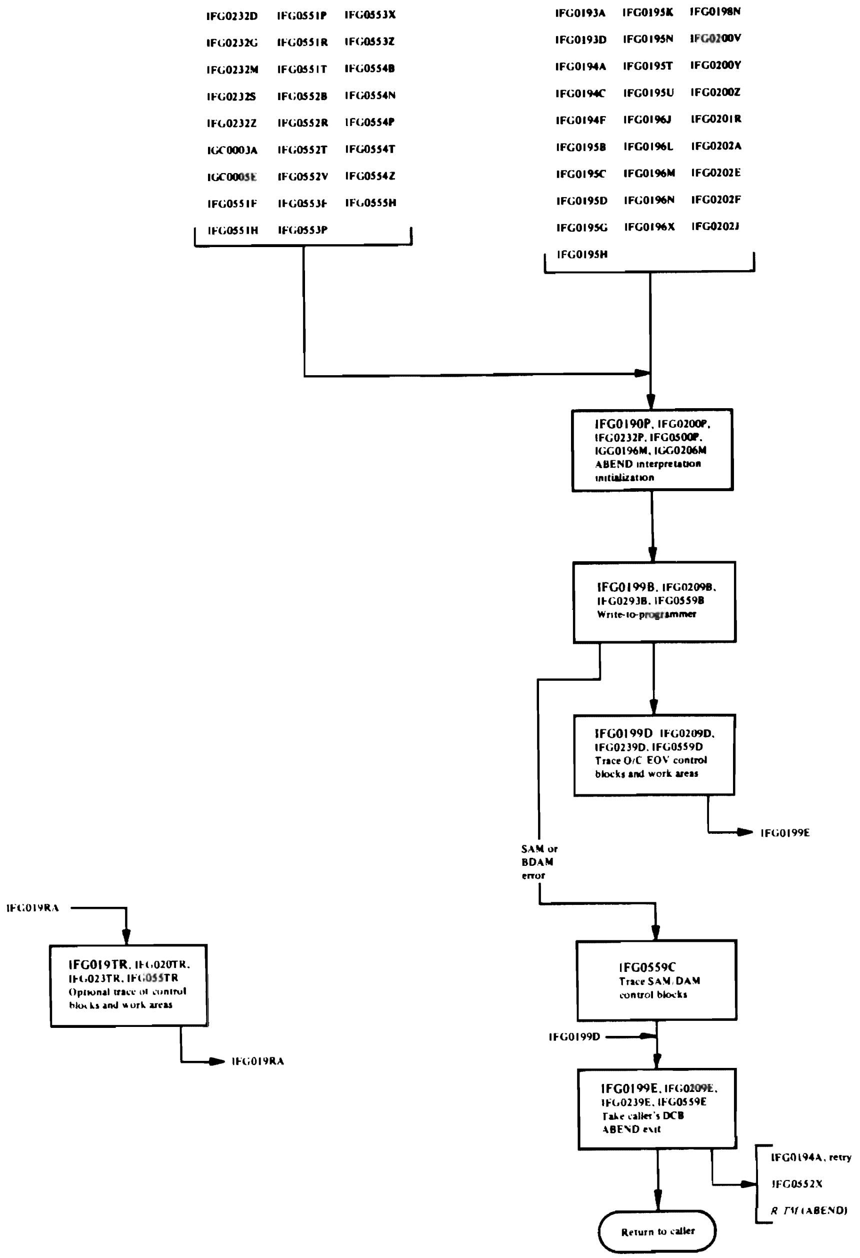

IFG0232D IFG0551P IFG0553X
IFG0232G IFG0551R IFG0553Z
IFG0232M IFG0551T IFG0554B
IFG0232S IFG0552B IFG0554N
IFG0232Z IFG0552R IFG0554P
IGC0003A IFG0552T IFG0554T
IGC0005E IFG0552V IFG0554Z
IFG0551F IFG0553F IFG0555H
IFG0551H IFG0553P
IFG0193A IFG0195K IFG0198N
IFG0193D IFG0195N IFG0200V
IFG0194A IFG0195T IFG0200Y
IFG0194C IFG0195U IFG0200Z
IFG0194F IFG0196J IFG0201R
IFG0195B IFG0196L IFG0202A
IFG0195C IFG0196M IFG0202E
IFG0195D IFG0196N IFG0202F
IFG0195G IFG0196X IFG0202J
IFG0195H
IFG0190P, IFG0200P, IFG0232P, IFG0500P, IGG0196M, IGG0206M ABEND interpretation initialization
IFG0199B, IFG0209B, IFG0293B, IFG0559B Write-to-programmer
IFG0199D IFG0209D, IFG0239D, IFG0559D Trace O/C EOV control blocks and work areas
IFG0199E
SAM or BDAM error
IFG019RA
IFG019TR, IFG020TR, IFG023TR, IFG055TR Optional trace of control blocks and work areas
IFG019RA
IFG0559C Trace SAM/DAM control blocks
IFG0199D
IFG0199E, IFG0209E, IFG0239E, IFG0559E Take caller's DCB ABEND exit
IFG0194A, retry
IFG0552X
R TM (ABEND)
Return to caller

Chart 5.1. ABEND Interpretation and Optional Module Trace

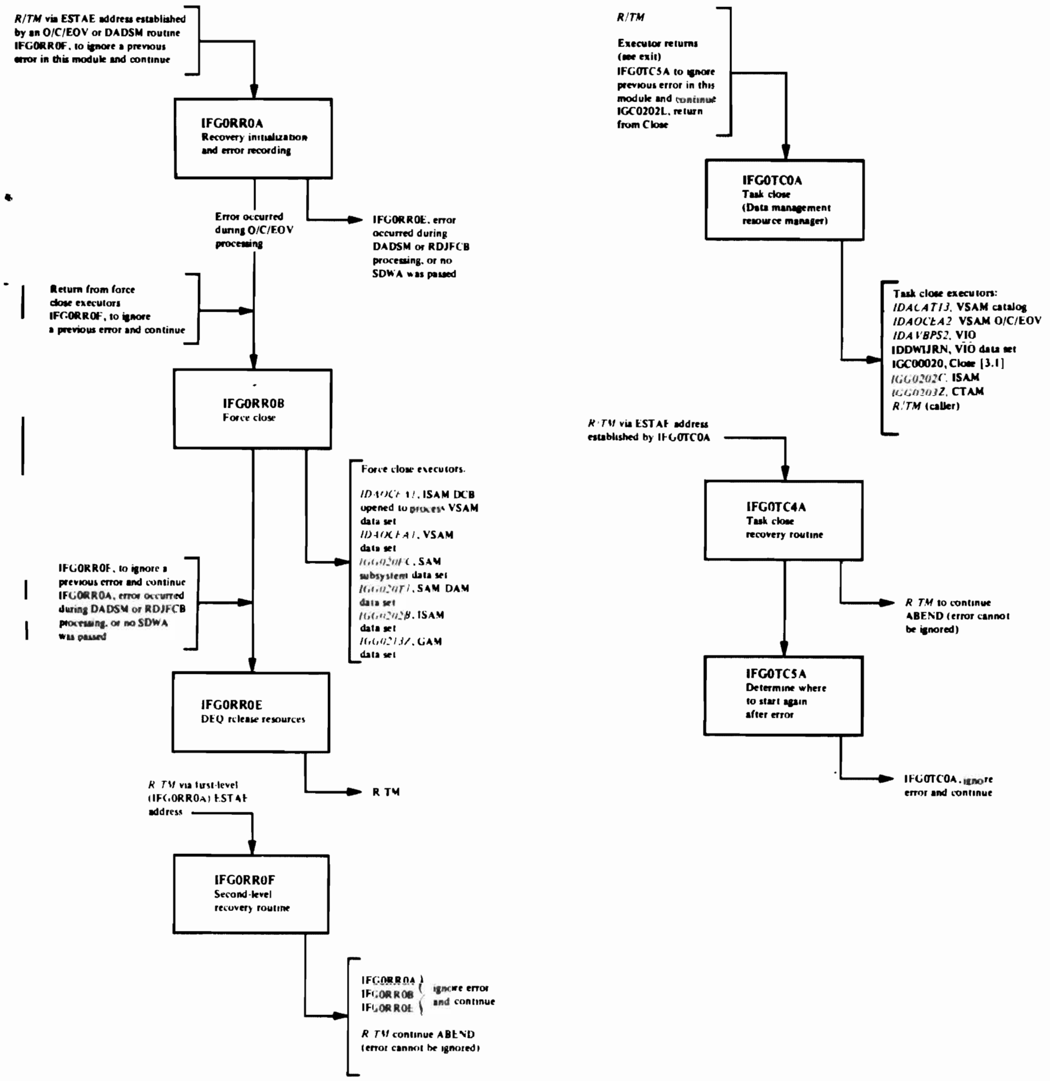

R/TM via ESTAE address established by an O/C/EOV or DADSM routine
IFG0RR0F, to ignore a previous error in this module and continue
IFG0RR0A
Recovery initialization and error recording
Error occurred during O/C/EOV processing
IFG0RR0E, error occurred during DADSM or RDJFCB processing, or no SDWA was passed
Return from force close executors
IFG0RR0F, to ignore a previous error and continue
IFG0RR0B
Force close
Force close executors.
IDAOCEA1, ISAM DCB opened to process VSAM data set
IDAOCEA1, VSAM data set
IGG020FC, SAM subsystem data set
IGG020T1, SAM DAM data set
IGG0202B, ISAM data set
IGG0213Z, GAM data set
IFG0RR0F, to ignore a previous error and continue
IFG0RR0A, error occurred during DADSM or RDJFCB processing, or no SDWA was passed
IFG0RR0E
DEQ release resources
R TM
R TM via first-level (IFG0RR0A) ESTAE address
IFG0RR0F
Second-level recovery routine
IFG0RR0A
IFG0RR0B
IFG0RR0E
ignore error and continue
R TM continue ABEND (error cannot be ignored)
R/TM
Executor returns (see exit)
IFG0TC5A to ignore previous error in this module and continue
IGC0202L, return from Close
IFG0TC0A
Task close
(Data management resource manager)
Task close executors:
IDACAT13, VSAM catalog
IDAOCEA2 VSAM O/C/EOV
IDAVBPS2, VIO
IDDWIJRN, VIO data set
IGC00020, Close [3.1]
IGG0202C, ISAM
IGG0203Z, CTAM
R/TM (caller)
R/TM via ESTAE address established by IFG0TC0A
IFG0TC4A
Task close recovery routine
R TM to continue ABEND (error cannot be ignored)
IFG0TC5A
Determine where to start again after error
IFG0TC0A, ignore error and continue

Chart 5.2. Recovery and Termination

DIRECTORY

This directory lists the object module names (which in all cases, are the same as the microfiche name) of the O/C/EOV modules in alphameric order. The object module names are the same as the control section (CSECT) names of the O/C/EOV modules. A table showing the object modules included in each load module follows the directory.

The column headings and their meanings are:

Microfiche or CSECT name: This is the name that appears on the microfiche card for each CSECT.

Alias: If the control section has an alias, the name will appear in this column. Note also that all aliases appear in the **Microfiche or CSECT name** column, followed by the real name of the CSECT.

M.O. Diagram: This column refers to the method of operation (HIPO) diagram or diagrams that make reference to the module; all of these diagrams appear in Section 2, "Method of Operation."

Module Flowchart: This column contains the chart number of the module flowchart on which the CSECT can be found; all of these charts appear in Section 3, "Program Organization."

Function: This column provides a brief description of the function provided by each module.

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
EMODVOL1	see IFG0553C			
IFGDEBCK		5.3	—	Validate DEB
IFGDEBVR	see IFGDEBCK			
IFG0EX0A		1.2, 2.6, 2.7	1.3	Installation format-1 DSCB-not-found exit routine
IFG0RR0A		6.2	5.2	Recovery routine initialization and error recording
IFG0RR0B		6.3	5.2	Force Close
IFG0RR0E		6.4	5.2	Recovery routine resource clean-up and final processing
IFG0RR0F		—	5.2	Second-level recovery routine
IFG0TC0A	IFG0TC4A	6.5	5.2	Task Close (Data Management Resource Manager)
IFG019RA		5.1, 5.2	—	O/C/EOV-DADSM service routine
IFG019TR	IFG020TR IFG023TR IFG055TR	5.1	5.1	Optional trace routine
IFG0190P		6.1	5.1	Open ABEND interpretation, initialization
IFG0193A		1.1	1.1	Open initialization

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
IFG0193C	OMODVOL1	1.4	1.1	Open, modify volume 1 label, tape label conflict
IFG0193D	IGG0190A IGG0550P IFG0553D	1.4, 2.5	1.1	Open/EOV destroy standard tape label
IFG0193E	IFG0553E	1.4, 2.5	1.1	Open/EOV create standard tape label
IFG0194A	IFG0554A IFG0204A	1.2, 1.4, 2.3, 2.5, 2.6, 2.7	1.1, 3.1	Open/EOV unit and volume selection
IFG0194C	IFG0554C	1.2, 2.6, 2.7	1.3	Open/EOV DA volume verification
IFG0194F	IFG0554F	1.4, 2.3, 2.5	1.1	Open/EOV tape volume verification
IFG0194J	IFG0204J IFG0554J	1.2, 1.4, 1.9, 2.3, 2.5, 2.6, 2.7	1.1	O/C/EOV message routine
IFG0195A		1.3	1.3	Open DA, DSCB-to-JFCB merge
IFG0195B		1.5	1.2	Open, position SL tape
IFG0195C		1.6	1.2	Open, position NL (BLP) tape
IFG0195D	IGG0190B	1.6	1.2	Open, NSL input tape, gives control to NSLOHDRI
IFG0195F	see IFG0554L			
IFG0195G		1.2	1.3	Open, DA 213 ABEND, exit to user
IFG0195H		1.5	1.2	Open, SL input tape, verify HDR1 label
IFG0195J		1.3	1.3	Open, DA input, reads F1 DSCB
IFG0195K		1.5	1.2	Open, SL input tape, verify HDR2 label
IFG0195N		1.5	1.2	Open, SL input tape, read user labels
IFG0195T	IFG0555T	1.7	1.3	Open/EOV password protection, initialization
IFG0195U		1.7	1.3	Password protection, update password record
IFG0195V		1.7	1.3	Password protection, read F1 DSCB for PASSWORD data set
IFG0196J		1.8	1.4	Open merge, JFCB to DCB
IFG0196K		1.8	1.4	Open merge, JFCB to DCB (telecommunications)
IFG0196L		1.8	1.4	Open merge, calls user's DCB exit routine

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
IFG0196M		1.8	1.4	Open merge, DCB to JFCB
IFG0196N		1.9	1.2	Open, SL output tape, read existing HDR1 label
IFG0196O	IGG0190R	1.6	1.2	Open, NSL output tape, gives control to NSLOHDRO
IFG0196P	see IFG0555J			
IFG0196Q		1.9	1.2	Open, SL output tape, check expiration date, rewrite volume label
IFG0196T		1.9	1.2	Open, SL output tape, write HDR1 and HDR2 labels
IFG0196U		1.9	1.2	Open, SL output tape, finish HDR labels, write user labels
IFG0196V		1.10	1.4	Open, gives control to access method open executors
IFG0196W	IGG0190S	1.10	1.4	Open access method executor return, JFCB-to-DSCB merge
IFG0196X		1.10	1.4	Open, Load EXCP appendages, journal JFCB
IFG0198N		1.10	1.4	Open final
IFG0199B	IFG0209B IFG0239B IFG0559B	6.1	5.1	ABEND interpretation, message routine
IFG0199D	IFG0209D IFG0239D IFG0559D	6.1	5.1	ABEND interpretation, control block and work area trace routine
IFG0199E	IFG0209E IFG0239E IFG0559E	6.1	5.1	Gives control to DCB ABEND exit routin
IFG0200P	IGG0206M	6.1	5.1	Close ABEND interpretation, initialization
IFG0200V		3.1	3.1	Close initialization, read JFCB
IFG0200W		3.1	3.1	Gives control to close access method executors
IFG0200Y	IGG0200F IGG0200G	3.1, 3.3	3.1	Close access method executor return
IFG0200Z		3.2	3.1	Close, SL tape output, write TRL1 and TRL2 labels
IFG0201R		3.3	3.2	Close, DA output, update F1 DSCB
IFG0202A		3.2	3.1	Close, SUL tape, read or write user labels

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
IFG0202B	IFG0232Y IFG0556B IGG0200B	3.2, 4.2	3.1	Close, Gives control to NSLTRLO or NSLETRL
IFG0202C		3.3	3.2	Close, SUL DA input, reads user labels
IFG0202D		3.3	3.2	Close, SUL DA output, write user labels
IFG0202E		3.3	3.2	Close DA final
IFG0202F		3.2	3.1	Close, Tape volume disposition
IFG0202H	IFG0553B	2.2, 2.4, 2.6, 2.7, 3.2, 3.3	3.2	Close/EOV, SMF record builder, tape and DA
IFG0202I		2.6, 2.7, 3.3	3.2	Close/EOV, SMF record builder, DA only
IFG0202J		3.4	3.2	Close, Restore ACB/DCB to preopen status
IFG0202K		3.4	3.2	Close, Delete DEB
IFG0202L	IFG0209L	3.4	3.2	Close final
IFG0230P		6.1	5.1	TCLOSE ABEND interpretation, initialization
IFG0232D		4.3	4.1	TCLOSE DA, read F1 DSCB
IFG0232G		4.2	4.1	TCLOSE SL output tape, write EOF1 label
IFG0232J		4.3	4.1	TCLOSE DA output, write user labels
IFG0232M		4.2	4.1	TCLOSE SUL tape, write EOF2 label and process user labels
IFG0232S		4.2	4.1	TCLOSE NL, SL tape positioning
IFG0232Z		4.1, 4.3	4.1	TCLOSE final
IFG0550P		6.1	5.1	EOV ABEND interpretation, initialization
IFG0551B		2.1	2.1	SYNAD diagnostic routine
IFG0551D		2.1	2.1	EOV message routine
IFG0551F		2.1, 2.2, 2.6	2.1	EOV initialization, read JFCB
IFG0551H		2.1, 2.2, 2.4, 2.6	2.1	EOV initialization, determine device type
IFG0551L	IFG0551N	2.3, 2.5, 2.6, 2.7	2.2	SAM EOV executor
IFG0551P		2.4	2.3	EOV tape output, write EOV1 label
IFG0551R		2.4	2.3	EOV tape output, write EOV2 label

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
IFG0551T	IGG0550F	2.4	2.3	EOV tape output, position current volume
IFG0552B		2.5	2.3	EOV tape output, rewrite volume label, new volume
IFG0552D		2.5	2.3	EOV tape output, write HDR1 label, new volume
IFG0552F		2.5	2.3	EOV tape output, write HDR2 and user labels, new volume
IFG0552N	IGG0550H	2.4	2.3	EOV tape input recover from invalid tape
IFG0552R	IGG0550B	2.2	2.2	EOV tape input, position current volume
IFG0552T		2.2	2.2	EOV tape input, read user labels, current volume
IFG0552V		2.2	2.2	EOV tape input, position current volume
IFG0552X		2.1, 2.2, 2.6	2.1	EOV,EOD/ concatenation
IFG0553B	see IFG0202H			
IFG0553C	EMODVOL1	2.5	1.1	EOV, modify volume 1 label, tape label conflict
IFG0553D	see IFG0193D			
IFG0553E	see IFG0193E			
IFG0553F		2.3	2.2	EOV input tape, label processing, new volume
IFG0553H	IGG0550D	2.3	2.2	EOV tape input, label processing, new volume
IFG0553P		2.6	2.4	EOV DA input, read F1 DSCB
IFG0553X		2.6	2.4	EOV DA input, build new DEB
IFG0553Z		2.6	2.4	EOV DA input, final
IFG0554A	see IFG0194A			
IFG0554B		—	2.4	EOV DA, repositions for records that span volumes
IFG0554C	see IFG0194C			
IFG0554F	see IFG0194F			
IFG0554J	see IFG0194J			
IFG0554L	IFG0195F	1.3,2.6	2.4	EOV DA input, reads user header and trailer labels
IFG0554N		2.7	2.4	EOV DA output, FEOV
IFG0554P		2.7	2.4	EOV DA output, get more space
IFG0554T		2.7	2.4	EOV DA output, B37 ABEND

Microfiche or CSECT Nam	Alias	M.O. Diagram	Module Flowchart	Function
IFG0554Z		2.7	2.4	EOV DA output, update F1 DSCB, current volume
IFG0555B		2.7	2.4	EOV DA output, write SUL on current volume
IFG0555H		2.7	2.4	EOV DA output build new DEB
IFG0555J	IFG0196P	1.8, 2.7	2.4	EOV DA output, write SUL on new volume
IFG0555T	see IFG0195T			
IFG0556B	see IFG0202B			
| IFG0558X		2.3	2.2	Checkpoint at EOV facility
IGC0001I	IGC0101I	1.1	1.1	First load of Open, SVC 19
IGC0002B		1.1	1.1	OPEN TYPE=J, SVC 22
IGC0002C		4.1	4.1	First load of TCLOSE, SVC 23
IGC00020	IGC01020	3.1	3.1	First load of Close, SVC 20
IGC0003A		2.1	2.1	FEOV, SVC 31
IGC0005E		2.1	2.1	First load of EOV, SVC 55
IGC0006D		—	—	RDJFCB, SVC 64
IGC0010C		—	—	XLATE, SVC 103
IGC0101I	see IGC0001I			
IGC01020	see IGC00020			
IGC117	see IFGDEBCK			
IGG0190A	see IFG0193D			
IGG0190B	see IFG0195D			
| IGG0190R	see IFG0196O			
IGG0190S	see IFG0196W			
IGG0200F	see IFG0200Y			
IGG0200G	see IFG0200Y			
IGG0206M	see IFG0200P			
IGG0550B	see IFG0552R			
IGG0550D	see IFG0553H			
IGG0550F	see IFG0551T			
IGG0550H	see IFG0552N			
IGG0550P	see IFG0193D			
IGG0193K		see appendix		JES access method open executor
IGG0196M		6.1	5.1	RDJFCB and open access method executor ABEND interpretation initialization
IGG0203K		see appendix		JES access method close executor

Microfiche or CSECT Name	Alias	M.O. Diagram	Module Flowchart	Function
IMDUSRFF		—	—	Formatting appendage called by AMDPRDMP to format GTF output
OMODVOL1	see IFG0193C			
READPSWD		1.7	1.3	Open/EOV password protection, asks operator for password
SECLOADA		1.7	1.3	Open/EOV password protection, initialization for Scratch, Rename and Checkpoint/Restart

Load Module-to-Object Module Cross-Reference

The table that follows shows the O/C/EOV object modules that are link-edited into load modules and the names of these load modules.

Load Module	Object Modules Contained in This Load Module
IFG0RR0A	IFG0RR0A, IFG0RR0B, IFG0RR0E, IFG0RR0F
IFG0TC0A	IFG0TC0A, IFG0TC4A
IFG019RA	IFG019RA
IFG019TR	IFG019TR
IFG0190P	IFG0190P, IFG0200P, IFG0230P, IFG0550P, IGG0196M
IFG0194A	IFG0194A, IFG0194C, IFG0194F, IFG0194J, IFG0193D, IFG0193E, IFG0195G
IFG0199B	IFG0199B, IFG0199D, IFG0199E
IGC0001I	IGC0001I, IGC0002B, IFG0193A, IFG0195A, IFG0195B, IFG0195C, IFG0195D, IFG0554L, IFG0195H, IFG0195J, IFG0195K, IFG0195N, IFG0195T, IFG0195U, IFG0195V, IFG0196J, IFG0196K, IFG0196L, IFG0196M, IFG0196N, IFG0196O, IFG0555J, IFG0196Q, IFG0196T, IFG0196U, IFG0196V, IFG0196W, IFG0196X, IFG0198N, IGG0193K
IGC00020	IGC00020, IFG0200V, IFG0200W, IFG0200Y, IFG0200Z, IFG0201R, IFG0202A, IFG0202B, IFG0202C, IFG0202D, IFG0202E, IFG0202F, IFG0202H, IFG0202I, IFG0202J, IFG0202K, IFG0202L, IGC0002C, IFG0232D, IFG0232G, IFG0232J, IFG0232M, IFG0232S, IFG0232Z, IGG0203K
IGC0005E	IGC0005E, IGC0003A, IFG0551B, IFG0551D, IFG0551F, IFG0551H, IFG0551L, IFG0551P, IFG0551R, IFG0551T, IFG0552B, IFG0552D, IFG0552F, IFG0552N, IFG0552R, IFG0552T, IFG0552V, IFG0552X, IFG0553F, IFG0553H, IFG0553P, IFG0553X, IFG0553Z, IFG0554B, IFG0554N, IFG0554P, IFG0554T, IFG0554Z, IFG0555H, IFG0555B
IGC0006D	IGC0006D
IGC0010C	IGC0010C
IMDUSRFF	IMDUSRFF
EMODVOL1	IFG0553C
OMODVOL1	IFG0193C
READPSWD	READPSWD
SECLOADA	SECLOADA

DATA AREAS

This section provides reference information about the data areas used by the O/C/EOV routines. Tables are included that show which of the O/C/EOV object modules modify the major control blocks. These tables can be used to determine which modules may be causing invalid information to be put in the control blocks.

This section also includes illustrations showing the relationships among the work areas used by the O/C/EOV routines. This information can be used to locate these work areas in storage dumps.

Several of the key internal control blocks and key fields of control blocks are included here for reference. This section also lists the data areas and control blocks used by the O/C/EOV routine to be found in *OS/VS2 System Programming Library: Debugging Handbook* and in the *OS/VS2 Data Areas* microfiche.

Control Block Activity

The tables that follow provide reference information concerning which fields of major system control blocks are modified by the O/C/EOV routines. A table is provided for each of the following control blocks:

Acronym	Common name of the data area
ACB	access method control block
DCB	data control block
DEB	data extent block
DEBX	DEB extension
DSAB	data set association block
DSCB	data set control block
FL1*xxxxx*	the first data set label on a standard (SL or AL) labeled magnetic tape
FL2*xxxxx*	the second data set label on a standard (JL or AL) labeled magnetic tape
JFCB	job file control block
TIOT	task input/output table
UCB	unit control block
VOL*xxxxx*	volume label, magnetic tape and direct access devices

All of these control blocks appear in *OS/VS2 System Programming Library: Debugging Handbook*.

For each of the control blocks in the list, the fields modified by an O/C/EOV module are listed, followed by the object module name of the module or modules that modify the field.

Note that the smallest field in the list is a single byte; if a single bit is modified, the byte of which it is a part is listed. For example, the ACBVSAM bit of the ACB does not appear; instead, the ACBAMETH field, of which ACBVSAM is a single bit, appears.

ACB—Access Method Control Block

Field Name	Module(s) That Modify the Field
ACBAMETH	IFG0193A, IGC0001I
ACBBLKSZ	IFG0196V, IFG0202K
ACBCCTYP	IFG0196V
ACBDEB	IGG0193K
ACBERFLG	IFG0193A, IFG0200V, IGC0001I, IGC0002B, IGC00020, IGC0002C, IGC0003A, IGC0005E
ACBINFL	IGC0001I
ACBLRECL	IFG0196V, IFG0202K
ACBOFLGS	IFG0193A, IFG0196V, IFG0200V

DCB—Data Control Block

Field Name	Module(s) That Modify the Field
DCBBFTEK	IFG0196J, IFG0196M
DCBBLKCT	IFG0195B, IFG0195C, IFG0195H, IFG0232S, IFG0552R, IFG0552V, IFG0553F
DCBBLKSI	IFG0196J, IFG0196K, IFG0196L
DCBBUFNO	IFG0196J
DCBBUFOF	IFG0196J
DCBIND1	IFG0232G, IFG0551L, IFG0554T, IGC0002C, IGC0005E
DCBCIND2	IFG0198N, IFG0551B, IFG0551L, IFG0554T, IGC0003A, IGC0005E
DCBDDNAM	IFG0RR0B, IFG019RA, IFG0202K
DCBDEBAD	IFG0555H, IGC0001I, IGC0002B
DCBDEVT	IFG0196J, IFG0202K, IFG0551L
DCBDIRCT	IFG0195A, IFG0200Y
DCBDSORG	IFG019RA, IFG0193A, IFG0194A, IFG0194C, IFG0195A, IFG0196L, IFG0555H
DCBDVTBL	IFG0196J, IFG0202K, IFG0551L
DCBFDAD	IFG0202E, IFG0232D, IFG0232Z, IFG0553Z, IFG0555H
DCBHIARC	IFG0RR0B, IFG0202J
DCBH0	IFG0196J, IFG0196M, IFG0202J
DCBH1	IFG0196J, IFG0196M, IFG0202J
DCBIFLG	IFG0RR0B, IFG0202K
DCBIFLGS	IFG0193D, IFG0193E, IFG0196N, IFG0196Q, IFG0196T, IFG0196U, IFG0200Y, IFG0232S, IFG0551B, IFG0551F, IFG0551L, IFG0552X, IGC0001I, IGC0002B
DCBIOBA	IFG0551B, IFG0551L, IGC0003A
DCBIOBAD	IFG0553X, IFG0553Z, IFG0554B
DCBKEYCN	IFG0551L
DCBKEYLE	IFG0196J
DCBLRECL	IFG0196J, IFG0196K, IFG0196L
DCBMACR	IFG0RR0B, IFG0202K
DCBNCRHI	IFG0196J
DCBOFFSR	IFG0196X, IFG0198N
DCBOFFSW	IFG0196X, IFG0198N

DCB—Data Control Block

Field Name	Modules(s) That Modify the Field
DCBOFLGS	IFG019RA, IFG0193A, IFG0195D, IFG0195G, IFG0195N, IFG0196L, IFG0196O, IFG0196U, IFG0196W, IFG0198N, IFG0199E, IFG0200V, IFG0200Y, IFG0202A, IFG0202C, IFG0202D, IFG0202K, IFG0232J, IFG0323M, IFG0232S, IFG0232Z, IFG0551F, IFG0551H, IFG0551L, IFG0551R, IFG0551T, IFG0552F, IFG0552R, IFG0552T, IFG0553F, IFG0553H, IFG0553Z, IFG0554L, IFG0554P, IFG0554T, IFG0554Z, IFG0555B, IFG0555J, IGC0001I, IGC0002B, IGC0002C, IGC00020, IGC0003A, IGC0005E
DCBOPTCD	IFG0RR0B, IFG0196L, IFG0202J
DCBQSWS	IFG0198N, IFG0202J
DCBRECAD	IFG0551L, IFG0554T, IGC0003A
DCBRECFM	IFG0RR0B, IFG0195K, IFG0196K, IFG0196L, IFG0202J, IFG0552F
DCBTIOT	IGC0001I, IGC0002B
DCBTRBAL	IFG0551L
DCBWCPO	IFG0551L
DCBUSASI	IFG0196J, IFG0196M, IFG0202J

DEB—Data Extent Block

Field Name	Module(s) That Modify the Field
DEBAPPAD	IFG0553X, IFG0555H, IGG0193K
DEBDCBAD	IFG0232G, IFG0232S, IFG0232Z, IFG0551F, IFG0552F, IFG0552X, IFG0553H, IFG0553X, IFG0553Z, IFG0554B, IFG0554T, IFG0555H, IGC0002C, IGG0193K
DEBDEBAD	IFG0553X, IFG0554B, IFG0555A, IGG0193K
DEBDEBID	IGG0193K
DEBDVMOD	IFG0552B, IFG0553X, IFG0554B, IFG0555H
DEBFLGS1	IFG0195T, IFG0196W, IFG0200W, IFG0552B, IFG0552X, IFG0553F, IFG0554P
DEBLNGTH	IGG0193K
DEBNMEXT	IFG0551F, IGC0001I, IGC0002B, IGC00020, IGG0193K
DEBNMSUB	IFG0196X
DEBNMTRK	IFG0553X, IFG0555H
DEBOFLGS	IFG0196W, IFG0196X, IFG0232Z, IFG0552X, IFG0553X, IGC0002C, IGG0193K
DEBOPATB	IGG0193K
DEBPROTG	IGG0193K
DEBSTRCC	IFG0553X, IFG0555H
DEBSUCBA	IFG0552X
DEBTCBAD	IFG0553X, IGG0193K
DEBUCBAD	IFG0554B, IGC00020, IGG0193K
DEBVOLNM	IFG0553X
DEBVOLSQ	IFG0552F, IFG0552X, IFG0553X, IFG0554B

DEBX—DEB Extension

Field Name	Module(s) That Modify the Field
DEBXDBPR	IFG0196W, IFG0553X, IFG0554B, IFG0555H
DEBXDCBM	IFG0196W
DEBXDSAB	IFG0193A, IFG0552V, IFG0552X
DEBXDSO1	IFG0193A, IFG0194C
DEBXFLG1	IFG0TC0A, IFG0232G, IFG0232S, IFG0232Z, IFG0551F, IFG0552F, IFG0552X, IFG0553H, IFG0553X, IFG0553Z, IFG0554T, IFG0555H, IGC0002C
DEBXLNGH	IFG0193A
DEBXMCF1	IFG0193A

DSAB—Data Set Association Block

Field Name	Module(s) That Modify the Field
DSABFLG2	IFG0196M, IFG0198N, IFG0552V, IFG0552X
DSABFLG4	IFG0193D, IFG0194A, IFG0194C, IFG0195B, IFG0195K, IFG0196Q, IFG0196T, IFG0196V, IFG0551R, IFG0552B, IFG0552X, IFG0553F, IGC0002B
DSABOPEN	IFG0198N, IFG0552V, IFG0552X
DSABPTTR	IFG0195U, IFG0195V

DSCB—Format-1 Data Set Control Block

Field Name	Module(s) That Modify the Field
DS1REFD	IFG0194C
DS1DSCHA	IFG0194C
DSCCREDT	IFG0195A
DSCDSIND	IFG0194C,IFG0196W,IFG0200Y,IFG0232D,IFG0554N,IFG0544Z
DSCEXPDT	IFG0195A
DSCEXTYP	IFG0553X, IFG0555H
DSCEXTZ	IFG0194C, IFG0196W, IFG0200V, IFG0201R, IFG0232D, IFG0553P, IFG0554N, IFG0554T, IFG0554Z
DSCFILSR	IFG0193D, IFG0196N
DSCFILTY	IFG0194C, IFG0196W, IFG0553X
DSCFMTID	IFG0194C, IFG0196W, IFG0200V, IFG0201R, IFG0232D, IFG0553P, IFG0554N, IFG0554T, IFG0554Z
DSCKEYL	IFG0196W
DSCLRECL	IFG0196W
DSCLSTAR	IFG0195A, IFG0200Y, IFG0202E, IFG0232D, IFG0554N, IFG0554T, IFG0554Z
DSCNEXT	IFG0194C, IFG0554P
DSCOPTCD	IFG0196W
DSCRECFM	IFG0196W
DSCRKP	IFG0196W
DSCSCALO	IFG0194C
DSCTRBAL	IFG0554Z

FL1—First Standard Data Set Label on a Magnetic Tape

Field Name	Module(s) That Modify the Field
FL1BLKCT	IFG0196T, IFG0200Z, IFG0232G, IFG0552D
FL1CREDT	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1EXPDT	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1FILSQ	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1FILSR	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1FSEC	IFG0196N, IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552B, IFG0552D
FL1GNO	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1ID	IFG0193D, IFG0195H, IFG0196Q, IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552B, IFG0552D
FL1LABI	IFG0196T, IFG0200Z, IFG0232G, IFG0232M, IFG0551P, IFG0551R, IFG0552D, IFG0552F
FL1NO	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0551R, IFG0552F
FL1SYSCD	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1VNG	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D
FL1VOLSQ	IFG0196T, IFG0200Z, IFG0232G, IFG0551P, IFG0552D

FL2—Second Standard Data Set Label on a Magnetic Tape

Field Name	Module(s) That Modify the Field
FL2BLKA	IFG0196T, IFG0200Z
FL2BLKL	IFG0196T, IFG0200Z, IFG0232G, IFG0551R, IFG0552F
FL2BUFOF	IFG0196T, IFG0200Z, IFG0232M, IFG0551R, IFG0552F
FL2CNTRL	IFG0196T, IFG0200Z, IFG0232M, IFG0551R, IFG0552F
FL2DEN	IFG0196T, IFG0200Z, IFG0232G, IFG0551R, IFG0552F
FL2FILP	IFG0196T, IFG0200Z, IFG0232G, IFG0551R, IFG0552F
FL2JOBD	IFM0196T, IFG0200Z, IFG0232M, IFG0551R, IFG0552F
FL2JSSP	IFG0196T, IFG0200Z, IFG0232M, IFG0551R, IFG0552F
FL2LRECL	IFG0196T, IFG0200Z, IFG0232G, IFG0551R, IFG0552F
FL2RECFM	IFG0196T, IFG0200Z, IFG0232G, IFG0551R, IFG0552F
FL2STEPD	IFG0196T, IFG0200Z, IFG0232M, IFG0551R, IFG0552F
FL2TRTCH	IFG0196T, IFG0200Z, IFG0232G, IFG0232M, IFG0551R, IFG0552F

JFCB—Job File Control Block

Field Name	Module(s) That Modify the Field
JFCBCRDT	IFG0195A
JFCBCTRI	IFG0195A
JFCBDSNM	IFG0194C, IFG0553X, IFG0554P, IFG0555H
JFCBELNM	IFG0553X
JFCBEXAD	IGC0002B
JFCBFLG1	IFG0193D, IFG0194F, IGC0002B
JFCBFLG2	IFG0193A, IFG0194A, IFG0196X, IFG0198N
JFCBFLSQ	IFG0193A, IFG0553D, IFG0553F
JFCBFOFL	IFG0196M, IFG0196T, IFG0200Z

JFCB—Job File Control Block

Field Name	Module(s) That Modify the Field
JFCBFTEK	IFG0196M
JFCBIND1	IFG0193A, IFG0553X
JFCBIND2	IFG0194A, IFG0195K, IFG0196N, IFG0196X
JFCBLGTH	IFG019RA, IFG0194A, IFG0551H, IFG0552N, IFG0553X, IFG0555H, IGC0002C
JFCBLKSI	IFG0195A, IFG0195K, IFG0196V
JFCBLTYP	IFG0553X
JFCBMASK	IFG0193A, IFG0193D, IFG0193E, IFG0194A, IFG0194C, IFG0194F, IFG0195A, IFG0195K, IFG0196J, IFG0196K, IFG0196L, IFG0196M, IFG0196N, IFG0196V, IFG0196W, IFG0196X, IFG0198N, IFG0199E, IFG0200V, IFG0202K, IFG0553P, IFG0553Z, IFG0554Z
JFCBNVOL	IFG0193A, IFG0194A, IGC0002B
JFCBOTTR	IFG0195A
JFCBSQTY	IFG0554P
JFCBTSDM	IFG0194A, IFG0195A
JFCBUFOF	IFG0195K, IFG0196M
JFCBVLSQ	IFG0195B, IFG0196X
JFCBVOLS	IFG0194A, IFG0196O, IFG0199E, IGC0002B
JFCBXPDT	IFG0195A, IFG0195H
JFCCPRI	IFG0196M
JFCDEN	IFG0194F
JFCDSORG	IFG0193A, IFG0195A
JFCDSRG2	IFG0193A, IFG0200V, IFG0202K, IFG0551F
JFCFLGS1	IFG0196X, IFG0551F, IFG0551L
JFCFUNC	IFG0196L
JFCINTVL	IFG0196M, IFG0553X
JFCIPLTX	IFG0196L
JFCKEYLE	IFG0195A, IFG0196W
JFCLRECL	IFG0195A, IFG0195K, IFG0196V
JFCOPTCD	IFG0194F, IFG0195A
JFCRECFM	IFG0195A, IFG0195K, IFG0196L, IFG0196V
JFCRKP	IFG0195A
JFCSOWA	IFG0196M
JFCTRKBL	IFG0195A
JFCTRTCH	IFG0195K

TIOT—Task Input/Output Table

Field Name	Module(s) That Modify the Field
TIOELINK	IFG0202E
TIOERLOC	IFG0232D
TIOESTTB	IFG0194C, IFG0194F

UCB—Unit Control Block

Field Name	Module(s) That Modify the Field
UCBDMCT	IFG0193C, IF0194F, IFG0195B, IFG0195D, IFG0196O, IFG0551T, IFG0552B, IFG0552V, IFG0553C, IFG0553H, IGC0005E
UCBERADR	IFG0RR0B, IFG0193A, IFG0202J
UCBFSCT	IFG0RR0B, IFG0194F, IFG0194J, IFG0195B, IFG0195C,
UCBFSEQ	IFG0RR0B, IFG0194F, IFG0194J, IFG0195B, IFG0195C, IFG0202F
UCBFSER	IFG0194C, IFG0194F, IFG0194J
UCBSTAB	IFG0193D, IFG0193E, IFG0194C, IFG0194F, IFG0194J
UCBSTAT	IFG0193D, IFG0193E, IFG0194F
UCBVOLI	IFG0194A, IFG0194C, IFG0194F, IFG0194J, IFG0195D, IFG0553H, IGC0002B
UCBVTOC	IFG0194C
UCBWGT	IFG0194F, IFG0194J, IGG0202F, IFG0551H, IFG0551T, IFG0552V

Volume Label

Field Name	Module(s) That Modify the Field
VOLLABI	IFG0193E, IFG0196Q, IFG0552B
VOLOWNER	IFG0193E
VOLSEC	IFG0193E, IFG0196Q, IFG0552B
VOLSERNO	IFG0193E, IFG0194F, IFG0196Q, IFG0552B

Data Area Layouts

The illustrations that follow establish the relationships among the work areas used by O/C/EOV.

Data Area Relationships During O/C/EOV Processing

Several work areas may be obtained, based on the type of request being processed by the O/C/EOV routines. These work areas are obtained separately and at different times and may not, therefore, be contiguous in virtual storage. Figure 3 is intended to help you find the work areas obtained during an O/C/EOV operation in a storage dump. In the example in Figure 3, two DCBs are being opened. The first work area obtained, whose ID is O/C, contains the base prefix and the only extended prefix obtained for the OPEN request. This first work area contains an extended prefix, a WTG table (the only WTG table obtained for this OPEN request), a copy of the Open parameter list, a recovery routine audit trail, and the storage used by the O/C/EOV-DADSM service routine, IFG019RA, to synchronize parallel processing. Figure 4 illustrates some of the key information included in this work area.

As processing of the Open request progresses, an O/C/EOV main work area and an area into which the DCB can be copied is obtained for each DCB. The IDs of these areas (which appears as the first four characters of the prefix of each work area) is OPWA and DCB for the copy of the DCB. Each block of storage obtained includes a prefix.

Next, the Open routines obtain a work area used to build an extended channel program for reading a format-1 DSCB (the ID is CCWS), and an area to be used for reading and writing JFCBs from the SWA using the QMNGRIO

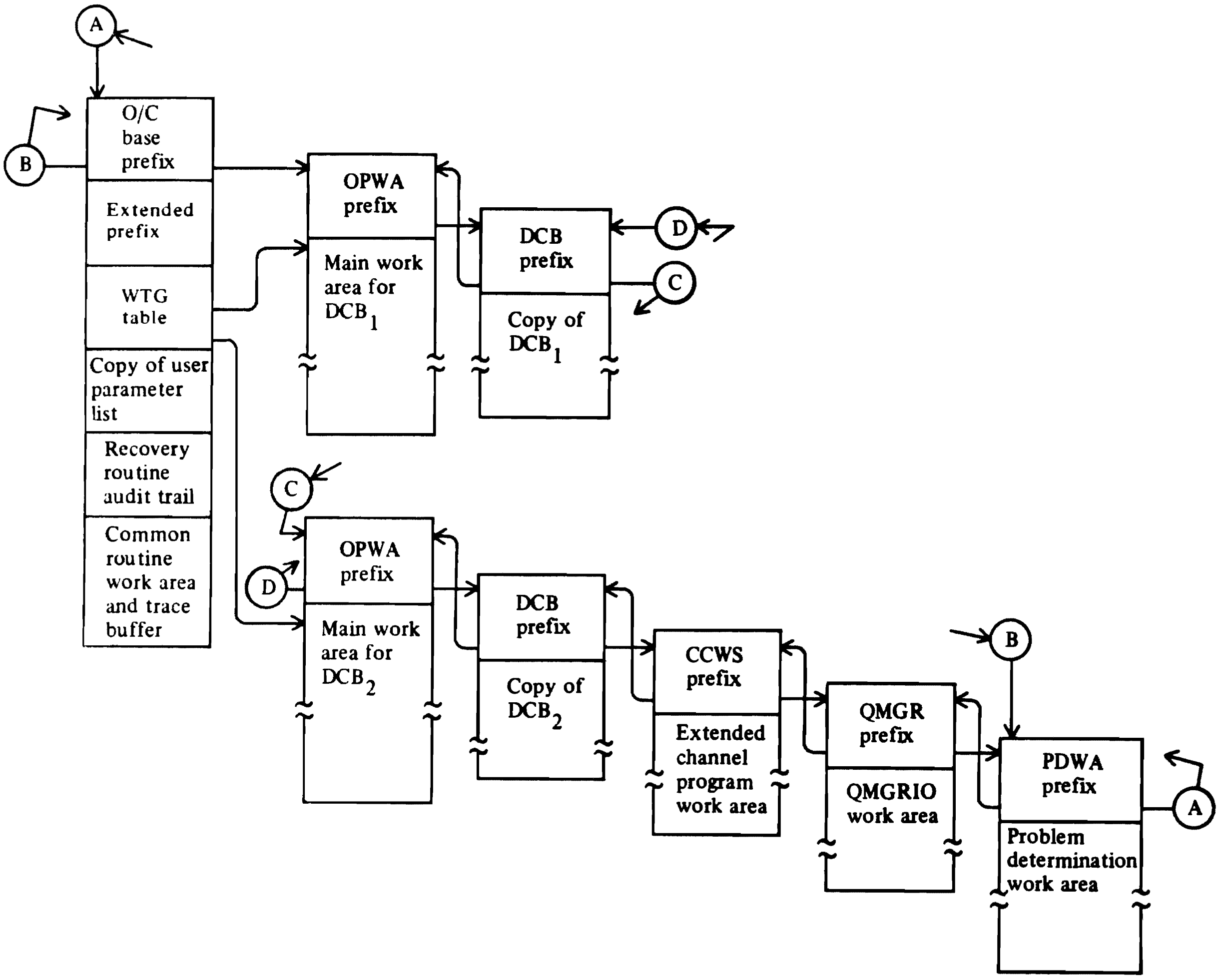

A
B
O/C base prefix
Extended prefix
WTG table
Copy of user parameter list
Recovery routine audit trail
Common routine work area and trace buffer
OPWA prefix
Main work area for DCB_1
DCB prefix
Copy of DCB_1
D
C
C
D
OPWA prefix
Main work area for DCB_2
DCB prefix
Copy of DCB_2
CCWS prefix
Extended channel program work area
QMGR prefix
QMGRIO work area
B
PDWA prefix
Problem determination work area
A

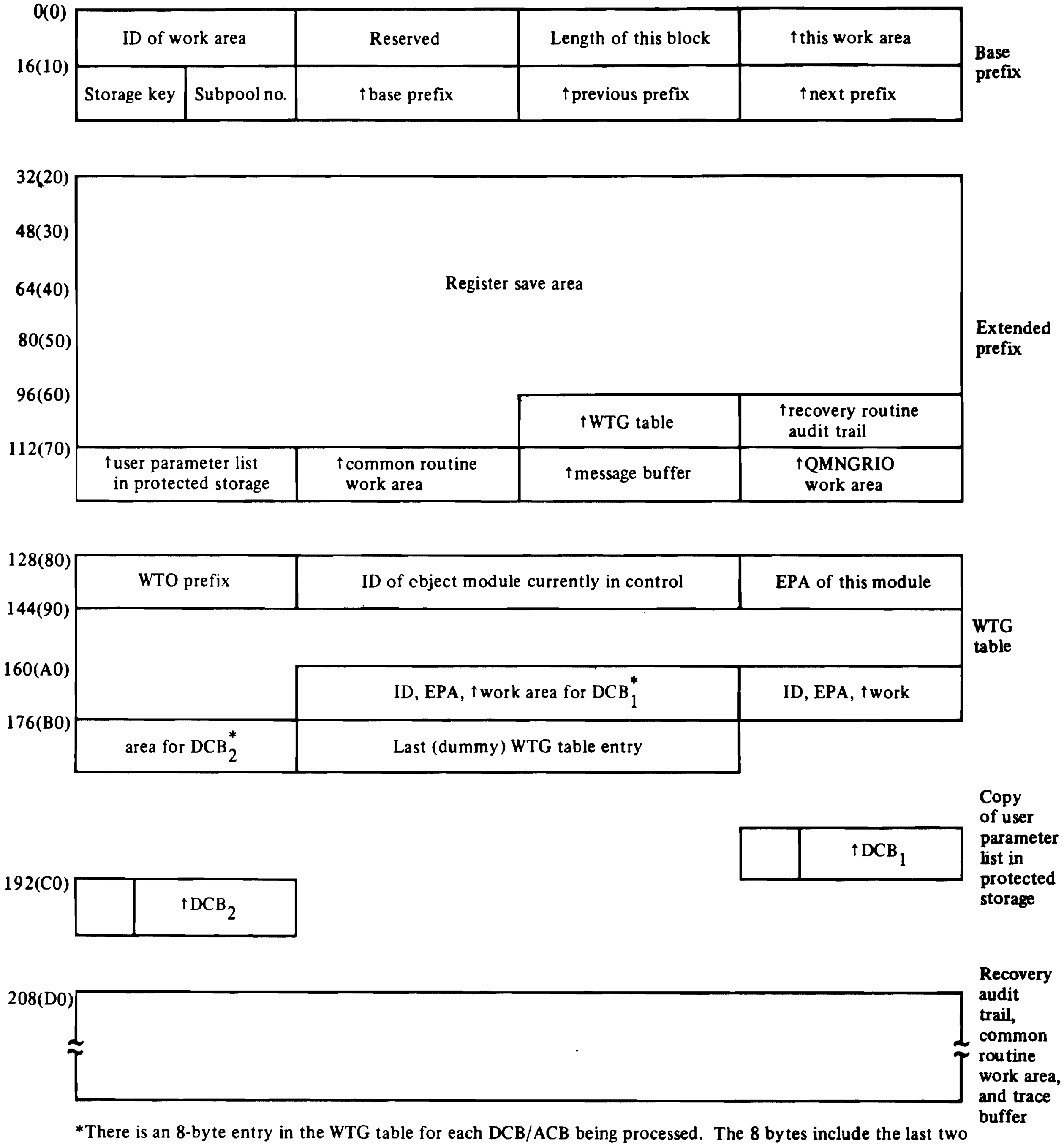
Figure 3. Work Areas Obtained and the Chain Structure: Two DCBs Being Opened in Parallel

interface (ID is QMGR), and finally an abnormal termination condition is encountered and an area is obtained for the problem determination routines to gather diagnostic information (ID is PDWA). A list of all of the work area and control block IDs that are obtained during O/C/EOV processing appears in the section "ABEND Error Recording From the O/C/EOV System Recovery Routines."

Using a Work Area Prefix to Find Another Related Work Area

During the initialization of a multiple Open or Close request—initiated, for example by OPEN (DCB1,,DCB2(OUTPUT))—main work areas and prefixes are obtained to process each DCB and ACB. These work areas are used to read and write labels, to contain the JFCB, etc. The contents of the work area is provided in *OS/VS2 Data Areas.*

The prefix for the first work area obtained is called the *initial* or *base* prefix. No main work area is obtained with this prefix if the request is being processed by the Open or Close routines. For EOV requests, a main work area is obtained. For an Open or Close request, this work area contains control information used by the Open or Close routines for all the DCBs and



*There is an 8-byte entry in the WTG table for each DCB/ACB being processed. The 8 bytes include the last two bytes of the module name of the next O/C/EOV module that is to process the DCB, the entry point address of that module, and the address of the main work area associated with the DCB/ACB.

Figure 4. Contents of the Initial Work Area Obtained to Process the Two DCBs in Figure 3

ACBs included in the request. Each related work area obtained subsequently is chained, forward and backward, from the initial prefix.

The WTG table can also be used to locate the main work area associated with an ACB or DCB. One WTG table is obtained for each OPEN or CLOSE macro issued. This table is used to transfer control during Open and Close processing, as described in "WTG Table" in this section. The WTG table also contains addresses of the main work areas used for processing each DCB or

ACB. Beginning at +32, the WTG table contains an 8-byte entry for each DCB or ACB being processed. The first 5 bytes of each entry contains a 2-byte module ID and a 3-byte entry point address of the next module that is to process the DCB/ACB. The last 3 bytes of each 8-byte entry contains the address of the work area associated with the DCB or ACB. Additional chaining is done from the extended prefix to the WTG table. Field IECWTGTB at +104 from the beginning of the prefix contains the address of the WTG table. This field follows an 18-word register save area.

See also "Register Usage" in the "Diagnostic Aids" section for the registers that contain addresses relating to these data areas.

Only one DCB/ACB is processed at a time by the EOV component. Otherwise, the data area relationships mentioned in the foregoing descriptions of the Open and Close work areas apply.

Copying the DCB/ACB

To avert malicious and unintentional access to the DCB/ACB during O/C/EOV processing, the DCB/ACB and the parameter list containing the DCB/ACB addresses, passed by the caller of O/C/EOV, are copied into protected storage (virtual storage with a system protection key, key 0 through 7).

The DCB/ACB copy can be located using the DXPDCBAD field of the main O/C/EOV work area associated with the DCB/ACB. The copy of the parameter list can be located using the IECUPRML field of the extended prefix.

All modifications to the DCB/ACB (for example, the JFCB-to-DCB merge during Open, the restoring of the DCB to preopen status during Close, and the updating of the DCBBLKCT field during EOV processing) are made to the copy. Before giving control to a user's routine during O/C/EOV processing (for example, to the user's DCB, ACB, or EOV exit routine) and before returning control at the conclusion of O/C/EOV processing, the DCB/ACB is copied back to the user's storage.

Internal Data Areas and Key Fields

The data area layouts and control block presentations that follow are for key data areas and fields of data areas that are important in O/C/EOV processing. They are presented in alphameric order.

DCBs and ACBs Processed

The table that follows shows the two types of DCBs and the three types of ACBS processed by O/C/EOV and how each is identified.

Control Block Name	Type of Data Set Represented by Control Block	Control Block Indicators Used to Identify the Control Block
VSAM ACB	User VSAM data set	ACBAMETH field set to X'11' JFCB has been read in
VTAM ACB	User VTAM data set	ACBAMETH field set to X'60'
SYSIN/SYSOUT DCB	SYSIN/SYSOUT data set spooled by a job entry subsystem	TIOELINK field in TIOT contains X'04' for input or X'02' for output
SYSIN/SYSOUT ACB	SYSIN/SYSOUT data set, created from spooled DCB	TIOELINK field in TIOT contains X'04' for input or X'02' for output
Unspooled DCB	User data set	None of the above indicators present

DCBOFLGS or ACBOFLGS Field

This 1-byte field is located in the foundation segment of the DCB or ACB. The field contains input and output information necessary for the Open, Close, and EOV operations to process the DCB or ACB. The names of the O/C/EOV object modules that modify the OFLGS field are shown earlier in the "Control Block Activity" section.

Description of DCBOFLGS or ACBOFLGS Field

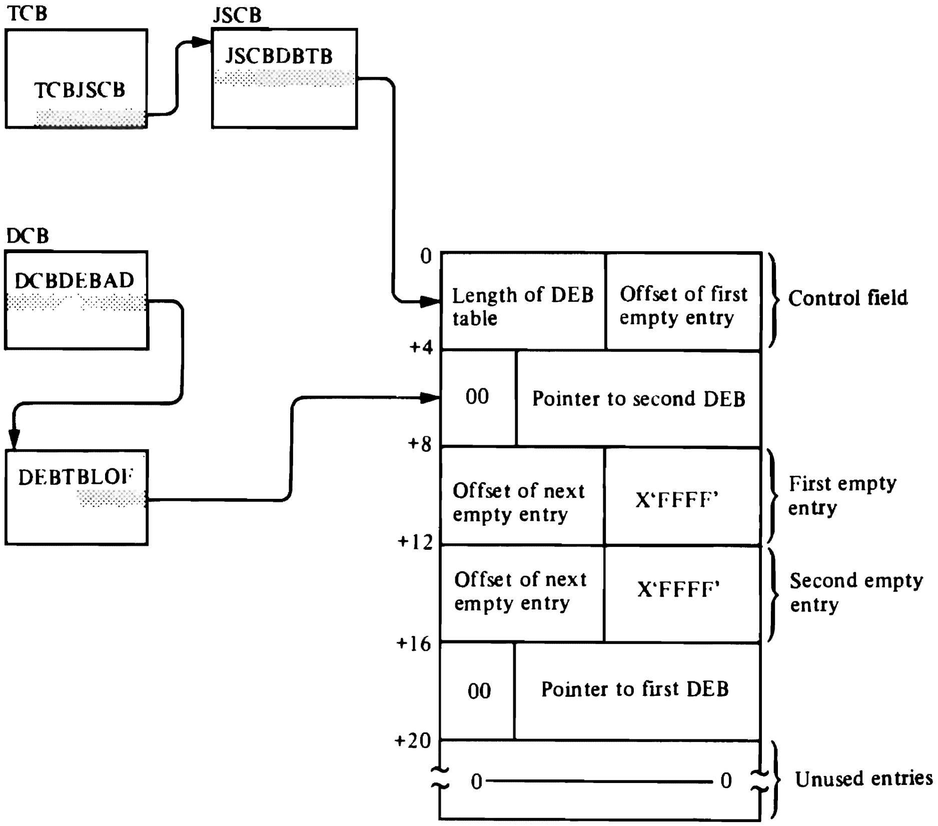
Bit	Meaning
1...	Last I/O operation was a WRITE.
0...	Last I/O operation was a READ or POINT. For direct-access devices, the track balance field is invalid.
.1..	Last I/O operation was a READ backward.
..1.	Concatenation of data sets in process.
...1	An Open operation has been successfully completed.
.... 1...	A processing program has indicated a concatenation of unlike attributes.
.... .1..	A tapemark has been read.
.... ..0.	An Open, Close, or EOV routine took a user exit and the user's processing has not been completed. (This bit setting prevents other Open, Close, and EOV routiness from processing this DCB or ACB.)
.... ..1.	An Open, Close, or EOV routine took a user exit and the user's processing has been completed.
....1	This DCB or ACB is to be processed by the Open, Close, or EOV operation that sets this bit to 1.

DEB Table

A DEB table is obtained and updated for each job step by the DEB Validity Checking routine. (For a description of the routine, see Diagram 5.3.) The DEB table is located in protected storage in subpool 255, key 0, and contains the address of each DEB in the job step.

When the address of a DEB is added to the DEB table, the high-order byte of the address is set to zero. When a DEB pointer is deleted or purged from the DEB table, the first two bytes are set to the offset of the next empty entry and the last two bytes are set to binary ones.

The DEB Validity Checking routine gets 80 bytes of storage for the DEB table and adds DEB addresses to the table until it is full. Then the DEB Validity Checking routine gets storage space equal to the current table, plus 80 bytes, and copies the DEB table to the new area. The maximum DEB table size allowed is 32,760 bytes. When the maximum size is exceeded, the job step will be terminated.



DEBXDCBM (DEB Extension Mask) Field

Each DEB and its extension is associated with a DCB or ACB. The DEB contains information about the physical characteristics of the data set and other information that is being used by the control program.

The DEBXDCBM field, located in the DEB extension, is a DCB or ACB modification mask, which is built by the Open merge routines to indicate which DCB or ACB fields are modified during merging of the JFCB fields into the DCB. One bit corresponds to one or more fields. If a DCB or ACB field is modified, the Open merge function sets the bit associated with the field to 1. The mask is then stored in the DEB extension.

The Close final function examines the DEBXDCBM field to determine which fields of the DCB or ACB are to be restored to the status they had before they were processed by the Open merge function. (Close restores the fields whose corresponding bits are set to 1.)

The three columns of the following list show the bit definition of each byte of the DEBXDCBM field and which fields in the DCB may be modified.

Description of DEBDCBMK Field

Byte	Bit	Field(s) Modified
DEBXDCBM+0	xxxx xxxx	Not used
DEBXDCBM+1	x...	Not used
	.1..	DCBIFLG
	..1.	DCBNTM,DCBPCI
	...1	DCBCYLOF
	.... 1...	DCBRKP,DCBRESER
	.... .1..	DCBDBUFN
	.... ..1.	DCBNCP(ISAM)
	1	DCBEROPT
DEBXDCBM+2	1...	DCBLRECL
	.1..	DCBLIMCT
	..1.	DCBNCP(BSAM)
	...1	DCBBLKSI,DSBBUFSI
	.... 1...	DCBOPTCD
	.... .1..	DCBRECFM
	.... ..1.	DCBDEN
	1	DCBFTEK,DCBBFALN, DCBGNCP,DCBBUFMA
DEBXDCBM+3	1...	DCBBUFL
	.1..	DCBBUFNO
	..1.	DCBKEYLE,DCBTRTCH
	...1	DCBSOWA
	.... 1...	DCBCPRI,DCBTHRES
	.... .1..	DCBINTVL
	.... ..1.	DCBLRECL
	1	DCBDSORG

JFCBMASK Field

A JFCB is built in the scheduler work area (SWA) by the scheduler routines for each ddname specified in a job step. A JFCB is brought into virtual storage when a DCB or ACB with the corresponding ddname is to be processed.

The JFCBMASK field, located in the JFCB, is used by Open, Close, and EOV to indicate that a DCB or ACB has been modified, and to flag conditions that alter the normal processing of a DCB.

Open uses the first 4 bytes to indicate which fields in the DCB or ACB are modified during the Open operation. One bit corresponds to one field (or more than one field if the fields are related). If a field is to be modified, Open sets its bit to 1. When the JFCB-to-DSCB merge has been completed, these 4 bytes of the JFCBMASK field are stored in the DEBXDCBM field of the DEB extension. These first 4 bytes of the JFCBMASK field are set to zero before the JFCB is copied back to its original location in the SWA. See "DEBXDCBM Field" for more information on this field.

The Open, Close, and EOV routines use the remaining 4 bytes of the JFCBMASK field to signify certain conditions under which the associated DCB or ACB is to be processed.

The first two columns of the following list show each byte of the JFCBMASK field and each bit used within the byte. The third column shows which operations use the bit. The fourth column tells what each bit is used to indicate.

Description of JFCBMASK Field

Byte	Bit	Used by	Indicates
JFCBMASK+0	xxxx xxxx		Reserved
JFCBMASK+1	1...	Open	DCBIPLTX field has been modified (TCAM only)
	.1..	Open	DCBUFOF field merged from JFCBUFOF field of the JFCB for BSAM or QSAM.
	..1.	Open Open	DCBNTM field modified for ISAM. DCBPCI field modified for TCAM.
	...1	Open	DCBCYLOF field modified for ISAM.
	.... 1...	Open Open	DCBRKP field modified for ISAM. DCBRESER field modified for TCAM.
	.... .1..	Open	DCBDBUFN field modified for ISAM.
	.... ..1.	Open	DCBNCP field modified for ISAM.
	1	Open	DCBEROPT field modified for ISAM or QSAM.
JFCBMASK+2	1...	Open	DCBLRECL field modified for ISAM or QSAM.
	.1..	Open	DCBLIMCT field modified for BDAM.
	..1.	Open	DCBNCP field modified for BSAM.
	...1	Open	DCBBLKSI and DCBBUFSI fields modified for TCAM.
	.... 1...	Open	DCBOPTCD field modified.
	.... .1..	Open	DCBRECFM field merged from JFCRECFM field of the JFCB.
	.... ..1.	Open	DCBDEN field modified.
	1	Open Open	DCBFTEK and DCBBFALN fields modified, and DCBGNCP field merged from JFCNCP field of the JFCB. DCBBUFMA field modified for TCAM.
JFCBMASK+3	1...	Open	DCBBUFL field modified.
	.1..	Open	DCBBUFNO field modified.
	..1.	Open	DCBKEYLE field merged from JFCKEYLE field in the JFCB, and DCBTRTCH field modified.
	...1	Open	DCBSOWA field merged from JFCSOWA field of the JFCB for QTAM.
	.... 1...	Open Open	DCBCPRI field merged from JFCCPRI field of the JFCB for QTAM. DCBTHRES field modified for TCAM.
	.... .1..	Open	DCBINTVL field merged from JFCINTVL field in the JFCB for QTAM.
	.... ..1.	Open	DCBLRECL field merged from JFCLRECL field in the JFCB.

Description of JFCBMASK Field

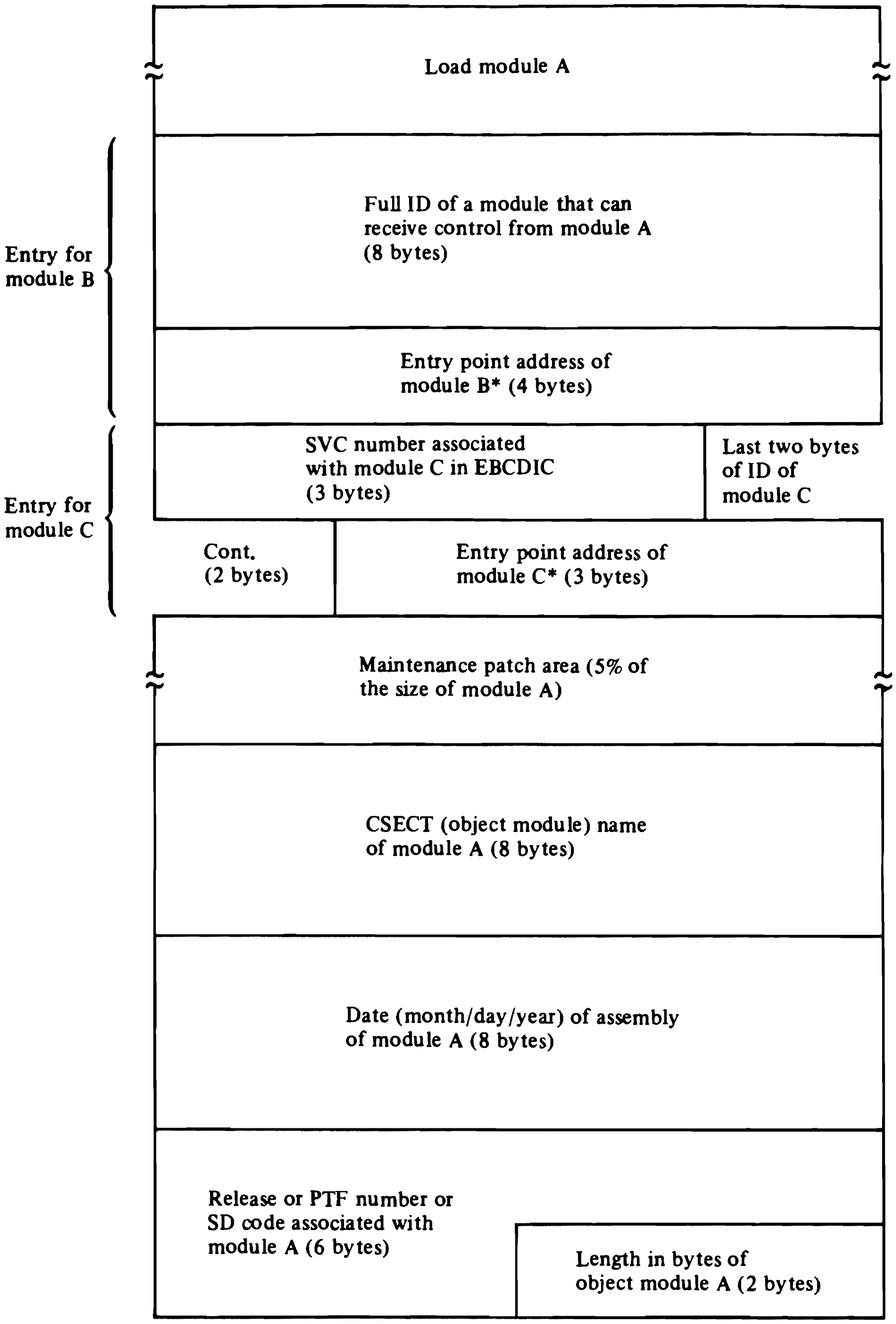
Byte	Bit	Used by	Indicates
	1	Open	DCBDSORG field merged from JFCDSORG field of the JFCB.
JFCBMASK+4	1...	Open	JFCB has been modified and is to be written back, unless bit 4 of JFCBTSDM field is set to 1.
		Close	JFCB has been read.
	0...	Open	JFCB has not been modified; however, the Open operation assumes, at the end of its processing, that the bit is set to 1 and copies the JFCB back to the SWA.
	.1..	Open	DSCB has been modified and is to be written back.
	..1.	Open	Data set being processed is a dummy (null) data set.
	...1	Open	BPAM concatenation.
	.... 1...	Open	Direct access parallel mounting for ISAM or BDAM.
	.... .1..	Open	Password required.
	.... ..1.	Open, Close, EOV	Delayed ABEND pending (bit 7 of DCBOFLGS is set to zero, or the processing program recovered from an ABEND (bit 7 of DCBOFLGS set to 1).
	1	Open	Volume sequence number modified by DISP=MOD; direct-access recovery volume sequence number set to 0 after DEB built.
JFCBMASK+5	1...	Open, EOV	Output magnetic tape label conflict.
	.1..	Open, EOV	IBM standard or American National Standard labeled tape required for output magnetic tape.
	..1.	Open, EOV	Nonstandard or nonlabeled tape required for output magnetic tape.
	...1	Open, EOV	Dual density check required for output magnetic tape. 1100 .... create standard label 1010 .... destroy standard label or American National Standard label 1110 .... change IBM standard label to American National Standard or American National Standard label to IBM standard label
	.... 1...	Open	Merge during standard label or nonstandard label processing to be bypassed after a remount of an output tape volume.
	.... .00.	Open, EOV	User-label routine issued a CLOSE TYPE=T macro instruction or SVC 23.
	.... .10.	Open, EOV	User-label routine issued an OPEN macro instruction or SVC 19.
	.... .11.	Open, EOV	User-label routine issued an SVC 55 or SVC 31.
	x		Reserved
JFCBMASK+6	1...	Open	LABEL=(,,,IN) is specified on DD card. This bit is set to 1 by the job scheduler to change the open operation from INOUT mode to INPUT mode.

Description of JFCBMASK Field

Byte	Bit	Used by	Indicates
	.1..	Open	LABEL=(,,,OUT) is specified on DD card. This bit is set to 1 by the job scheduler to change the open operation from OUTIN mode to OUTPUT mode.
			Note: Both bits 0 and 1 are set to 1 during Open if the mode is being overridden.
	..0.	Open	A direct-access device with the rotational positioning feature is being used.
	..1.	Open	The direct-access RPS feature is not being used.
	...1	Open	DISP=NEW is to be changed to DISP=MOD.
	.... 0...	Open	Search direct is not being used.
	.... 1...	Open	Search direct is being used.
	.... .1..	Open, Close, EOV	Work areas associated with this DCB or ACB are to be written to the GTF data set each time control is transferred from one O/C/EOV object to another.
	.... ..0.	Open	Open operation, after it was called, stored a TTR in the JFCBOTTR field.
	.... ..1.	Open	JFCBUFOF field contains, before Open processing begins, a buffer offset or invalid information resulting from a JFCB-to-JFCB merge.
	1	Open	TTR of the identifier (format-1) DSCB of the first volume has been updated in the JFCBDSCB field by Open and should be updated in the catalog by the scheduler at step termination, if the data set is cataloged.
JFCBMASK+7	1...	Open	Another tape volume is to be mounted and/or verified during Open.
	.xx.	Open	When the open option is OUTINX or EXTEND, the JFCDISP flags are saved in these bit positions. The JFCDISP flags are restored after the access method executors complete processing.
	...x xxxx		Reserved

Transfer Control Table

The transfer control table (a list of module names and their entry point addresses) is used to transfer control among the O/C/EOV modules. The transfer control table appears at the end of each object module (CSECT) as a result of the expansion of the XCTLTABL macro instruction. There are two options that can be used in coding the XCTLTABL macro, depending on how the module ID is coded in the ID=operand of the macro. If the full, 8-charcter module ID is coded, the result is shown in "Entry for module B" in the illustration below. If only the last two characters of the ID are coded in the ID=operand, the result is shown in "Entry for module C" below. The first option (ID=2-byte object module ID) is used with the IECRES LOAD,BRANCH=QUEUED macro. It is used to make it possible to process multiple-DCB open and close requests in parallel (see"Parallel Processing of DCBs and ACBs and Overlapping I/O"). The 8-byte option is used with the IECRES LOAD,BRANCH=DIRECT macro. This form is used when sequential processing of a single DCB is desired (EOV routines use BRANCH=DIRECT).



Load module A

Entry for module B

Full ID of a module that can receive control from module A (8 bytes)

Entry point address of module B* (4 bytes)

Entry for module C

SVC number associated with module C in EBCDIC (3 bytes)

Last two bytes of ID of module C

Cont. (2 bytes)

Entry point address of module C* (3 bytes)

Maintenance patch area (5% of the size of module A)

CSECT (object module) name of module A (8 bytes)

Date (month/day/year) of assembly of module A (8 bytes)

Release or PTF number or SD code associated with module A (6 bytes)

Length in bytes of object module A (2 bytes)

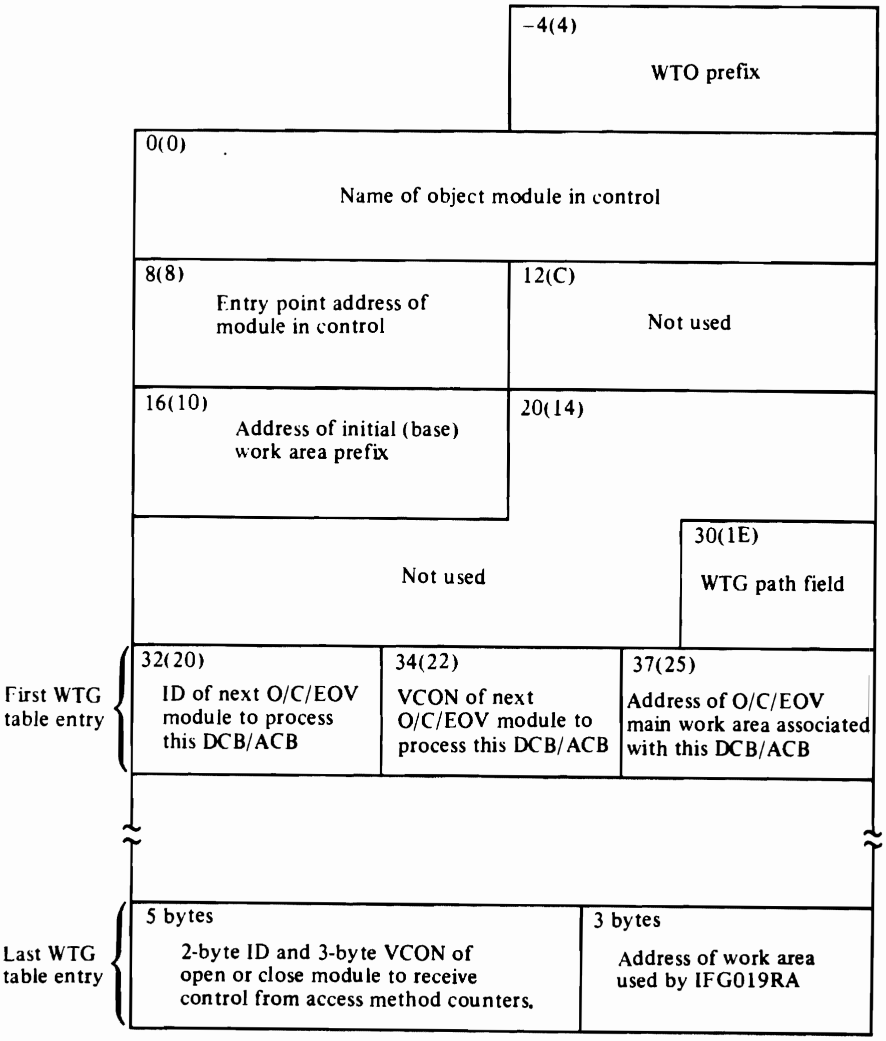
*If the object module is not in the same load module as module A, this field will contain zeros (an unresolved VCON).

Both forms of the XCTLTABL macro instruction generate V-type address constants (VCONs). These VCONs contain the entry point address of the object module with which the VCON is associated, if the object modules are in the same load module. If the object modules are in different load modules, the VCON will not be resolved and will contain zeros. When the IECRES LOAD macro is issued to transfer control to the service routine, IFG019RA checks to see if the VCON contains an address; if so, control is passed to the address in the VCON. If the VCON is unresolved, the service routine initiates a search of the link pack area for the entry point address of the object module to receive control.

If you are modifying the O/C/EOV routines, you should be aware that every unresolved VCON must have a defined entry point name or alias in the link pack area.

Where-to-go (WTG) Table

When an OPEN, CLOSE, or EOV macro is issued, one of the first data areas obtained by the O/C/EOV routines to process the request is a WTG table. This table is used to transfer control among the O/C/EOV and access method executor modules.



-4(4) WTO prefix

0(0) Name of object module in control

8(8) Entry point address of module in control

12(C) Not used

16(10) Address of initial (base) work area prefix

20(14) Not used

30(1E) WTG path field

First WTG table entry

32(20) ID of next O/C/EOV module to process this DCB/ACB

34(22) VCON of next O/C/EOV module to process this DCB/ACB

37(25) Address of O/C/EOV main work area associated with this DCB/ACB

Last WTG table entry

5 bytes 2-byte ID and 3-byte VCON of open or close module to receive control from access method counters.

3 bytes Address of work area used by IFG019RA

The WTG table is preceded by a 4-byte prefix used to specify message length and routing and descriptor codes for use when issuing WTO macros to trace the flow of control among the O/C/EOV modules (see "The Optional Module Listing Routine").

The first 8 bytes of the WTG table are used to contain an object module (CSECT) ID, which is used by the service routine, IFG019RA, to determine which routine is to receive control next. The next 4 bytes contain the entry point address of the module named in the preceding field. The address of the work area prefix associated with the request is at offset 16 (X'10'); this prefix is used to chain the work areas obtained to process the request (see "Data Area Layouts" earlier in this section).

Beginning at offset 32 (X'20'), the O/C/EOV initialization functions build an 8-byte entry for each DCB and ACB being processed as a result of the issuance of the OPEN, CLOSE, or EOV macro instruction. These entries are constructed in the same order as the Open or Close parameter list. Each of the 8-byte WTG table entries contains:

- The last 2 bytes of the module ID of the O/C/EOV module that is to receive control next to process the DCB or ACB represented by this WTG table entry. This module ID is selected and moved into the WTG table by the module that has completed processing the DCB or ACB.
- The 3-byte V-type address constant (VCON) of the next module to receive control. This VCON will be unresolved (contain zeros) unless the object module transferring control is in the same load module as the object module receiving control
- The 3-byte address of the O/C/EOV main work area used to process this DCB or ACB.

Each subsequent O/C/EOV module that receives control modifies the entries to again show the next module to receive control to process the DCB or ACB represented by each WTG table entry.

The O/C/EOV modules (but not the access method executors) transfer control by issuing the IECRES LOAD macro instruction. The access method executors use the XCTL macro to pass control. The IECRES macro instruction results in a branch to the O/C/EOV service routine, IFG019RA. Diagram 5.2 describes the processing done by the service routine in transferring control.

The last 8-byte WTG table entry contains a 2-byte module ID and a 3-byte V-type address constant (VCON) of the Open or Close module to which the access method executors return control, followed by the 3-byte address of the work area used by IFG019RA. Module names in this field will be IFG0196W or IFG0200Y or their aliases. This last entry of the WTG table is used only by the access method executors.

WTGPATH Field

The WTG table contains a field called WTGPATH. The settings of bits in this field are used to pass information between the O/C/EOV modules.

The first two columns of the following list show each byte of the WTGPATH field and each bit within the byte that is used. The third column shows which functions use each bit, and the fourth column tells what each bit indicates when set to 1.

Description of WTGPATH Field

Byte	Bit	Used by	Indicates
WTGPATH+0	1...	Open	Release of unused direct-access space was requested.
	.1..	Open	Nonstandard label tape input processing requires that a WTO be issued to tell the operator the data set name.
	..1.	EOV TCLOSE	NSL processing is deferred. NSL routine to be given control. routine.
	...1	Close, EOV	SMF data set activity information required.
	.... 1...	Open	JFCB has been modified and must be copied back to the SWA.
	.... .1..	Open	Additional device not ready; messages are suppressed for the 3270.
	xx		Reserved
WTGPATH+1	1...	Open, Close, EOV	Optional trace requested.
	.... .1..	Open, Close, TCLOSE, EOV	TIOT must be dequeued.
	.xxx x...		Reserved
	.... ..1.	Open	Tape volume with nonstandard labels must be remounted and checked.
	1	Open	Abnormal termination condition encountered in processing a DCB/ACB, requiring problem determination processing, but other DCB/ACBs in the same request must first be processed.

DIAGNOSTIC AIDS

This section presents message and system code cross-reference information, describes how to read and interpret the information about errors recorded by O/C/EOV routines, and tells how to activate the optional problem determination and diagnostic routines provided with the O/C/EOV component.

System Code-to-Module and Message Cross-References

You can use the cross-reference tables that follow, in conjunction with *OS/VS Message Library: VS2 System Messages* and *OS/VS Message Library: VS2 System Codes*, to determine which O/C/EOV module detected an error condition. You must know the system code and the return code (contents of register 15) in order to use the table. The system code and return code are included in the message text.

Following the system-code-to-module cross-reference table is a list of the internal codes passed to the O/C/EOV problem determination routines by the mainline O/C/EOV modules, the access method executors, and some of the SAM and BDAM processing routines. Abbreviated descriptions of the error condition associated with each internal ABEND code are shown in this table. Complete descriptions and additional information can be found in *OS/VS Message Library: VS2 System Messages.*

Problem Determination and Recovery Routine Diagnostics

This section describes the output of the O/C/EOV problem determination and recovery routines. It is intended for use in interpreting the information put in SYS1.LOGREC, a GTF data set, and system dumps by the O/C/EOV problem determination and recovery routines.

Optional Work Area Trace

This section tells how to activate the optional work area trace routine, which when activated, saves the contents of work areas and control blocks.

Module Listing Routine and Examples of Lists

This section tells how to activate the optional module listing routine. This routine prints the CSECT ID of each O/C/EOV module entered during the execution of an Open, Close, or End-of-Volume operation. Examples of the module lists put out by the routine are included.

System Code, Message, and Internal Code Cross-Reference

System Code-to-Module Cross-Reference

This table is intended to be used to find the O/C/EOV module (or access method module) that detected an abnormal termination condition. You must know the system code and return code (contents of register 15) associated with the condition. The system code and return code are included in dumps and in the message text. For example, in the message

IEC141I 013-BC, JJUOFPN5,,SYSIN

013 and BC are the system code and return code, respectively.

Also included in this table are the message ID, the O/C/EOV internal ABEND code, the name of the module that detected the condition, and a terse description of the function of the module.

All of the messages in the table that follows are issued by problem determination module IFG0199B. The text for these messages is contained in a table in the first load of the problem determination routines: IFG0190P for Open, IFG0200P for Close, IFG0232P for TCLOSE, IFG0550P for EOV, IGG0196M for RDJFCB and the open access method executors, or IGG0206M for the close access method executors.

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
002	04	IEC036I	253	IGG019AB	QSAM GET routine
				IGG019AD	QSAM GET routine
				IGG019AE	QSAM GET routine
				IGG019BN	QSAM GET-update routine
				IGG019BO	QSAM GET routine
				IGG019FB	QSAM GET routine
				IGG019FD	QSAM GET routine
				IGG019FF	QSAM GET routine
	08	IEC036I	254	IGG019CD	SAM EOB routine
				IGG019CV	SAM EOB routine
				IGG019C2	SAM EOB routine
				IGG019TD	SAM EOB routine
				IGG019TV	SAM EOB routine
				IGG019T2	SAM EOB routine
	0C	IEC036I	202	IGG019CD	SAM EOB routine
				IGG019CV	SAM EOB routine
				IFG019TD	SAM EOB routine
				IGG019TV	SAM EOB routine
	10	IEC036I	203	IGG019FG	SAM EOB routine
				IGG019FG	QSAM PUT routine
	14	IEC036I	204	IGG019C2	SAM EOB routine
				IGG019T2	SAM EOB routine

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
				IGG019AL	QSAM PUT routine
	18	IEC036I	213	IGG019AJ	QSAM PUT routine
				IGG019BP	QSAM PUT routine
				IGG019FJ	QSAM PUT routine
	1C	IEC036I	205	IGG019DA	BDAM create, format F
				IGG019DB	BDAM create, format U,V
	20	IEC036I	206	IGG019DD	BDAM create, format F, track overflow
	24	IEC036I	207	IGG019DD	BDAM create, format F, track overflow
	28	IEC036I	208	IGG0191M	BDAM create, open executor
005	04	IEC041I	200	IGG0196R	SAM open executor
	08	IEC041I	201	IGG0196R	SAM open executor
008	04	IEC037I	209	IGG019BS	BDAM create, check, format VS
				IGG019DC	BDAM create, check, format F,V,U
	08	IEC037I	210	IGG019BS	BDAM create, check, format VS
				IGG019DC	BDAM create, check, format F,V,U
013	04	IEC141I	17	IFG0194F	Open tape volume verification
	08	IEC141I	18	IFG0194F	Open tape, volume verification
	0C	IEC141I	51	IGG0193A	BDAM open executor
	10	IEC141I	52	IGG0191C	SAM open executor
	14	IEC141I	53	IGG0191B	SAM open executor
	18	IEC141I	54	IGG0191B	SAM open executor
	1C	IEC141I	55	IGG0191B	SAM open executor
	20	IEC141I	56	IGG0191A	SAM open executor
	24	IEC141I	57	IGG0191A	SAM open executor
				IGG0199G	SAM subsystem interface open executor (JES)
	28	IEC141I	58	IGG0191A	SAM open executor
				IGG0199G	SAM subsystem interface open executor (JES)
	2C	IEC141I	59	IGG0196J	SAM open executor
	30	IEC141I	63	IGG0191F	SAM open executor
	34	IEC141I	61	IGG0191I	SAM open executor
				IGG0199G	SAM subsystem interface open executor (JES)

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	38	IEC141I	62	IGG0191H	SAM open executor
				IGG0199K	SAM open executor
	3C	IEC141I	60	IGG0191D	SAM open executor
	40	IEC141I	64	IGG0191O	SAM open executor
				IGG0199O	SAM open executor
	44	IEC141I	65	IGG0191K	SAM open executor
				IGG0191W	SAM open executor
	48	IEC141I	66	IGG0191I	SAM open executor
	4C	IEC141I	72	IGG0196B	SAM open executor
				IGG0191A	SAM open executor
				IGG0199G	SAM subsystem interface open executor (JES)
	50	IEC141I	73	IGG0196B	SAM open executor
	54	IEC141I	74	IGG0196A	SAM open executor
	58	IEC141I	75	IGG01912	SAM open executor
	5C	IEC141I	76	IGG01915	SAM open executor
				IGG01916	SAM open executor
	60	IEC141I	83	IGG0191A	SAM open executor
	64	IEC141I	43	IFG0196J	Open merge, JFCB to DCB
				IFG0194A	Open, volume/unit selection
	68	IEC141I	44	IFG0196L	Open merge, DCB exit
	6C	IEC141I	46	IFG0196L	Open merge, DCB exit
	70	IEC141I	47	IFG0196M	Open merge, DCB to JFCB
	74	IEC141I	85	IGG0197A	SAM open executor
	78	IEC141I	86	IGG0197A	SAM open executor
	7C	IEC141I	87	IGG0197A	SAM open executor
	80	IEC141I	88	IGG0197A	SAM open executor
	84	IEC141I	84	IGG0197B	SAM open executor
	88	IEC141I	94	IGG0191B	SAM open executor
	90	IEC141I	113	IGG0197V	SAM open executor
	94	IEC141I	114	IGG0191A	SAM open executor
	98	IEC141I	115	IGG0191A	SAM open executor
	9C	IEC141I	116	IGG0197V	SAM open executor
	A0	IEC141I	117	IGG0191A	SAM open executor
	A4	IEC141I	106	IGG0199G	SAM subsystem interface open executor (JES)
	A8	IEC141I	107	IGG0199G	SAM subsystem interface open executor (JES)

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	B0	IEC141I	221	IGG0191A	SAM open executor
	B4	IEC141I	222	IGG0191A	SAM open executor
	B8	IEC141I	223	IGG0197V	SAM open executor
	BC	IEC141I	226	IGG0199G	SAM subsystem interface executor
	C0	IEC141I	227	IGG0199G	SAM subsystem interface executor
	C4	IEC141I	229	IGG0191L	BDAM open executor
	C8	IEC141I	252	IGG0193K	Subsystem open executor
	CC	IEC141I	118	IGG0196Q	SAM Open Executor for 3800
020	04	IEC038I	211	IGG0193A	BDAM open executor
	08	IEC038I	212	IGG0193A	BDAM open executor
03D	04	IEC156I	214	IFG0194C	Open, DA volume verification
	08	IEC156I	215	IFG0194C	Open, DA volume verification
113	08	IEC142I	32	IFG0194A	Open, volume/unit selection
	08	IEC142I	120	IGG0196Q	SAM Open Executor for 3800
	0C	IEC142I	34	IFG0193A	Open initialization
	18	IEC142I	50	IFG0194A	Open, volume/unit selection
117	04	IEC218I	131	IFG0232Z	TCLOSE final
	08	IEC218I	139	IFG0232S	TCLOSE, NL, SL tape
	10	IEC218I	127	IFG0232S	TCLOSE, SL output tape
			128	IFG0232S	TCLOSE, SL output tape
	14	IEC218I	137	IFG0232S	TCLOSE, SL output tape
	18	IEC218I	129	IFG0232S	TCLOSE, SL output tape
	1C	IEC218I	130	IFG0232S	TCLOSE, SL output tape
	20	IEC218I	134	IFG0232S	TCLOSE, SL output tape
	24	IEC218I	135	IFG0232S	TCLOSE, SL output tape

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	28	IEC218I	136	IFG0232S	TCLOSE, SL output tape
	2C	IEC218I	140	IFG0232G	TCLOSE, SL output tape
				IFG0232S	TCLOSE NL, SL tape positioning
				IFG0551R	EOV output tape, write EOV2 label
	30	IEC218I	141	IFG0232M	TCLOSE, SVL output tape
	34	IEC218I	233	IFG0232Z	TCLOSE, final
137	04	IEC022I	141	IFG0551P	EOV output tape write EOV1 label
				IFG0551R	EOV output tape, write EOV2 label
	08	IEC022I	146	IFG0552R	EOV input tape, position current volume
	0C	IEC022I	147	IFG0552R	EOV input tape, position current volume
	10	IEC022I	150	IFG0552R	EOV input tape, position current volume
				IFG0552T	EOV input tape, read user labels
	14	IEC022I	164	IFG0553F	EOV input tape, label processing, new volume
	18	IEC022I	191	IFG0553F	EOV input tape, label processing, new volume
	1C	IEC022I	180	IFG0552R	EOV input tape, position current volume
	20	IEC022I	195	IFG0553F	EOV input tape, label processing, new volume
	24	IEC022I	108	IFG0552B	EOV output tape, rewrite volume label, new volume

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
16E[1]	04			IFGDEBCK	DEB validity check
	08			IFGDEBCK	DEB validity check
	0C			IFGDEBCK	DEB validity check
	10			IFGDEBCK	DEB validity check
	14			IFGDEBCK	DEB validity check
	18			IFGDEBCK	DEB validity check
	1C			IFGDEBCK	DEB validity check
	20			IFGDEBCK	DEB validity check
213	04	IEC143I	30	IFG0195G	Open direct-access 213 ABEND exit function
				IFG0194C	Open, DA volume verification
	08	IEC143I	21	IFG0195T	Open/EOV password protection
	0C	IEC143I	216	IFG0194C	Open DA volume verification
	18	IEC143I	31	IFG0196X	Open, load EXCP appendages, move JFCB
				IFG0194C	Open DA volume verification
	20	IEC143I	230	IFG0194C	Open, DA volume verification
	24	IEC143I	231	IFG0194C	Open, DA volume verification
	28	IEC143I	246	IFG0194A	Open, volume/unit selection
	2C	IEC143I	242	IFG0194C	Open DA volume verification
214	04	IEC210I	64	IFG0202A	Close, SUL tape, read or write user labels
	08	IEC210I	65	IFG0202F	Close tape, volume disposition function
237	04	IEC023I	148	IFG0552R	EOV input tape, position current volume
	08	IEC023I	165	IFG0553F	EOV input tape, label processing, new volume

[1] This ABEND is issued by the DEB validity checking routine (IFGDEBCK), not the problem determination routines.

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
240	04	IEC155I		IGC0006D	RDJFCB macro instruction
	08	IEC155I	80	IGC0006D	RDJFCB macro instruction
	0C	IEC155I	81	IGC0006D	RDJFCB macro instruction
	10	IEC155I	82	IGC0006D	RDJFCB macro instruction
313	04	IEC144I	36	IFG0194C	Open DA, volume verification
314	04	IEC211I	86	IFG0200V	Close initialization, read JFCB
				IFG0200Y	Close access method executor return
	08	IEC211I	89	IFG0201R	Close, DA output, update F1 DSCB
317	04	IEC220I	132	IFG0232D	TCLOSE DA, read F1 DSCB
337	04	IEC024I	174	IFG0552X	EOV, EOD/concatenation
				IGC0005E	EOV, initialization
	08	IEC024I	252	IGG019AV	SAM dummy data set routine
413	04	IEC145I	26	IFG0194A	Open, volume/unit selection
			1	IFG0195D	Open tape, NSL input
				IFG0196O	Open tape, NSL output
	08	IEC145I	2	IFG0194F	Open tape, volume verification
	0C	IEC145I	3	IFG0194F	Open tape, volume verification
	10	IEC145I	4	IGC0002B	Open, TYPE=J
	18	IEC145I	8	IFG0194A	Open, volume/unit selection
	1C	IEC145I	37	IFG0193A	Open initialization, read JFCB
				IFG0194A	Open, volume/unit selection
	20	IEC145I	27	IFG0194C	Open DA, volume verification
	24	IEC145I	23	IFG0194F	Open tape, volume verification
	2C	IEC145I	111	IFG0194J	OPEN message routine
		IEC166I		IGG0191N	SAM open executor
				IGG0193A	BDAM open executor

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	30	IEC145I	112	IFG0194C	Open, direct-access volume verification
	34	IEC145I	253	IFG0194C	Open, direct-access volume verification
414	04	IEC212I	88	IFG0201R	Close, DA output, update F1 DSCB
417	04	IEC221I	133	IFG0232D	TCLOSE DA, write F1 DSCB
437	08	IEC025I	182	IFC0003A	FEOV initialization
	0C	IEC025I	199	IFG0553X	EOV, DA input, build new DEB
				IFG0555H	EOV, DA output, build new DEB
513	04	IEC146I	6	IFG0194A	Open, volume/unit selection
613	04	IEC147I	9	IFG0195B	Open, position SL tape
				IFG0195C	Open, position NL (BLP) tape
				IFG0195K	Open, SL input tape, verify HDR2 label
				IFG0195N	Open, SL input tape, read user labels
	08	IEC147I	11	IFG0195B	Open, position SL tape
				IFG0195H	Open, SL input tape, verify HDR1 label
				IFG0195K	Open, SL input tape, verify HDR2 label
				IFG0196N	Open, SL output tape, read existing HDR1 label
	0C	IEC147I	12	IFG0194A	Open, volume/unit selection
				IFG0194F	Open, tape volume verification
				IFG0195H	Open, SL input tape, verify HDR1 label
				IFG0195K	Open, SL input tape, verify HDR2 label
				IFG0196N	Open, SL output tape, read existing HDR1 label
	10	IEC147I	15	IFG0196Q	Open, SL output tape, check expiration date, write volume label
				IFG0196T	Open, SL output tape, write HDR1 and HDR2 labels

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	14	IEC147I	16	IFG0196T	Open, SL output tape, write HDR1 and HDR2 labels
614	04	IEC214I	90	IFG0202E	Close, DA final
	08	IEC214I	234	IFG0202E	Close, DA final
637	04	IEC026I	144	IFG0551T	EOV output tape, volume disposition
	08	IEC026I	149	IFG0552T	EOV input tape, read user labels, current volume
	0C	IEC026I	176	IFG0552X	EOV, EOD/concatenation
	10	IEC026I	194	IFG0553F	EOV input tape, label processing, new volume
	14	IEC026I	151	IFG0552V	EOV input tape, position current volume
	18	IEC026I	152	IFG0552V	EOV input tape, position current volume
	1C	IEC026I	153	IFG0552V	EOV input tape, position current volume
	24	IEC026I	155	IFG0554F	EOV tape, volume verification
				IFG0552V	EOV input tape, position current volume
	2C	IEC026I	158	IFG0554F	EOV tape, volume verification
	34	IEC026I	160	IFG0554F	EOV tape, volume verification
	38	IEC026I	161	IFG0554F	EOV tape, volume verification
	3C	IEC026I	162	IFG0553F	EOV input tape, label processing, new volume
	40	IEC026I	163	IFG0553F	EOV input tape, label processing, new volume
	44	IEC026I	192	IFG0554F	EOV tape, volume verification
	4C	IEC026I	196	IFGO553F	EOV input tape, label processing, new volume
713	04	IEC148I	13	IFG0195H	Open SL tape, verify HDR1 label
				IFG0196Q	Open, SL ouput tape, check expiration date, rewrite volume label
	08	IEC148I	28	IFG0194A	Open, volume/unit selection
714	04	IEC215I	61	IFG0200Z	Close; SL tape, write TRL1 and TRL2 labels
	08	IEC215I	63	IFG0202A	Close, SUL tape, read or write user labels

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
				IFG0202F	Close, tape volume disposition
	0C	IEC215I	62	IFG0200Z	Close SUL tape, read or write user labels
				IFG0202F	Close, tape volume disposition
717	04	IEC222I	122	IFG0232G	TCLOSE SL output tape, write EOF1 label
	08	IEC222I	123	IFG0232G	TCLOSE SL output tape, write EOF1 label
	08	IEC222I	124	IFG0232M	TCLOSE SUL output tape, write EOF2 and user labels
	0C	IEC222I	125	IFG0232S	TCLOSE, NL, SL tape positioning
			126	IFG0232S	TCLOSE, NL, SL tape positioning
	10	IEC222I	138	IFG0232S	TCLOSE, NL, SL tape positioning
737	04	IEC027I	190	IFG0554C	EOV, DA volume verification
				IFG0554N	FEOV, DA output
				IFG0554P	EOV, DA output, get more space
				IFG0554T	EOV, DA output, B37 ABEND
				IFG0554Z	EOV, DA output, update F1 DSCB, current volume
	08	IEC027I	189	IFG0554C	EOV, DA volume verification
	0C	IEC027I	171	IFG0194C	EOV, DA volume verification
				IFG0554C	EOV, DA volume verification
				IFG0553Z	EOV DA input final
	10	IEC027I	187	IFG0554N	FEOV, DA output
	14	IEC027I	175	IFG0553P	EOV, DA input, read F1 DSCB
	1C	IEC027I	178	IFG0554C	EOV, DA volume verification
	24	IEC027I	218	IFG0553Z	EOV, DA input, final
	28	IEC027I	250	IFG0554C	EOV, DA volume verification
	2C	IEC027I	232	IFG0554N	FEOV, DA output
	34	IEC027I	111	IFG0553X	EOV direct-access input build new DEB

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
		IEC066I		IFG0555H	EOV direct-access output build new DEB
	38	IEC027I IEC066I	112	IFG0554C	EOV direct-access volume verification
	3C	IEC027I	113	IFG0554C	EOV, DA volume verification
813	04	IEC149I	14	IFG0195H	Open, SL input tape, verify HDR1 label
837	08	IEC028I	197	IFG0554A	EOV, volume/unit selection
				IFG0551H	EOV, determine device type
	0C	IEC028I	248	IFG0554A	EOV, volume/unit selection
913	04	IEC150I	19	IFG0194F	Open, tape volume verification
	08	IEC150I	20	IFG0195K	Open SL tape, verify HDR2 label
				IFG0196N	Open, SL output tape, read existing HDR1 label
	0C	IEC150I	22	IFG0195U	Password protection, udpate password record
	10	IEC150I	24	IFG0196L	Open merge, DCB exit
	14	IEC150I	41	IFG0194C	Open, DA volume verification
	18	IEC150I	139	IGC0002B	Open, type=J
	1C	IEC150I	140	IGC0002B	Open, type=J
	20	IEC150I	228	IFG0196X	Open, load EXCP appendages, copy JFCB
	28	IEC150I	239	IFG0194C	Open, DA volume verification
				IFG0196V	Open, gives control to executor
	2C	IEC150I	243	IFG0194C	Open, DA volume verification
	30	IEC150I	244	IFG0194J	Open, volume mount message
				IFG0196N	Open, SL output tape, reading existing HDR1 label
	38	IEC150I	141	IFG0194C	OPEN, DA volume verification
				IFG0193E	OPEN, create standard tape label
				IFG0194F	OPEN, tape volume verification
	3C	IEC150I	142	IFG0194C	OPEN, tape volume verification

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	40	IEC150I	143	IFG0194C	OPEN, tape volume verification
	44	IEC150I	245	IFG0196T	OPEN, SL output tape, write HDR1 and HDR2 labels
	48	IEC150I	247	IFG0194F	OPEN, SL output tape, write HDR1 and HDR2 labels
	4C	IEC150I	248	IFG0194F	OPEN, SL output tape, write HDR1 and HDR2 labels
	50	IEC150I	249	IFG0196T	OPEN, SL output tape, write HDR1 and HDR2 labels
	54	IEC150I	250	IFG0194F	OPEN, SL output tape, write HDR1 and HDR2 labels
	58	IEC150I	251	IFG0194F	OPEN, SL output tape, write HDR1 and HDR2 labels
	5C	IEC150I	252	IFG0194F	OPEN, SL output tape, write HDR1 and HDR2 labels
937	10	IEC029I	240	IFG0193D	EOV, destroy standard tape label verification
				IFG0552B	EOV, tape output, rewrite volume label, new volume
				IFG0553F	EOV, tape input, label processing, new volume
	14	IEC029I	241	IFG0554C	EOV, DA volume verification
				IFG0553F	EOV, tape input, label processing, new volume
	18	IEC029I	242	IGC0003A	FEOV, initialization
	1C	IEC029I	245	IFG0554B	EOV, DA, repositioning records that span volumes
	20	IEC029I	249	IFG0553H	EOV, tape input, label processing, new volume
	24	IEC029I	109	IFG0552B	EOV, tape output, rewrite volume label, new volume
	28	IEC029I	1	IFG0554C	EOV,SL output tape, write HDR1 and HDR2 labels
				IFG0553E	EOV, SL output tape, write HDR1 and HDR2 labels
				IFG0554F	EOV, SL output tape, write HDR1 and HDR2 labels
	2C	IEC029I	2	IFG0554C	EOV, SL output tape, write HDR1 and HDR2 labels
	30	IEC029I	3	IFG0554C	EOV, SL output tape, write HDR1 and HDR2 labels

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	34	IEC029I	4	IFG0554C	EOV, SL output tape, write HDR1 and HDR2 labels
	38	IEC029I	5	IFG0554P	EOV, DA input, read F1 DSCB
	3C	IEC029I	6	IFG0554C	EOV, DA input, read F1 DSCB
				IFG0554F	EOV, DA input, read F1 DSCB
	40	IEC029I	7	IFG0553E	EOV, DA input, read F1 DSCB
				IFG0554F	EOV, DA input, read F1 DSCB
				IFG0552B	EOV,tape output, rewrite volume label, new volume
				IFG0554C	EOV, tape output, rewrite volume label, new volume
A13	04	IEC151I	10	IFG0195B	Open,position SL tape
				IFG0195C	Open, position NL (BLP) tape
	08		152	IFG0195B	Open, position SL tape
	0C		153	IFG0195B	Open, position SL tape
	10		154	IFG0195B	Open, Position SL tape
	14		155	IFG0195H	Open, SL input tape verify HDR1 label
	18		156	IFG0195B	Open, position SL tape
A14	04	IEC215I	87	IFG0202E	Close DA final
A37	04	IEC015I	198	IGC0005E	EOV, initialization
	08	IEC015I	219	IFG0551F	EOV,read JFCB
				IFG0005E	EOV,initialization
B13	04	IEC152I	71	IGG0191U	SAM open executor
				IGG0196R	SAM open executor
	08	IEC152I	70	IGG0191U	SAM open executor
	0C	IEC152I	69	IGG0191V	SAM open executor
				IGG0196R	SAM open executor
	10	IEC152I	67	IGG0197U	SAM open executor
	14	IEC152I	68	IGG0197U	SAM open executor
	18	IEC152I	89	IGG0197F	SAM open executor
	1C	IEC152I	90	IGG0197F	SAM open executor
	20	IEC152I	95	IGG0191T	SAM open executor
				IGG0196R	SAM open executor
	24	IEC152I	96	IGG0191T	SAM open executor
				IGG0196R	SAM open executor
	28	IEC152I	97	IGG0191T	SAM open executor

System Code-to-Module Cross-Reference

System Code	Return Code	Message ID	Internal Code	Module Detecting Error	Function of Module
	2C	IEC152I	101	IGG0197E	SAM open executor
	30	IEC152I	102	IGG0197E	SAM open executor
	34	IEC152I	103	IGG0197E	SAM open executor
B14	04	IEC217I	121	IGG0201B	SAM close executor
		IEC217I		IGG0201Z	SAM close executor
	08	IEC217I	122	IGG0201B	SAM close executor
		IEC217I		IGG0201Z	SAM close executor
	0C	IEC217I	123	IGG0201B	SAM close executor
				IGG0201Z	SAM close executor
	10	IEC217I	124	IGG0201B	SAM close executor
				IGG0201Z	SAM close executor
	14	IEC217I	125	IGG0201B	SAM close executor
				IGG0201Z	SAM close executor
	18	IEC217I	126	IGG0201B	SAM close executor
				IGG0201Z	SAM close executor
B37	04	IEC030I	142	IFG0554A	EOV, volume/unit selection
			156	IFG0554A	EOV, volume/unit selection
			168	IFG0554A	EOV, volume/unit selection
			185	IFG0554T	EOV, DA output, B37 ABEND
	08	IEC030I	177	IFG0554P	EOV, DA output, get more space
	0C	IEC030I	247	IFG0554A	EOV, volume/unit selection
C13	04	IEC157I	91	IGG01934	GAM executor
	10	IEC153I	42	IFG0194A	Open, volume/unit selection
C37	04	IEC033I	230	IFG0194C	EOV, DA volume verification
	08	IEC033I	231	IFG0194C	EOV, DA volume verification
D13	04	IEC158I	92	IGG01934	GAM executor
D37	04	IEC031I	186	IFG0554P	EOV DA output, get more space
E13	04	IEC159I	93	IGG01934	GAM executor
E37	04	IEC032I	184	IFG0554P	EOV DA output, get more space
	08	21	183	IFG0554P	EOV DA output, get more space

Internal ABEND Codes

When an abnormal termination condition arises during an Open, Close, or EOV operation, the module detecting the condition passes the O/C/EOV problem determination function an internal code using the DMABCOND macro. This internal code indicates which system code, return code, and message are to be issued.

The following tables are divided into the Open, Close, Close (TYPE=T), and EOV operations with the internal codes that each passes listed under it. The tables give the internal code, its associated ABEND code and return code, and an abbreviated description of the reason for the error. The *OS/VS Message Library: VS2 System Messages* and *OS/VS Message Library: VS2 System Codes* provide complete descriptions of the error conditions, along with system actions, programmer responses, and the sources of diagnostic and problem determination information. These tables may be used in conjunction with the preceding cross-reference table to find the name and function of the module that issued the internal code.

Open Codes

Internal Code	ABEND Code	Return Code	Reason
1	413	04	No unit available to mount volume, or volume inproperly specified
2	413	08	I/O error during positioning of tape volume
3	413	0C	I/O error during reading of tape volume label
4	413	10	I/O error during writing of tapemark
6	513	04	Second OPEN was issued for a data set on a magnetic tape volume
8	413	18	Data set was opened for input and no volume serial number was specified
9	613	04	I/O error during positioning of tape volume
10	A13	04	I/O error due to incorrect file sequence number during positioning of tape volume
11	613	08	I/O error during reading of tape label
12	613	0C	Invalid tape label was read
13	713	04	Expiration date on magnetic tape data set has not passed
14	813	04	DSNAME on header label does not match that on DD statement
15	613	10	I/O error during writing of tape label
16	613	14	I/O error during writing of tapemark following header labels
17	013	04	American National Standard labels specified but not specified at system generation.
18	013	08	American National Standard labels specified for 7-track tape drive
19	913	04	ASCII volume accessibility byte not blank
20	913	08	ASCII security byte in HDR1 label indicates password-protected volume
21	213	08	Format-1 DSCB for PASSWORD data set not found

Open Codes

Internal Code	ABEND Code	Return Code	Reason
22	913	0C	Invalid password supplied by operator
23	413	24	800 BPI specified and data set allocated to 1600 BPI drive, or 1600 BPI specified and data set allocated to 800 BPI drive
24	913	10	Unauthorized job step attempted to open a VTOC for output
26	413	04	Open was not able to demount volume and mount required volume
27	413	20	I/O error during reading of volume label on direct-access device
28	713	08	Expiration date on direct-access data set has not passed
30	213	04	DSCB not found on first volume of data set
31	213	18	I/O error during writing back of identifier (format-1) DSCB
32	113	08	No JFCB extension found, PS data set
34	113	0C	OPEN TYPE=J issued but no JFCB address specified
36	313	04	I/O error during reading of index (format-2) or extension (format-3) DSCB
37	413	1C	Volume sequence number is greater than volume count
41	913	14	Checkpoint and non-checkpoint data sets concatenated
42	C13	10	Illegal concatenation of PO data sets for output specified
43	013	64	Dummy (null) data set specified but (1) invalid for EXCP or (2) DSORG did not specify BSAM or QSAM
44	013	68	Blocksize is greater than 32760
46	013	6C	Allocated unit does not support track overflow
47	013	70	Magnetic tape volume has mixed EBCDIC and ASCII attributes, or DCBOPTCD=Q specified for a nontape device, or DCBOPTCD indicates ASCII translation and/or user totaling for a spooled data set
50	113	18	No JFCB extension found, ISAM or BDAM data set
51	013	0C	BUFL of 0 specified, dynamic buffering requested
52	013	10	For QSAM dummy data set using PUT LOCATE; DCBBUFCB=0, DCBBUFL=0, and DCBBLKSI=0
53	013	14	DCB specified PO data set but DSCB specified different data set organization
54	013	18	Member of PO data set not found during execution of BLDL instruction
55	013	1C	I/O error during searching of directory of PO data set

Open Codes

Internal Code	ABEND Code	Return Code	Reason
56	013	20	QSAM blocksize not a multiple of logical record length for blocked data set with fixed-length records, or for variable-length records, blocksize not four (4) greater than logical record length
57	013	24	OPEN option 1 conflicts with MACRF parameter in DCB or DCB or ACB
58	013	28	OPEN option 1 conflicts with MACRF parameter in DCB or ACB
59	013	2C	No buffers available
60	013	3C	No buffers available
61	013	34	Buffers cannot be obtained because no blocksize or buffer length is specified, or logical record length is zero for a spooled QSAM data set not specifying V-type records, or blocksize is zero for a spooled BSAM data set
62	013	38	No buffers available
63	013	30	No buffers available
64	013	40	No buffers available
65	013	44	No buffers available
66	013	48	No buffers available
67	B13	10	I/O error during verifying of UCS print chain
68	B13	14	Requested print chain not available
69	B13	0C	I/O error during loading of UCS buffers or permanent I/O error loading the format record into the IBM 3886 Optical Character Reader
70	B13	08	Permanent I/O error when BLDL was issued to locate a UCS image in SYS1.IMAGELIB
71	B13	04	UCS image not found in SYS1.SVCLIB by BLDL instruction or format record identifier specified for the IBM 3886 Optical Character Reader was not found in the SYS1.IMAGELIB
72	013	4C	Buffer length specified is smaller than blocksize
73	013	50	OUTPUT not specified in OPEN macro for printer data set
74	013	54	Secondary control unit for an IBM 1419 Magnetic Character Reader not found
75	013	58	Concatenation of unlike attributes was specified for data set on paper tape
76	013	5C	QSAM spanned variable-length records not in locate mode
79	240	04	RDJFCB issued but no foundation block present in DCB or ACB
80	240	08	RDJFCB issued but no exit list address specified in DCB or ACB
81	240	0C	RDJFCB issued but no JFCB exit specified
82	240	10	RDJFCB issued but exit list address is invalid
83	013	60	QSAM DCB specified RECFM=F, but logical record length not equal to blocksize

Open Codes

Internal Code	ABEND Code	Return Code	Reason
84	013	84	Number of buffers in buffer pool control block not equal to BUFNO in DCB
85	013	74	Optical character recognition (OCR) data set not open for
86	013	78	OCR data set opened with BUFL=0
87	013	7C	OCR data set opened with LRECL=0
88	013	80	OCR data set opened with buffer length less than logical record length
89	B13	18	I/O error during FCB image verification
90	B13	1C	Operator replied 'CANCEL' to message IEC1281
91	C13	04	Graphics device already open for this DCB
92	D13	04	DCB specifies GAM and requested unit not graphics device
93	E13	04	DCBGNCP field exceeds limits
94	013	88	QSAM specified in error for a communications device
95	B13	20	Storage not available for SYS1.IMAGELIB or unable to load the IBM 3886 OCR format record from SYS1.IMAGELIB (either the volume was not mounted, or a permanent I/O error occurred while reading from SYS1.IMAGELIB)
96	B13	24	Volume containing SYS1.IMAGELIB was not mounted or format record identifier was not specified in the DCB or DD statement for the IBM 3886 OCR data set
97	B13	28	I/O error encountered during search of SYS1.IMAGELIB
100	none	none	Failure to block or unblock results in data checks on a printer with UCS
101	B13	2C	I/O error encountered during search of SYS1.IMAGELIB
102	B13	30	I/O error encountered during loading of forms control buffer (FCB)
103	B13	34	Forms control buffer (FCB) image not found in SYS1.IMAGELIB
104	none	none	User totalling: (1) no DCB exit list specified, or (2) no block-count-unequal exit address specified, or (3) image area address is zero.
106	013	A4	DCBDSORG does not indicate PS (physical sequential) for a spooled data set
107	013	A8	DCBRECFM indicates VS (variable spanned) records for a spooled input data set
108	137	24	Expiration date not passed on current data set (first data set on the volume); EOV reached for output data set on magnetic tape
109	937	24	Protection mode indicated (password-protected data set) if JFCB did not agree with the protection mode in the HDR1 label of the existing data set (first data set on volume). EOV reached for output data set on magnetic tape.

Open Codes

Internal Code	ABEND Code	Return Code	Reason
111	413	2C	MSS failure. See *OS/VS Message Library: VS2 System Messages* for reason codes.
112	413	30	MSS failure. A MSS volume other than the one requested was mounted.
113	013	90	Data set not open for input
114	013	94	BUFL or BLKSIZE was less than 16 times the specified LRECL, IBM 3890 MICR device
115	013	98	RECFM did not specify F or FB
116	013	9C	MACRF was not GM or DSORG was not PS, IBM 3890 MICR device
117	013	A0	LRECL was not specified, was less than 12, or was greater than 48, IBM 3890 MICR device
139	913	18	JFCB was modified to indicate BLP, but task was not authorized
140	913	1C	ENQ issued on data set name; resource not available
141	913	38	An unauthorized user tried to open a RACF-protected data set or tape volume.
142	913	3C	Inconsistent data set definition: a multivolume data set is being opened. The data set is not RACF-protected, and the data set's format-1 DSCB on one of its volumes indicates the data set is RACF-protected.
143	913	40	A VSAM data space being opened is RACF-protected.
152	A13	08	FSEQ before start of volume while positioning to start of data set
153	A13	0C	FSEQ before start of volume while positioning to end of data set
154	A13	10	Tape mark read instead of HDR1 during forward space file positioning
155	A13	14	Tape mark read instead of HDR1 of desired file
156	A13	18	EOV1 read on last or only volume
214	03D	04	Volumes not specified in order they were created
215	03D	08	ISAM data set does not have required index (format-2) DSCB
216	213	0C	I/O error during reading of DSCB of ISAM or BDAM data set
221	013	B0	Incompatible DCB and OPEN operands: RDBACK (tape) with variable-spanned records
222	013	B4	Incompatible DCB and OPEN operands: INOUT/OUTIN with QSAM MACRF values
223	013	B8	DCBHDR operand was omitted from DCB, IBM 3890 MICR device
226	013	BC	SYSIN/SYSOUT data set, (1) QSAM data set opened for OUTPUT or RDBACK or (2) BSAM data set with POINT macro specified. Neither is permitted.

Open Codes

Internal Code	ABEND Code	Return Code	Reason
227	013	C0	SYSIN/SYSOUT data set could not be opened
228	913	20	EXCP appendages requested; neither the program loading the appendage nor the appendages being loaded were authorized. (Authorized appendages are listed in the IEAAPP00 member of SYS1.PARMLIB).
229	013	C4	BDAM data set being created, but DCB MACRF field specified GET or READ. Only WL (WRITE-LOAD) allowed.
230	213	20	Second or subsequent volume of a DA input data set contained more than 16 extents (allocated under DOS)
231	213	24	Second or subsequent volume of a DA input data set contained an incompatible split-cylinder data set (allocated under DOS)
239	913	28	Unauthorized user tried to open a checkpoint data set with DSORG not PS or PO
242	213	2C	Format-1 DSCB not found. Installation exit module encountered an error.
243	913	2C	OPEN, TYPE=J issued for an ISAM data set. Data set names coded in the DD statements were not all the same or the JFCB had a different data set name than the DD statement.
244	913	30	OPEN issued to write a data set on a tape volume containing an existing data set. The security level (password protection or protection mode) of the new data set is different from the existing data set.
245	913	44	Invalid specification of PROTECT. PROTECT specified for a RACF-protected tape.
246	213	28	A volume was requested that contained 127 open data sets
247	913	48	Invalid specification of PROTECT. System tape option not in effect.
248	913	4C	Invalid specification of PROTECT. PROTECT specified but user not defined to RACF.
249	913	50	Invalid specification of PROTECT. PROTECT specified but request is for an NL tape.
250	913	54	Invalid specification of PROTECT. PROTECT specified but request is for INPUT or INOUT.
251	913	58	Invalid specification of PROTECT. PROTECT specified but volume sequence number or file sequence number not equal to one on a tape with LABEL=SL, SUL, AL, or AUL.
252	913	5C	Invalid specification of PROTECT. PROTECT specified for volume being used in a PUBLIC volume. This is a common OPEN internal abend code, processed by IFG0190P.
252	013	C8	Error encountered as a result of issuing the IEFSSREQ macro instruction. This is an OPEN executor internal abend code, processed by IGG0196M.

Open Codes

Internal Code	ABEND Code	Return Code	Reason
253	413	34	Error encountered as a result of issuing the IEFSSREQ macro instruction during OPEN, DA volume verification.

Close Codes

Internal Code	ABEND Code	Return Code	Reason
61	714	04	I/O error during writing of TRL1
62	714	0C	I/O error during writing of tapemark
63	714	08	I/O error during writing of TRL2
64	214	04	I/O error during reading of user label on tape volume
65	214	08	I/O error during positioning of tape volume
86	314	04	I/O error during reading of DSCB
87	A14	04	I/O error during partial release of space
88	414	04	I/O error during writing of DSCB
89	314	08	I/O error during reading of identifier (format-1) DSCB when SUL is specified
90	614	04	I/O error during writing of file mark
112	None	None	Invalid control block(s) found
121	B14	04	Duplicate name of partitioned data set found in directory
122	B14	08	Specified member of partitioned data set cannot be found
123	B14	0C	No space left in directory of partitioned data set
124	B14	10	I/O error during updating of directory of partitioned data set
125	B14	14	During STOW processing, it was determined that the specified DCB was not opened correctly
126	B14	18	Conditional GETMAIN failed during STOW processing
234	614	08	DCBFDAD greater than DEB extent encountered while attempting to write file mark (DA output data set)

Close (TYPE=T) Codes

Internal Code	ABEND Code	Return Code	Reason
122	717	04	I/O error during writing of tapemark following last data record
123	717	08	I/O error during writing of trailer label 1
124	717	08	I/O error during writing of TRL2 label
125	717	0C	I/O error during writing of first tapemark after trailer labels (SL) or after data (NL)

Internal Code	ABEND Code	Return Code	Reason
126	717	0C	I/O error during writing of second tapemark after trailer labels (SL) or after data (NL)
127	117	10	I/O error during backspace of file past second tapemark at end of data set
128	117	10	I/O error during backspace of file past first tapemark at end of data set
129	117	18	I/O error during forward space of file past tapemark preceding data for SL data set opened for RDBACK
130	117	1C	I/O error during forward space of file past tapemark preceding data for NL data set opened for RDBACK
131	117	04	I/O error during writing of file mark
132	317	04	I/O error during reading of format-1 DSCB
133	417	04	I/O error during writing of updated format-1 DSCB
134	117	20	I/O error during positioning of tape volume for NL INPUT or SL OUTPUT data set
135	117	24	I/O error during forward space of file past tapemark following data
136	117	28	I/O error during backspace of file past tapemark following data for data set opened for INPUT and LEAVE
137	117	14	I/O error during rewind or backspace past data when RDBACK with LEAVE or INOUT with REREAD are specified
138	717	10	I/O error during reading of TRL1 label to updated block count in DCB or ACB
139	117	08	I/O error during forward space of file past tapemark when RDBACK and LEAVE or INOUT and REREAD are specified
140	117	2C	I/O error during backspace of file past tapemark following data when volume is full and FEOV is to be issued
141	117	30	I/O error during tape volume positioning following user trailer label processing
142	None	00	DCB DSORG was not PS or PO
143	None	04	DCB was open for input to a a member of a partitioned data set
144	None	08	LEAVE was specified for an invalid data set orgainzation
145	None	0C	OUTPUT processing was requested for an invalid data set organization
151	None	10	REREAD was specified for an invalid DSORG
233	117	34	Extend in DCBFDAD refers to an extent that does not exist

EOV Codes

Internal Code	ABEND Code	Return Code	Reason
1	937	28	EOV processing determined that the caller is not authorized to access the data set or the tape volume on the new volume.
2	937	2C	Inconsistent data set or tape volume definition: a previously allocated multivolume data set is being processed for output. The data set's first volume is RACF-protected; the current volume is not RACF-protected.
3	937	30	EOV processing determined that the data set's new volume is RACF-protected (the format-1 DSCB indicates this), but the data set is not RACF-protected.
4	937	34	The VSAM data space requested at EOV is RACF-protected.
5	937	38	RACF processing failed to add a volume to an existing RACF-protected DASD data set when it extended the data set to a new volume.
6	937	3C	Inconsistent data set definition: a previously allocated multivolume data set is being processing for output. The data set's first volume is not RACF-protected; the current volume is RACF-protected.
7	937	40	Inconsistent RACF definition. Either a previously allocated multivolume DASD data set or a tape data set is being processed for output. The current volume being processed at EOV is defined to RACF but not as part of the same volume set as the previous volume.
108	137	24	Expiration date not passed on current data set (first data set on the volume). EOV reached for output data set on magnetic tape.
109	937	24	Protection mode indicated (password-protected data set) if JFCB did not agree with the protection mode in the HDR1 label of the existing data set (first data set on volume). EOV reached for output data set on magnetic tape.
111	737	34	MSS failure. See *OS/VS Message Library: VS2 System Messages* for reason codes
112	737	38	MSS failure. A MSS volume other than the one requested was mounted
113	737	3C	Format-1 DSCB not found. Installation exit module encountered an error.
141	137	04	I/O error tapemark
142	B37	04	Next volume cannot be mounted because no unit is available
144	637	04	I/O error during reading of label, writing of tapemark, or positioning of tape volume
146	137	08	I/O error during positioning of tape volume in preparation for label processing
147	137	0C	I/O error during reading of trailer label (header label if data set opened for RDBACK)
148	237	04	Block count in DCB does not match count in trailer label

Internal Code	ABEND Code	Return Code	Reason
149	637	08	I/O error during positioning of tape volume following user trailer label processing
150	137	10	I/O error during positioning of tape volume to logical end of data set following trailer label processing
151	637	14	I/O error during positioning of tape volume for OPEN LEAVE or FEOV LEAVE
152	637	18	I/O error during positioning of tape volume for OPEN REREAD
153	637	1C	I/O error during positioning of tape volume when DISP=PASS and no OPEN option 2 specified
155	637	24	I/O error during performing of rewind of tape volume during volume positioning
156	B37	04	Next volume cannot be mounted because no unit available
158	637	2C	I/O error during rewinding of next volume before verification of volume label
160	637	34	I/O error during reading of volume label of tape volume
161	637	38	I/O error during positioning of NL or NSL tape
162	637	3C	I/O error during positioning of tape volume to a concatenated data set
163	637	40	I/O error in positioning tape to be read forward at label (if SL volume) or at data (if NL volume)
164	137	14	I/O error during reading of header label or trailer label during opening for RDBACK
165	237	08	DSNAME in tape label does not match that in JFCB
168	B37	04	Next volume cannot be mounted because no unit available
171	737	0C	I/O error during reading of DSCB for concatenated partitioned data set
174	337	04	No EODAD routine had been specified in DCB when end-of-data detected
175	737	14	I/O error during reading of DSCB for user trailer label processing
176	637	0C	Concatenation of data sets with unlike attributes found but not specified in DCB
177	B37	08	Unable to convert VTOC during EOV secondary allocation
178	737	1C	I/O error during reading of extension (format-3) DSCB
180	137	1C	Invalid trailer label read
182	437	08	FEOV issued with invalid DCB
183	E37	08	DA output data set; no space on the next specified volume or duplicate data set found.
184	E37	04	No more space available on volume, all volumes used
185	B37	04	Next volume cannot be mounted, no unit available
186	D37	04	Primary space exhausted, no secondary space specified
187	737	10	I/O error during writing of file mark

Internal Code	ABEND Code	Return Code	Reason
189	737	08	I/O error during reading of volume label
190	737	04	I/O error during reading or writing of DSCB
191	137	18	I/O error during positioning of SL tape to first data record
192	637	44	I/O error during reading of sense bytes to check for file protect ring on data set opened for INOUT
194	637	10	I/O error in positioning tape volum when data set opened for RDBACK
195	137	20	Invalid header label read
196	637	4C	I/O error during positioning of tape to be read backward at label (if volume is SL) or at data (if volume is NL)
197	837	08	New JFCB extension block not found
198	A37	04	EOV (SVC 55) issued against a DCB that was not open
199	437	0C	DEB not in DEB table
200	005	04	Required address not specified, IBM 3886 OCR device
201	005	08	RBL READ not specified, IBM 3886 OCR device
202	002	0C	Record or block length greater than DA device track capacity during QSAM PUT or BSAM WRITE
203	002	10	RDW of variable-length record specified more than 32,752 bytes during QSAM PUT operation
204	002	14	Record length greater than blocksize, during QSAM PUT operation
205	002	1C	Record length greater than track capacity encountered during BDAM create operation
206	002	20	Block larger than secondary space quantity specification encountered during BDAM create operation
207	002	24	Block larger than primary quantity specification encountered during BDAM create operation
208	002	28	Processing error in BSAM or QSAM, or error during ISAM open or close
209	008	04	Unable to allocate more space on current volume during BDAM create operation
210	008	08	Permanant I/O error during BDAM create
211	020	04	BDAM open error
212	020	08	BDAM open error
213	002	18	Invalid record encountered during QSAM PUT operation
218	737	24	Missing member name detected during BLDL while searching for TTR of a concatenated member
219	A37	08	DEB does not point to DCB.
230	C37	04	Second or subsequent volume of a DA input data set contained more than 16 extents (DOS allocation)
231	C37	08	Second or subsequent volume of a DA input data set contained an incompatible split-cylinder data set (allocated under DOS)

Internal Code	ABEND Code	Return Code	Reason
232	737	2C	File mark could not be written during EOV processing; DCBFDAD extent number too big
240	937	10	Second or subsequent volume of checkpoint data set was not password protected
241	937	14	Second or subsequent volume of a non-checkpoint data set was found, during EOV processing, to be a password-protected, checkpoint data set
242	937	18	Checkpoint security interface established; FEOV macro cannot be issued
245	937	1C	DSCB does not match the old DEB, which is used during repositioning by FEOV
247	B37	0C	A volume was requested that contained 127 open data sets
248	837	0C	A second OPEN macro was issued for a data set on a tape volume
249	937	20	ASCII volume accessability byte not blank
250	737	28	Error encountered as a result of issuing the IEFSSREQ macro instruction during EOV, DA volume verification or EOV, DA output, get more space.
252	337	08	No EODAD specified in the DCB for a dummy data set
253	002	04	Record with RDW less than 4 encountered during QSAM GET operation
254	002	08	Record or block length greater than 32,767 encountered during QSAM PUT or BSAM WRITE operation

Register Usage

The use of all registers is consistent throughout DCB or ACB processing. The following list indicates for each register the register number, its equated name(s), and its use. Where a register has more than one equated name, the use depends on the equated name.

The contents of register 0 and 1 and 9 through 14 are saved and restored by the service routine, IFG019RA.

Register	Name	Use
0	R0	Work register and parameter register
1	R1	Work register and parameter register
2	RDCB	Address of current DCB or ACB
3	RBASE	Base register to be modified on entry to a module
4	RCORE	Current work area address
5	RES	Address of the common routine
	RPAR	Address of copy of parameter list in protected storage.
6	RWTG	Address of WTG table
	R6	Work register and parameter register
7	RPARC	Address of current entry in copy of parameter list
	R7	Work register and parameter register
8	RWTGC	Address of current entry in WTG table
	R8	Work register and parameter register

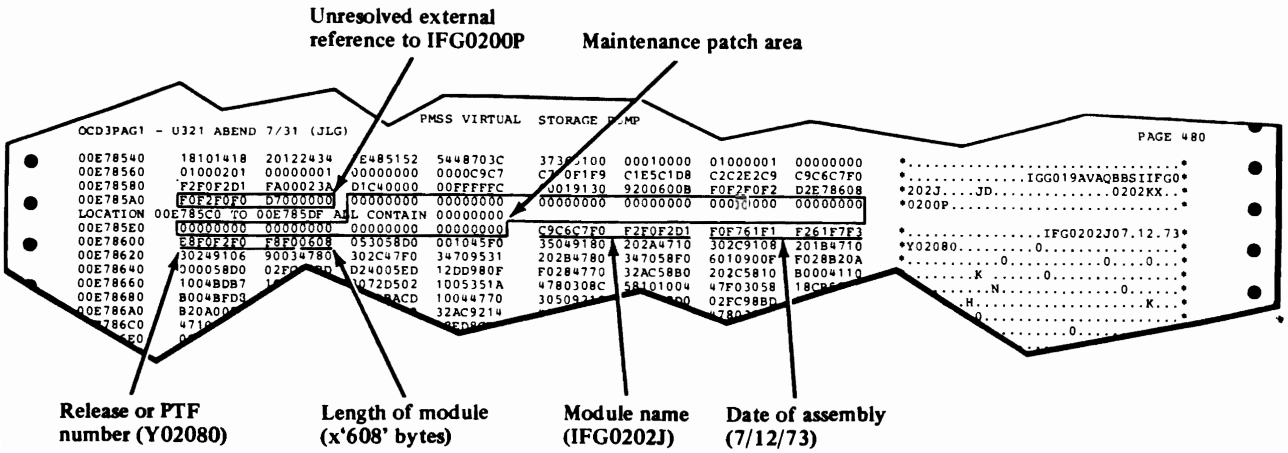
Internal Code	ABEND Code	Return Code	Reason
9	RTIOT		Address of TIOT DD entry
10	RUCB		Address of UCB
11	RDEB		Address of DEB
	RB		Work register and parameter register
12	RC		Work register and parameter register
13	RD		Work register and parameter register
14	RET		Return register when a module branches to IFG019RA to issue a WAIT macro, offset of a branch-table entry in the next module to receive control when a module is branching to IFG019RA to transfer control, or work register
15	RF		Work register

Using the Transfer Control Table

The transfer control table can be used to add a temporary change to an O/C/EOV module and to locate a given object module in a storage dump. An example of a transfer control table appears in the "Data Areas" section.

Locating an Object Module in a Storage Dump

Unlike other system control program components, the module name of O/C/EOV modules appears at the **end** of the module in the storage dump. The 8-byte module name follows the maintenance patch area and precedes the date of assembly, the release or PTF number, and length of the module. Figure 5 below shows the module name of module IFG0202J. The object code **precedes** this module name.


Unresolved external reference to IFG0200P
Maintenance patch area
Release or PTF number (Y02080)
Length of module (x'608' bytes)
Module name (IFG0202J)
Date of assembly (7/12/73)

```
OCD3PAG1 - U321 ABEND 7/31 (JLG)           PMSS VIRTUAL  STORAGE D MP                                                    PAGE 480

00E78540   18101418  20122434   E485152  5448703C   373 5100  00010000  01000001  00000000   *................................*
00E78560   01000201  00000001  00000000  0000C9C7   C7 0F1F9  C1E5C1D8  C2C2E2C9  C9C6C7F0   *...............IGG019AVAQBBSIIFG0*
00E78580   F2F0F2D1  FA00023A  D1C40000  00FFFFFC    0019130  9200600B  F0F2F0F2  D2E78608   *202J....JD.............0202KX..*
00E785A0   F0F2F0F0  D7000000  00000000  00000000   00000000  00000000  00000000  00000000   *0200P..........................*
LOCATION 00E785C0 TO 00E785DF ALL CONTAIN 00000000
00E785E0   00000000  00000000  00000000  00000000   C9C6C7F0  F2F0F2D1  F0F761F1  F261F7F3   *................IFG0202J07.12.73*
00E78600   E8F0F2F0  F8F00608  053058D0  001045F0   35049180  202A4710  302C9108  201B4710   *Y02080........0..................*
00E78620   30249106  90034780  302C47F0  34709531   202B4780  347058F0  6010900F  F028B20A   *..........0............0....0...*
00E78640   000058D0  02F        D24005ED  12DD980F   F0284770  32AC58B0  202C5810  B0004110           K      0            *
00E78660   1004BDB7  1           072D502  1005351A   4780308C  58101004  47F03058  18C               N              0      *
00E78680   B004BFD3                ACD  10044770   30509                  D0  02FC98BD                H                    K...*
 0E786A0   B20A00                      32AC9214                            4780                        0                     *
   786C0   4710                                                                                              0
     6E0   0
```

Figure 5. Storage Dump Showing the Transfer Control Table for Module IFG0202J

Adding a Temporary Change in the Maintenance Patch Area of an O/C/EOV Module

A maintenance patch area is included in each O/C/EOV object module as the result of the expansion of the XCTLTABL macro. The maintenance patch area is 5% of the size of the object module and is located at the end of each module, immediately preceding the 8-byte module name. You can use the AMASPZAP utility to add a modification to an object module:

1. Locate the transfer control table in the source code.
2. Locate the maintenance patch area in the transfer control table. The beginning of the patch area is the the third lable from the end of the transfer control table (and the end of the module).
3. Using the displacement (location-counter value) of the maintenance patch area, verify that the patch area is all zeros, and apply the modification to the object module.

Notes:

- Remember to perform a cold start of the system after the alteration. A warm start will not reformat the PLPA to include your modification.
- Unresolved V-type address constants (VCONs), which are 4 bytes of zeros, sometimes immediately precede the maintenance patch area. Be careful not to modify the VCONs when adding a patch.

Using Error Records for Debugging

The O/C/EOV problem determination and system recovery routines record work areas and control blocks when abnormal termination conditions are encountered during O/C/EOV processing. The sections that follow describe the kinds of information recorded and how to use it. How to invoke the optional work area tracing and module-name listing routines is also described.

ABEND Error Recording Initiated by O/C/EOV

Both the O/C/EOV problem determination and recovery routines trace control blocks and work areas when abnormal termination conditions are encountered in O/C/EOV processing.

The problem determination routines are invoked only when the error is associated with known, describable errors. These errors, called *determinate errors*, are identified with a system completion code and a return code in register 15. The problem determination routines (described in Diagram 6.1 and Chart 5.1) are not invoked when an indeterminate error (for example, a program check) is encountered.

The O/C/EOV system recovery routines are called by the system recovery/termination manager (R/TM) as a result of the ESTAE macro issued during Open, Close, or EOV initialization. These recovery routines are called on every ABEND condition, determinate and indeterminate.

The problem determination and recovery routines pass information to the Generalized Trace Facility (GTF). The records passed to GTF appear in the ABEND dump. The records will also be written to the GTF data set under the following conditions: (1) if GTF is active and operating in external mode while the job to be traced is running and (2) if the operator responds "TRACE=USR" when the GTF message "SPECIFY TRACE OPTIONS" appears at the operator's console. GTF makes the records available to the ABDUMP routine and writes them to the GTF data set specified by the user. The user must define the data set that is to be used for the GTF output. For information on how to use GTF, see *OS/VS2 System Programming Library: Service Aids.*

The results of the GTF trace that is put into the GTF data set can be examined by executing the AMDPRDMP service aid utility. You should specify the following parameters on the EDIT statement of AMDPRDMP.

EDIT DDNAME=*name of GTF data set*,USR=DMA1

DMA1 is the name GTF uses to identify data put in the data set by O/C/EOV routines. The job control language used with the EDIT statement is described in *OS/VS2 System Programming Library: Service Aids*.

ABEND Error Recording by Problem Determination Routines

When a determinate error arises during O/C/EOV processing or during Open or Close access method executor processing, the problem determination routines gather data related to the abnormal termination condition and related information, and then issue the GTRACE macro to pass the information to GTF. After issuing the GTRACE macro, the O/C/EOV routines issue the ABEND macro instruction to pass control to the R/TM. The information passed to GTF will also appear in the ABEND dump. It will also be written to the GTF data set. An example of GTF output appears in Figure 6.

This GTF output resulted from a 213 system code with a return code of X'04', meaning that the format-1 DSCB for the data set to be opened could not be found.

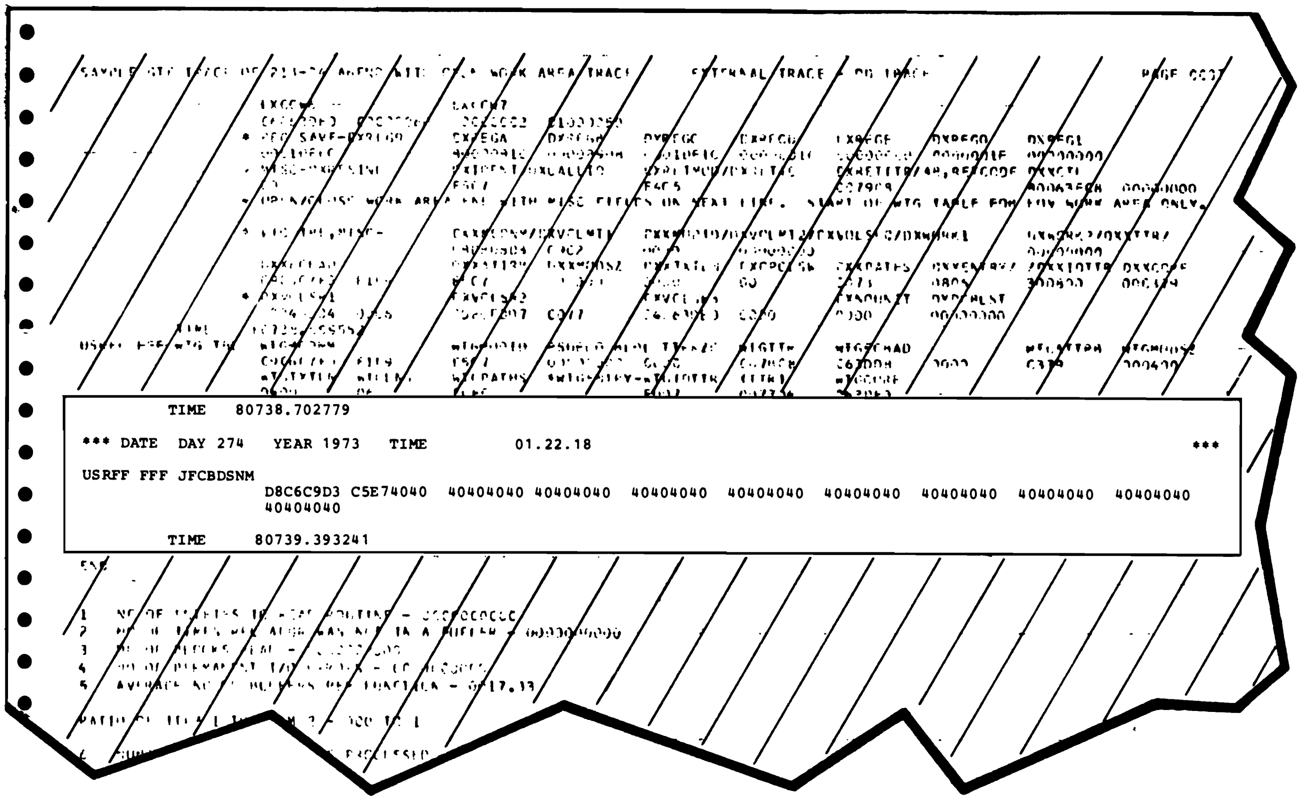
The problem determination routines write the data set name from the JFCB into the GTF data set.

ABEND Error Recording from the O/C/EOV System Recovery Routines

The O/C/EOV system recovery environment is established via the ESTAE macro, issued when storage was obtained during initialization of the O/C/EOV function. The O/C/EOV system recovery routines receive control from R/TM when any ABEND condition (determinate or indeterminate) is encountered during O/C/EOV processing.

When given control, the O/C/EOV recovery routines free storage, dequeue resources, and force data sets to closed status. Before doing any of these things, however, the O/C/EOV system recovery routines attempt to preserve a picture of the O/C/EOV chained work areas as they exist at time of entry. This is done by building a recovery routine control block save area (RRCBSA). Copies of the O/C/EOV chained work areas are saved in the RRCBSA.

Like the problem determination routines, the O/C/EOV system recovery routines issue the GTRACE macro instruction in order to record the contents of the RRCBSA built on entry. The RRCBSA segments are made available to the ABDUMP routine and written to the GTF data set by GTF.



```
          TIME    80738.702779

*** DATE   DAY 274    YEAR 1973    TIME          01.22.18                                                                  ***

USRFF FFF JFCBDSNM
                    D8C6C9D3 C5E74040  40404040 40404040  40404040  40404040  40404040  40404040  40404040  40404040
                    40404040

          TIME     80739.393241
```

Figure 6. Output from the GTF Data Set Resulting from Problem Determination ABEND Error Recording Routine (213 - 04 ABEND)

When O/C/EOV recovery processing has been completed, the RRCBSA is used to refresh the O/C/EOV chained work areas, which may have been modified during recovery processing. The RRCBSA is then freed. Note that an RRCBSA will not appear in a dump of the original error, but it will appear in a dump of an error that occurs during recovery processing.

The contents of the RRCBSA are defined in Figure 7. An example of a dump of this save area is shown in Figure 8.

The recovery routine gained control in this example when a 213 - 04 error condition was encountered (no format-1 DSCB could be found for the data set being opened).

The RRCBSA is made up of segments. Each segment begins with an ID of CBSA, which appears in the EBCDIC column of the dump. The first CBSA segment obtained is different from subsequent segments; in addition to a prefix and a section containing a copy of a control block or work area, the first segment contains a header. The header identifies the RRCBSA, provides the job and step name of the step that encountered the 213 ABEND condition and the date and time the error occurred. In Figure 8, the bottom-most segment, which contains the header, is the first segment. This segment is chained to the segments obtained subsequently using the prefix. In addition to the header, the first segment contains a copy of the initial work area obtained to open the DCB. The ID of this work area is O/C, which appears as "O.C" in the EBCDIC column. This initial work area contains a copy of the base prefix, an extended prefix, and a WTG table. Addresses in these copies reflect original storage locations. The contents of these locations

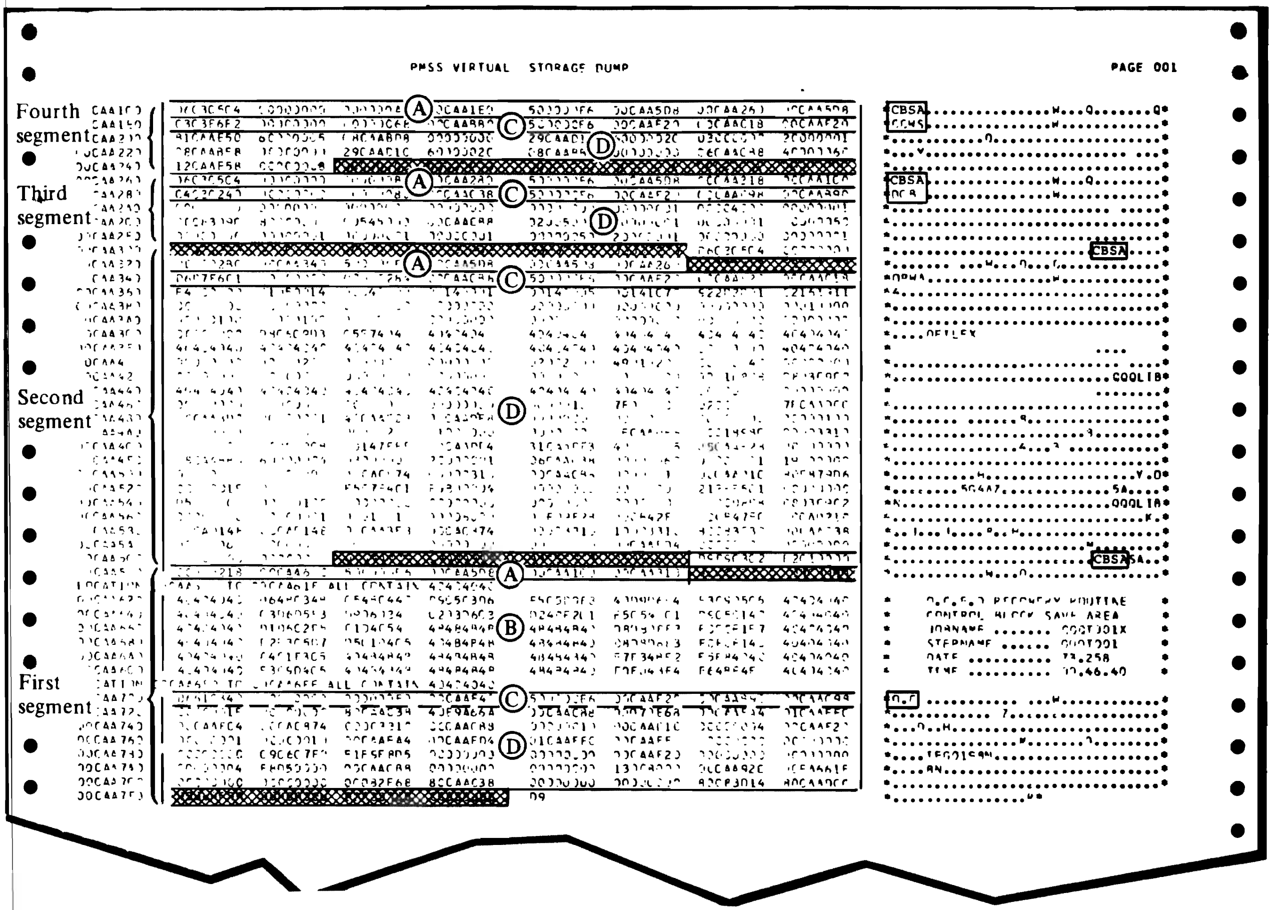
Prefix	0(0) Work area or control block ID	4(4) 4 bytes of zeros
	8(8) Work area length in bytes	12(C) ↑header or a copy of a work area
	16(10) Storage protection key and subpool	20(14) ↑initial (base) prefix
	24(18) ↑previous prefix	28(1C) ↑next prefix
Header	* * * O.C.E.D RECOVERY ROUTINE * * CONTROL BLOCK SAVE AREA * * JOBNAME .... JJJJJJJJ * * STEPNAME ...... SSSSSSSS * * DATE .... YY.DDD * * TIME .... HH.MM.SS * * *	
Copy of work area or control block saved by recovery routines		

Figure 7. Contents of Recovery Routine Control Block Save Area (RRCBSA)

may have been changed by recovery or termination processing. The address in these copies can be used, however, to find information in the copies.

The second segment of the example contains a copy of the main work area associated with the DCB. Its ID is "OPWA." The third segment contains a copy of the DCB (ID is "DCB"), and the fourth segment is an area that contains a channel program (ID is "CCWS"). Each of these segments is chained from the first segment using the prefixes. The prefix of each segment begins with the ID ("CBSA").

Each of the work areas obtained by the O/C/EOV routines has been assigned a 4-byte ID to make the work areas easy to identify in storage dumps. These IDS appear in the first word of every prefix of every work area



Ⓐ Prefixes to the RRCBSA segments.

Ⓑ RRCBSA header.

Ⓒ Prefix included in the copy of the work area or control block.

Ⓓ Copy of a control block or work area obtained by the main-line Open routines. Each area is identified by an ID in the first four bytes of its prefix (see C).

▨ Storage locations used to align IDs in the EBCDIC column.

Figure 8. Storage Dump Containing the RRCBSA Passed by the O/C/EOV Recovery Routines

gotten by O/C/EOV routines and saved in the RRCBSA. Each of the work areas and control blocks is obtained with a prefix. The contents of the prefix are described in Figure 7. The IDs and the contents of the associated work areas follow:

Work Area ID	Contents
O/C	The initial virtual storage obtained for an Open or Close request. This area contains, in addition to the base prefix and an extended prefix, a WTG table, a copy of the user parameter list, an audit trail, and a service routine (IFG019RA) work area.
OPWA	A main work area associated with one DCB or ACB being opened.
CLWA	A main work area associated with one DCB or ACB being closed.
TCWA	A main work area associated with one DCB being temporarily closed.

Work Area ID	**Contents**
EOV	A prefix, extended prefix, and a main work area associated with the DCB or ACB that requires EOV processing.
ULWA	User labels are read into and written from this area.
JFWA	When a JFCB extension is read in by an O/C/EOV routine, it is read into this area.
F1	When more than one format-1 DSCB must be read in, the second and subsequent format-1 DSCBs are read into a work area with this ID.
F2F3	When ISAM data sets are being processed, the format-2 DSCBs are read into an area with this ID. When data sets with more than three extents are being processed, the format-3 DSCBs are read into this area.
SFWA	Type-14 and -15 SMF records are built in this work work area.
SEWA	The Open/EOV password protection routines read the DSCB for the
PASSWORD	data set and password records into this area.
PDWA	When the problem determination routines are invoked, this work area is used to gather diagnostic information.
RJWA	ID of work area used by the RDJFCB routine.
CCWS	ID for a work area obtained to execute an extended channel program.
MSGS	Work area for building messages.
RR0A	ID for a work area obtained by the recovery routines.
TC0A	ID for a work area obtained by Task Close (data management resource manager).
QMGR	ID for a work area for obtaining and updating JFCBs using the QMNGRIO interface.
CBSA	ID for a control block save area (RRCBSA). One of these save areas is obtained by the recovery routine for each control block and save area to be saved for error recording during recovery processing.
SSOB	SSOB parameter list, SSOB parameter list pointer, and register save area for JES IEFSSREQ macro interface.

Certain control blocks are also saved in the RRCBSA. Their IDs are:

Control **Block ID**	**Contents**
DCB	Copy of user's DCB in protected storage.
DEB	Data extent block associated with the data set being processed.
DEBX	DEB extension
DEBT	DEB table associated with the job step.
UCB	Unit control block representing the unit on which data set being processed is mounted.

ABEND Error Recording in SYS1.LOGREC

The system diagnostic work area (SDWA) is passed among the various levels of system and user recovery (STAE, ESTAE, and FRR) routines. It is used to pass information among the recovery routines and to collect and record information about the environment in which the error occurred. The O/C/EOV recovery routines put the following information in the SDWA:

1. The CSECT name of the O/C/EOV routine that encountered the error will be moved into the SDWAMODN field.

2. The CSECT name of the O/C/EOV routine that encountered the error will also be moved into the SDWACSCT field.

3. The CSECT name of the O/C/EOV recovery routine handling the error will be moved into the SDWAREXN field.

4. Into the variable area of the SDWA, the O/C/EOV recovery routine moves the O/C/EOV recovery routine parameter list (RRPLIST). The contents of the list follow:

Offset	Length	Name	Description
0 (0)	1	RRFUNCTN	Identifies which O/C/EOV or DADSM function was in control at the time the error occurred
	...1 ..11	RRFOPEN	Open
	...1 .11.	RRFOPENJ	Open, Type=J
	...1 .1..	RRFCLOS	Close
	...1 .111	RRFTCLOS	Close, Type=T
	..11 .111	RRFEOV	End-of-Volume
	...1 1111	RRFFEOV	FEOV
	.1..	RRFRJFCB	Read JFCB
	1...	RRFDADSM	Indicates that one of the following DADSM functions was in control at the time the error occurred
	1.1.	RRFALLOC	DA space allocation
	1.11.111	RRFEXTND	Extend
	1..1 .1..	RRFPRLSE	Partial release
	11.. 111.	RRFLSPAC	LSPACE
	1..1 1.11	RRFOBTN	Obtain
	1..1 111.	RRFRENAM	Rename
	1..1 11.1	RRFSCRTH	Scratch
1 (1)	1	RRFLAGS1	Identifies the functions that have received control from the O/C/EOV and DADSM modules
	.1..	RRFUSREX	User exit routine in control
	..1.	RRFAMEXG	Access method executor in control
	...1	RRFAMEXR	Access method executor returned control
	.... 1...	RRFAMSTR	Access method executor string (e.g., VTAM or VSAM) in control
	.... .1..	RRFNSL	User's NSL routine given control
	.... ..1.	RRFSMF	SMF in control
	1	RRFTRACE	Trace routine in control
2 (2)	1	RRFLAGS2	Indicates key functions completed
	1...	RRFFIN1	First object module of the function has completed
	..1.	RRFENQPW	Enqueue on the PASSWORD data set ,
	...1	RRFQMNGR	QMNGR component given control
	.... ..1.	RRFPRDET	The O/C/EOV problem determination routines have been given control
	1	RRFENQUL	Enqueue on a user label track

ABEND Error Recording in SYS1.LOGREC

Offset	Length	Name	Description
3 (3)	1	RRFLAGS3	Indicates the type of termination that is in progress
	.1..	RRFNRTCA	No RTCA (same as the STAE diagnostic work area, SDWA) was passed
	..1.	RRFINFC	Final pass made by the Force Close function
	...1	RRFFCEXC	Force Close executor in control
	.... 1...	RRFTERM	Special termination entry
4 (4)	4	RR1WRKAR	Address of the recovery routine work area
8 (8)	4	RR1RETRY	First-level reccovery routine bypass address
12 (C)	4	RRXRETRY	For O/C/EOV, this is the recovery routine executor bypass address
		RRUCBPTR	For DADSM, this is the UCB address.
16 (10)	4	RRMLRTRY	O/C/EOV retry pointer
20 (14)	4	RRFWORK	Internal communications word

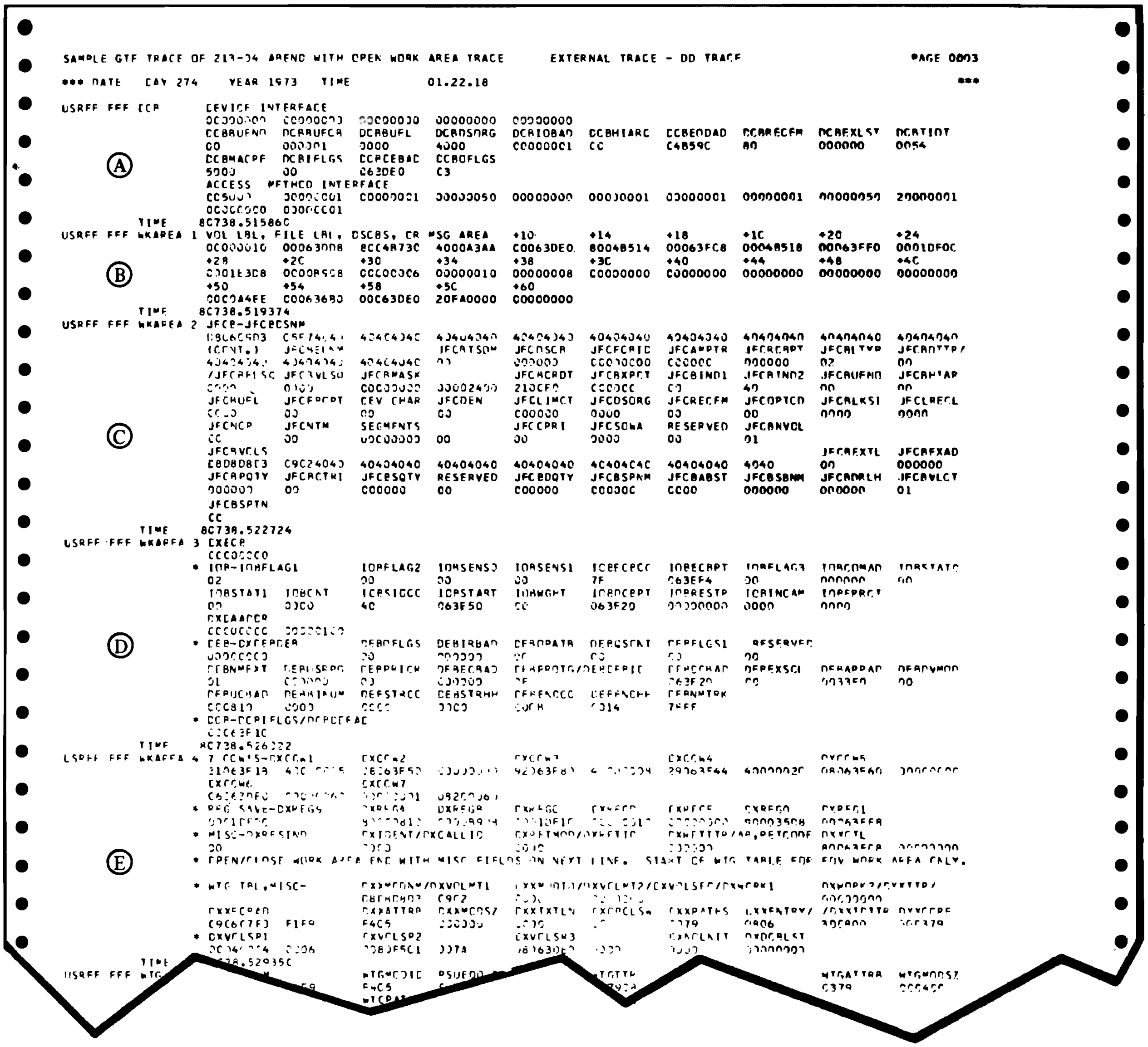
Problem Determination Optional Work Area Trace

The optional work area trace, normally used when a task fails repeatedly with evidence of an O/C/EOV problem, traces the ACB or DCB and the O/C/EOV work area associated with the ACB or DCB. This trace also includes a record of the address of the WTG table and the address of the current entry in the WTG table after each O/C/EOV module executes.

The user requests the optional work area trace by specifying DCB=DIAGNS=TRACE (AMP='TRACE' for VSAM) on the DD statement of a data set for which the trace is desired. The work area and control blocks are written to the GTF data set every time IFG019RA is entered. No tracing is done when control is passed to the O/C/EOV problem determination and recovery routines, the initial and final modules of each function of O/C/EOV. Tracing occurs only during access method executor processing for the SAM, BDAM create, and BPAM executors.

Figure 9 shows the GTF output resulting from activating the optional work area trace. In this example, the Open routines abnormally terminated the task because the format-1 DSCB for the data set could not be found. The associated system and return codes are 213 - 04. The traced record in Figure 9 includes a copy of the DCB, a copy of the O/C/EOV main work area associated with the DCB, various fields from the main work area, and the WTG table. This record will appear several times in the GTF output, once each time an Open object module is given control.

Because the optional work area trace increases execution time considerably and produces copious output, it should be used only when other diagnostic techniques fail.



```
SAMPLE GTF TRACE OF 213-04 ABEND WITH OPEN WORK AREA TRACE       EXTERNAL TRACE - DD TRACE                              PAGE 0003
*** DATE    DAY 274     YEAR 1973    TIME           01.22.18                                                                    ***
USRFF FFF DCB       DEVICE INTERFACE
                    0C000000  C0000000  00000030  00000000  00000000
                    DCBBUFNO  DCBBUFCB  DCBBUFL   DCBDSORG  DCBIOBAD  DCBHIARC  DCBEODAD  DCBRECFM  DCBEXLST  DCBTIOT
                    00        000001    0000      4000      C0000001  CC        C4B59C    80        000000    0054
        (A)         DCBMACRF  DCBIFLGS  DCBDEBAD  DCBOFLGS
                    5000      00        063DE0    C3
                    ACCESS  METHOD INTERFACE
                    CC5000    0000C001  C0000001  00000050  00000000  00000001  00000001  00000001  00000050  20000001
                    00000000  0000CC01
          TIME     8C738.51586C
USRFF FFF WKAREA 1 VOL LBL, FILE LBL, DSCBS, CR MSG AREA    +10       +14       +18       +1C       +20       +24
                    0C000010  00063008  8CC4B73C  4000A3AA  C0063DE0  80048514  00063FC8  00048518  00063FF0  0001DF0C
                    +28       +2C       +30       +34       +38       +3C       +40       +44       +48       +4C
        (B)         0001E3D8  0C00B5C8  00C00006  00000010  00000008  C0000000  C0000000  00000000  00000000  00000000
                    +50       +54       +58       +5C       +60
                    00C0A4EE  C00636B0  00C63DE0  20FA0000  C0000000
          TIME     8C738.519374
USRFF FFF WKAREA 2 JFCB-JFCBDSNM
                    D8D6C9D3  C5F74C4C  404C404C  40404040  40404040  40404040  40404040  40404040  40404040  40404040
                    (CONT.)   JFCBELNM            JFCBTSDM  JFCDSCB   JFCFCBID  JFCAMPTR  JFCBCBPT  JFCBLTYP  JFCBOTTR/
                    40404040  40404040  404C404C  00        000000    CC000C00  C0000C    000000    02        00
                    /JFCBFLSC JFCBVLSQ  JFCBMASK            JFCBCRDT  JFCBXPDT  JFCBIND1  JFCBIND2  JFCBUFNO  JFCBHIAR
                    C000      0100      C0C00000  00002400  210CF0    C000CC    C0        40        00        00
                    JFCBUFL   JFCERPPT  DEV CHAR  JFCDEN    JFCLIMCT  JFCDSORG  JFCRECFM  JFCOPTCD  JFCBLKSI  JFCLRECL
                    CC00      00        00        C0        C00000    0000      00        00        0000      0000
                    JFCNCP    JFCNTM    SEGMENTS            JFCCPRI   JFCSOWA   RESERVED  JFCBNVOL
        (C)         CC        00        00C00000  00        00        0000      00        01
                    JFCBVOLS                                                                      JFCBEXTL  JFCBEXAD
                    E8D8D8E3  C9C24040  40404040  40404040  40404040  4C404C4C  40404040  4040      00        000000
                    JFCBPQTY  JFCBCTRI  JFCBSQTY  RESERVED  JFCBDQTY  JFCBSPNM  JFCBABST  JFCBSBNM  JFCBDRLH  JFCBVLCT
                    000000    00        C00000    00        C00000    C0000C    C000      000000    000000    01
                    JFCBSPTN
                    CC
          TIME     8C738.522724
USRFF FFF WKAREA 3 DXECB
                    CCC0C0C0
                  * IOB-IOBFLAG1            IOBFLAG2  IOBSENS0  IOBSENS1  IOBECBCC  IOBECBPT  IOBFLAG3  IOBCOMAD  IOBSTATC
                    02                      00        00        00        7F        063EF4    00        000000    00
                    IOBSTAT1  IOBCNT        IOBSIOCC  IOBSTART  IOBWGHT   IOBDCBPT  IOBRESTR  IOBINCAM  IOBERRCT
                    00        0000          40        063F50    00        063F20    00000000  0000      0000
                    DXCAADCB
                    CCC0C0CC  00000100
        (D)       * DEB-DXDEBDEB            DEBDFLGS  DEBIRBAD  DEBOPATB  DEBQSCNT  DEBFLGS1   RESERVED
                    00000000                00        000000    0C        C0        C0        00
                    DEBNMEXT  DEBUSRPG  DEBPRIOR  DEBECBAD  DEBPROTG/DEBDEBID       DEBDCBAD  DEBEXSCL  DEBAPPAD  DEBDVMOD
                    01        C00000    00        000000    0F                      063F20    00        0033F0    00
                    DEBUCBAD  DEBBINUM  DEBSTRCC  DEBSTRHH  DEBENDCC  DEBENDHH  DEBNMTRK
                    00C810    0000      0000      0000      00C8      0014      7FFF
                  * DCB-DCBIFLGS/DCBDEBAD
                    00063F1C
          TIME     8C738.526222
USRFF FFF WKAREA 4 7 CCW'S-DXCCW1           DXCCW2              DXCCW3              DXCCW4              DXCCW5
                    31063F18  40C00005  08063F50  00000000  92063F80  40000008  29063F44  4000002C  08063F60  00000000
                    DXCCW6                  DXCCW7
                    C6063DF0  000000000 00000001  08200067
                  * REG SAVE-DXREGS         DXREGA    DXREGB    DXREGC    DXREGD    DXREGE    DXREGO    DXREG1
                    0001DF0C                80000812  00008918  0001DF1C  00000010  00000000  00003508  00063FF8
                  * MISC-DXRESIND           DXIDENT/DXCALLID    DXRETMOD/DXRETID    DXRETTTR/AP,RETCODE DXXCTL
                    00                      0000                0000                000000              80063FC8  00000000
        (E)       * OPEN/CLOSE WORK AREA END WITH MISC FIELDS ON NEXT LINE.  START OF WTG TABLE FOR EOV WORK AREA ONLY.

                  * WTG TBL,MISC-           DXXMODNM/DXVOLMT1   DXXMODID/DXVOLMT2/DXVOLSEQ/DXWORK1      DXWORK2/DXXTTR/
                                            D8C8D8D3  C9C2      0000      00000000                      00000000
                    DXXECBAD                DXXATTRB  DXXMODSZ  DXXTXTLN  DXOPCLSW  DXXPATHS  DXXENTRY/ /DXXIDTTR DXXCCPE
                    C9C6C7F0  F1F9          F4C5      000000    0000      00        0379      0806      300000    000379
                  * DXVOLSP1                DXVOLSP2            DXVOLSP3            DXNOLNIT  DXDCBLST
                    0C040014  0006          0080F5C1  0074      /8063000            0000      00000000
          TIME     8C738.529350
USRFF FFF WTG                               WTGMODID  PSUEDO              WTGTTR                          WTGATTRB  WTGMODSZ
                                    F9      F4C5                          0790A                           0379      000400
                                            WTCPA
```

(A) A formatted copy of the DCB that was being opened when the error condition was encountered.

(B) The first 100 bytes of the O/C/EOV main work area associated with the DCB being opened. Since no labels or DSCBs had been read in when the error occurred, this information is not significant.

(C) The JFCB being used by the Open routine.

(D) The internal I/O control blocks used by the Open routine.

(E) Various fields from the main work area followed by portions of the WTG table associated with the OPEN request.

Figure 9. Output from GTF data set resulting from the optional work area trace

The Optional Module Listing Routine

IFG019RA contains a module trace routine that, when active, lists the CSECT name of every O/C/EOV routine given control during O/C/EOV processing.

This tracing routine is located in the maintenance space of IFG019RA. Unless you modify the code, the routine is not entered. To activate the trace routine, change the NOP instruction at location RESTRACE to an unconditional branch (X'47F0') to location RESNOP. To make this change, you will need to locate IFG019RA in the PLPA. The address of IFG019RA is at offset X'110' in the CVT; that location is labeled CVTDMSVR.

Each time IFG019RA transfers control, the activated trace routine issues a WTO macro instruction with an 8-byte message text of the CSECT name of next CSECT to receive control. The message is assigned routing codes 2 and 11, so that it is sent to the console and is written to the SYSOUT data set associated with the job. You can route the message to other devices by changing the routing code, as described in *OS/VS2 Supervisor Services and Macro Instructions*.

The tracing routine is not reentrant; if more than one task tries to use it at the same time, it may be impossible to tell which messages result from which task; second, messages resulting from the two tasks may overlay each other in the O/C/EOV work area message buffer, so that some of the messages (CSECT names) are lost altogether.

If SMF is active, EOV may transfer control to Close routines to process SMF records. The result is that Close CSECT names appear between EOV CSECT names in the list.

When the trace routine gets control from the O/C/EOV modules, registers 2 through 8 are used as follows:

- Register 2 contains the address of the first DCB or ACB to be processed after the transfer of control.
- Register 3 contains the entry point address of the module currently in control.
- Register 4 contains the address of the O/C/EOV work area associated with the DCB or ACB whose address is in register 2.
- Register 5 is the base register.
- Register 6 contains the address of the WTG table for the Open or Close operation being performed.
- Register 7 contains the address of the parameter list entry for the DCB or ACB whose address is in register 2.
- Register 8 contains the address of the WTG table entry for the DCB or ACB whose address is in register 2.

If this routine is modified, registers 0, 1, 3, and 9 through 15 may be used without their contents being saved or restored.

Module Listing Routine Examples

Various combinations of O/C/EOV modules are executed based on the operation requested, the processing mode (input, output, or update), and the kind of device on which the data set resides. The module lists that follow have been prepared to indicate the flow of control during successful execution under several typical conditions. The conditions treated and their associated example numbers are:

OPEN Example	**Number**
SYSIN data set	1
SYSOUT data set	2
DA input data set	3
DA output data set	4
Two-volume ISAM input data set	5
SL tape input data set	6
SL tape output data set	7
NSL tape input data set	8
NSL tape output data set	9

EOV Example	**Number**
EOD, SYSIN data set	1
EOD, DA input data set	2
EOV, two-volume DA input data set	3
EOD concatenation, like attributes, DA input data sets	4
EOV, DA output data set, additional space obtained on same volume	5
EOV, DA output data set, additional space obtained on a new volume	6
EOD, SL input tape data set	7
EOV, SL tape input data set	8
EOV, SL tape output data set	9

CLOSE Example	**Number**
SYSIN data set	1
DA input data set	2
SL tape input data set	3

OPEN Trace Examples

OPEN Example 1—SYSIN

Open a SYSIN data set resulting from a //DD * in the jobstream. The cards will have been spooled to a direct-access device by the job entry subsystem before the problem program is given control. The problem program contains the following DD statement and DCB associated with the SYSIN file.

```
//DD1      DD       *

DCB1       DCB      DDNAME=DD1,BLKSIZE=80,LRECL=80,                    X
                    DSORG=PS,EROPT=ABE,MACRF=GM,                       X
                    EODAD=EOF1
```

The Open modules executed when the `OPEN DCB1` macro is issued:

IGC0001I	Open initial
IFG0193A	Read JFCB
IFG0196J	Merge JFCB to DCB
IFG0196L	Call user's DCB exit routine
IFG0196M	Merge DCB to JFCB
IFG0196V	Go to an Open access method executor

Open access method executor IGG0199F is given control. This module constructs an ACB with the attributes of DCB1 and the JFCB and issues an `OPEN ,,,TYPE=J` macro for the ACB, which causes the Open routines to be reentered:

IGC0002B	OPEN, TYPE=J
IFG0193A	Determine data set organization
IFG0196V	Go to an Open access method executor

Open executor IGG0193K is given control to build a DEB, etc. IGG0193K returns to:

IFG0196W	Access method executor return (ACB)
IFG0196X	Moves JFCB to SWA (ACB)
IFG0198N	Frees O/C/EOV work area (ACB)

Control is returned to the issuer of the OPEN for the ACB, IFG0199F, which returns control to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA (DCB)
IFG0198N	Frees O/C/EOV work area (DCB)

Return to the issuer of the OPEN DCB1 macro.

OPEN Example 2—SYSOUT

Open a SYSOUT data set to be written to a device with an output class of A. This data set will be spooled to an intermediate JES data set.

```
//DD2      DD       SYSOUT=A

DCB2       DCB      DDNAME=DD2,BLKSIZE=80,LRECL=80,                    X
                    DSORG=PS,EROPT=ABE,MACRF=PM,RECFM=F
```

The Open modules executed when the `OPEN DCB2` macro is issued:

IGC0001I	Open initial
IFG0193A	Determine data set organization
IFG0196J	Merge JFCB to DCB
IFG0196L	Call user's DCB exit routine
IFG0196M	Merge DCB to JFCB
IFG0196V	Go to an Open access method executor

Open access method executor IGG0199F is given control. This module constructs an ACB with the attributes of DCB2 and the JFCB and issues an OPEN , , ,TYPE=J macro for the ACB, which causes the Open routines to be reentered:

IGC0002B	OPEN, TYPE=J
IFG0193A	Determine data set organization
IFG0196V	Go to an Open access method executor

Open executor IGG0193K is given control to build a DEB, etc. IGG0193K returns to:

IFG0196W	Access method executor return (ACB)
IFG0196X	Moves JFCB to SWA (ACB)
IFG0198N	Frees O/C/EOV work area (ACB)

Control is returned to the issuer of the OPEN for the ACB, IFG0199F, which returns control to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA (DCB)
IFG0198N	Frees O/C/EOV work area (DCB)

Return to the issuer of the OPEN DCB2 macro.

OPEN Example 3—DA Input

Open a data set on a direct-access device for input.

```
//DD1     DD       UNIT=2314,DISP=OLD,VOL=SER=QQLIB,            X
//                 DSN=QFILE1

DCB1      DCB      DDNAME=DD1,BLKSIZE=80,LRECL=80,              X
                   DSORG=PS,EROPT=ABE,MACRF=GM,                 X
                   RECFM=F,EODAD=EOF1
```

The following O/C/EOV modules are executed when the problem program issues the OPEN DCB1 macro:

IGC0001I	Open initial
IFG0193A	Read JFCB
IFG0194A	Select a unit
IFG0194C	Verify a volume
IFG0194A	Volume verified
IFG0195A	DSCB-to-JFCB merge
IFG0195J	User label determination
IFG0196J	JFCB-to-DCB merge
IFG0196L	User DCB exit
IFG0196M	DCB-to-JFCB merge
IFG0196V	Go to an Open access method executor

Open access method executor IGG0191A is given control and returns control to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Frees O/C/EOV work area

Control is return to the issuer of the OPEN macro.

OPEN Example 4—DA Output

Open a data set on a direct-access device for output.

```
//DD2    DD     UNIT=SYSDA,DISP=(NEW,PASS),                    X
//              VOL=SER=SCRTCH,DSNAME=OUTFILE1,                X
//              SPACE=(TRK,(40,10)0,LABEL=RETPD=0

DCB2     DCB    DDNAME=DD2,BLKSIZE=80,LRECL=80,                X
                DSORG=PS,EROPT=ABE,                            X
                MACRF=PM,RECFM=F

         OPEN   (DCB2,(OUTPUT))
```

IGC0001I	Open initial
IFG0193A	Read JFCB
IFG0194A	Select a unit
IFG0194C	Verify a volume
IFG0194A	Volume verified
IFG0195A	DSCB-to-JFCB merge
IFG0195J	User label determination
IFG0196J	JFCB-to-DCB merge
IFG0196L	User DCB exit
IFG0196M	DCB-to-JFCB merge
IFG0196V	Go to an Open access method executor

Open access method executor IGG0191A is given control and returns control to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Frees O/C/EOV work area

Control is return to the issuer of the OPEN macro.

OPEN Example 5—Multivolume ISAM Input

Open a two-volume ISAM data set for input. Note that the list of modules executed would be the same for a BDAM data set opened for input.

```
//DD1         DD     DSNAME=PART86,UNIT=2314,                  X
//                   DISP=OLD,VOL=SER=538765,538766

DCB1          DCB    DSNAME=DD1,DSORG=IS,MACRF=GM

              OPEN   DCB1
```

IGC0001I	Open initial
IFG0193A	Read JFCB
IFG0194A	Select a unit
IFG0194C	Verify volume

If neither volume was mounted at the time the OPEN macro was issued, the following volume mount and verification modules would be executed to get both volumes mounted:

IFG0194J	Ask the operator to mount the first volume
IFG0194C	Verify the first volume
IFG0194J	Ask the operator to mount the second volume

IFG0194C	Verify the second volume
IFG0194A	Volume verified
IFG0195A	DSCB-to-JFCB merge
IFG0195J	User label determination
IFG0196J	JFCB-to-DCB merge
IFG0196L	User DCB exit
IFG0196M	DCB-to-JFCB merge
IFG0196V	Go to an Open access method executor

Open access method executor IGG0191A is given control and returns control to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Frees O/C/EOV work area

Control is returned to the issuer of the OPEN macro.

OPEN Example 6—SL Tape Input

Open an input data set on a standard labeled magnetic tape.

```
//DD1     DD      UNIT=2400,DISP=OLD,                              X
//                VOL=SER=PS6342,DSNAME=PARTJURN,                  X
//                LABEL=SL

DCB1      DCB     DDNAME=DD1,BLKSIZE=80,LRECL=80,                  X
                  DSORG=PS,MACRF=GM,                               X
                  RECFM=F,EODAD=EOF1

          OPEN    DCB1
```

IGC0001I	Open initial
IFG0193A	Determine access method
IFG0194A	Select a unit
IFG0194F	Verify a volume

If the volume had not been mounted at the time the OPEN macro was issued, the following modules would have been executed.

IFG0194J	Ask operator to mount the volume
IFG0194F	Verify the volume
IFG0194A	Volume verified
IFG0195B	Position SL tape
IFG0195H	Verify HDR1 label
IFG0195K	Verify HDR2 label

If the volume were password protected, the following modules would be executed to verify the password:

IFG0195T	Password initialization
IFG0195V	Read DSCB for PASSWORD data set
READPSWD	Ask operator for password
IFG0195U	Update password record
IFG0195T	Password verified
IFG0196J	JFCB-to-DSCB merge
IFG0196L	Give control to user's DCB exit routine

IFG0196M	DCB-to-JFCB merge
IFG0196V	Give control to access method Open executor

SAM Open access method executor, IGG0191A, is given control and when executors have completed processing, control is returned to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Open final

Returns control to issuer of OPEN macro.

OPEN Example 7—SL Tape Output

Open an output data set on a standard labeled magnetic tape.

```
//DD2        DD      UNIT=TAPE,DISP=(NEW,KEEP),               X
//                   VOL=SER=PS2093,LABEL=SL

DCB2         DCB     DDNAME=DD2,BLKSIZE=80,LRECL=80,          X
                     DSORG=PS,MACRF=PM

             OPEN    (DCB2,(OUTPUT))
```

IGC0001I	Open initial
IFG0193A	Determine data set organization
IFG0194A	Select a unit
IFG0194F	Verify volume
IFG0194A	Volume verified.
IFG0195B	Position tape
IFG0196J	JFCB-to-DCB merge
IFG0196L	Take user's DCB exit
IFG0196M	DCB-to-JFCB merge
IFG0196N	Read existing HDR1
IFG0196Q	Write volume label
IFG0196T	Write HDR1 and HDR2 labels
IFG0196V	Go to Open access method executor

SAM Open access method executor, IGG0191A, is given control and when the executors have completed processing, control is returned to:

IFG0196W	Open access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Open final

Returns control to issuer of the OPEN macro.

OPEN Example 8—NSL Tape Input

Open an input data set on an nonstandard-labeled magnetic tape.

```
//DD1    DD     UNIT=2400,DISP=OLD,VOL=SER=PS6342,            X
//              DSNAME=PARTJURN,LABEL=NSL

DCB1     DCB    DDNAME=DD1,BLKSIZE=80,LRECL=80,               X
                DSORG=PS,MACRF=GM,RECFM=F,                    X
                EODAD=EOF1

         OPEN   DCB1
```

IGC0001I	Open initial
IFG0193A	Determine access method
IFG0194A	Select a unit
IFG0194F	Verify a volume

If the volume had not been mounted at the time the OPEN macro was issued, the following modules would have been executed.

IFG0194J	Ask operator to mount the volume
IFG0194F	Verify the volume
IFG0194A	Volume verified
IFG0195D	Gives control to NSLOHDRI
IFG0196J	JFCB-to-DSCB merge
IFG0196L	Give control to user's DCB exit routine
IFG0196M	DCB-to-JFCB merge
IFG0196V	Give control to access method Open executor

SAM Open access method executor, IGG0191A, is given control and when executors have completed processing, control is returned to:

IFG0196W	Access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Open final

Returns control to issuer of OPEN macro.

OPEN Example 9—NSL Tape Output

Open an output data set on a nonstandard-labeled magnetic tape.

```
//DD2      DD    UNIT=TAPE,DISP=(NEW,KEEP),                  X
//               VOL=SER=PS2093,LABEL=NSL

DCB2       DCB   DDNAME=DD1,BLKSIZE=80,LRECL=80,             X
                 DSORG=PS,MACRF=PM

           OPEN  (DCB2,(OUTPUT))
```

IGC0001I	Open initial
IFG0193A	Determine data set organization
IFG0194A	Select a unit
IFG0194F	Verify volume
IFG0196O	Give control to NSLOHDRO
IFG0196V	Give control to an Open access method executor

SAM Open access method executor, IGG0191A, is given control and when the executors have completed processing, control is returned to:

IFG0196W	Open access method executor return
IFG0196X	Moves JFCB to SWA
IFG0198N	Open final

Returns control to issuer of the OPEN macro.

EOV Trace Examples

EOV Example 1—EOD, SYSIN

The end-of-data (EOD) condition arises for a SYSIN data set. When opened, the data set was identified by the following DD statement and DCB in the problem program:

```
//DD1     DD      *

DCB1      DCB     DDNAME=DD1,BLKSIZE=80,LRECL=80,                X
                  DSORG=PS,EROPT=ABE,MACRF=GM,EODAD=EOF1
```

The EOV modules executed when `EOV DCB1` is issued by the access method processing routine:

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0552X	Go to user's EODAD routine

Note: No EOV condition is raised for a SYSOUT data set.

EOV Example 2—EOD, DA Input

EOV macro issued for a data set on a direct-access device opened for input. When opened, the data set was identified by the following DD statement and DCB macro in the problem program.

```
//DD1     DD      UNIT=2314,DISP=OLD,VOL=SER=QQLIB,              X
//                DSN=QFILE1

DCB1      DCB     DDNAME=DD1,BLKSIZE=80,LRECL=80,                X
                  DSORG=PS,EROPT=ABE,MACRF=GM,RECFM=F,           X
                  EODAD=EOF1
```

The following EOV modules are executed when `EOV DCB1` is issued by an access method processing routine:

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0553P	Read F1 DSCB
IFG0552X	Go to user's EODAD routine

EOV Example 3—EOV, DA Input

EOV macro issued on the first volume of a two-volume direct-access data set opened for input.

At the time the data set was opened, it was identified with the following DD statement and DCB macro:

```
//DD1     DD      UNIT=(2314,2),DSNAME=QFILE1,DISP=OLD,          X
//                VOL=SER=(534632,534633)

DCB1      DCB     DDNAME=DD1,BLKSIZE=80,LRECL=80,                X
                  DSORG=PS,EROPT=ABE,MACRF=GM,RECFM=F,           X
                  EODAD=EOF1
```

When the EOVDCB1 macro was issued, the following EOV modules were executed:

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0553P	Read F1 DSCB
IFG0554A	Select unit for new volume
IFG0554C	Verify volume
IFG0554A	Volume verified
IFG0553X	Build new DEB
IFG0553Z	Free EOV work area
IFG0551L	SAM EOV input executor

Control is returned to the issuer of the GET/PUT or READ/WRITE macro.

EOV Example 4—EOD, DA Concatenation

An EOD condition is encountered for first of two data sets with like characteristics being concatenated for processing. The data sets were opened for input and reside on different volumes.

The DD statements and DCB macro associated with the data sets were:

```
//DD1     DD     DSNAME=PART84,UNIT=2314,VOL=SER=573426,      X
//               DISP=OLD

//        DD     DSNAME=PART85,UNIT=2314,DISP=OLD,            X
//               VOL=SER=573427

DCB1      DCB    DDNAME=DD1,BLKSIZE=80,LRECL=80,              X
                 DSORG=PS,EROPT=ABE,MACRF=GM,RECFM=F,         X
                 EODAD=EOF1
```

When the EOVDCB1 macro was issued, the following EOV modules were executed:

IGC0005E	EOV intitalization
IFG0551F	Determine DSORG, current data set
IFG0551H	Determine detive type, current data set
IFG0553P	Read F1 DSCB
IFG0552X	Update control blocks
IFG0551F	Determine DSORG, new data set
IFG0551H	Determine device type, new data set
IFG0554A	Select unit for new volume
IFG0554C	Verify volume
IFG0554A	Volume verified
IFG0553X	Build new DEB
IFG0553Z	Free EOV work area
IFG0551L	SAM EOV input executor

Control is returned to the issuer of the GET/PUT or READ/WRITE macro.

EOV Example 5—EOV, DA Output, Same Volume

The EOV condition arises while writing output to a physical sequential data set on a direct-access device. The DD statement indicates a secondary space allocation, the secondary quantity is available on the current volume, and it is allocated there.

The DD statement and DCB associated with the data set follows:

```
//DD2     DD      UNIT=SYSDA,DISP=(NEW,PASS),                    X
//                VOL=SER=SCRTCH,DSNAME=OUTFILE1,                X
//                SPACE=(TRK,(40,10)

DCB2      DCB     DDNAME=DD2,BLKSIZE=80,LRECL=80,                X
                  DSORG=PS,MACRF=PM,RECFM=F
```

When the EOV DCB2 macro was issued by the access method processing routine, the following EOV modules were executed:

IGC0005E	EOV initialization
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0554P	Call DADSM Extend routines to get more space on current volume
IGG0553A	Extend initialization
IGG0553G	Request conversion
IGG0553B	Find available space
IGG0553C	Update F1/F3 DSCBs
IGG0553D	Update F5 DSCBs
IGG0553E	Update F4 DSCB
IFG0554P	Space obtained on current volume
IFG0555H	Build new DEB
IFG0551L	SAM EOV executor

Return to issuer of GET/PUT or READ/WRITE macro.

EOV Example 6—EOV, DA Output, New Volume

EOV condition arises while writing output to a direct-access volume. The DD statement indicates secondary space, but none is available on the current volume, so the space must be acquired on a new volume.

```
//DD2     DD      UNIT=SYSDA,DISP=(NEW,PASS),                    X
//                VOL=SER=SCRTCH,DSNAME=OUTFILE1,                X
//                SPACE=(TRK,(40,10)

DCB2      DCB     DDNAME=DD2,BLKSIZE=80,                         X
                  LRECL=80,DSORG=PS,MACRF=PM,                    X
                  RECFM=F
```

When the EOV DCB2 macro was issued by the access method processing routine, the following EOV modules were executed:

IGC0005E	EOV initialization
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0554P	Call DADSM Extend routines to get more space on current volume

IGG0553A	Extend initialization
IGG0553G	Request conversion
IGG0553B	Looks for additional space but can't find it
IFG0553E	Returns to EOV indicating no space available on current volume
IFG0554P	Checks DADSM return code
IFG0554Z	Update F1 DSCB, current volume
IFG0554P	Call volume mount/verification to mount a new volume
IFG0554A	Select a unit
IFG0554C	Verify volume mounted
IFG0554J	Request that volume be mounted
IFG0554C	Verify volume mounted
IFG0554P	Call DADSM Extend to get space on the new volume
IGG0553A	Extend initialization
IGG0553G	Request conversion
IGG0553B	Find available space
IGG0553C	Update F1/F3 DSCBs
IGG0553D	Update F5 DSCBs
IGG0553E	Update F4 DSCB
IFG0554P	Space obtained on new volume
IFG0554C	Read F1 DSCB
IFG0554A	Volume verified
IFG0555H	Build a new DEB
IFG0551L	SAM EOV EXECUTOR

Returns to issuer of the EOV macro.

EOV Example 7—EOD, SUL Input Tape

End-of-file reached on an input tape data set with standard user labels (SUL).

```
//DD1     DD    UNIT=2400,DISP=OLD,VOL=SER=PS6342,          X
//              DSNAME=PARTJURN,                            X
//              LABEL=SUL

DCB1      DCB   DDNAME=DD1,BLKSIZE=80,LRECL=80,             X
                DSORG=PS,MACRF=GM,RECFM=F,                  X
                EODAD=EOF1
```

When the EOV DCB1 macro is issued, the following EOV modules are executed:

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0552R	Read first standard trailer label
IFG0552T	Read user labels
IFG0552V	Position the volume
IFG0552X	Gives control to the user's EODAD routine

EOV Example 8—EOV, SL Input Tape

EOV reached on the first volume of a two-volume input tape data set with standard labels. The second volume of the data set is already mounted.

```
//DD1       DD    UNIT=(2400,2),DISP=OLD,                      X
//          DSNAME=PARTJURN,VOL=SER=(PS6342,PS6343),
            LABEL=SL

DCB1        DCB   DDNAME=DD1,BLKSIZE=80,LRECL=80,              X
                  DSORG=PS,MACRF=GM,RECFM=F,                   X
                  EODAD=EOF1
```

When the EOV DCB1 macro is issued, the following EOV modules are executed:

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0552R	Read first standard trailer label
IFG0552V	Position current (first) volume
IFG0554A	Select unit for new (second) volume
IFG0554F	Verify new volume mounted
IFG0554A	New volume is mounted
IFG0553F	Verify HDR1 label
IFG0553H	Verify HDR2 label
IFG0551L	SAM EOV executor

Return to issuer of the EOV macro.

EOV Example 9—EOV, SL Output Tape

EOV condition results from filling the first volume of a two-volume output tape data set. Second volume is already mounted.

The data set was identified with the following DD statement and DCB in the problem program:

```
//DD2       DD    UNIT=TAPE,DISP=(NEW,KEEP),                   X
//                VOL=SER=SCRTCH,LABEL=SL

DCB2        DCB   DDNAME=DD2,BLKSIZE=80,LRECL=80,              X
                  DSORG=PS,MACRF=PM
```

The following EOV modules were executed when the EOV DCB2 macro was issued.

IGC0005E	EOV initial
IFG0551F	Determine DSORG
IFG0551H	Determine device type
IFG0551P	Write EOV1 label
IFG0551R	Write EOV2 label
IFG0551T	Position current volume
IFG0554A	Select unit for second volume
IFG0554F	Verify that volume is mounted
IFG0554A	Volume is mounted
IFG0552B	Rewrite the volume label on the new volume

IFG0552D	Write HDR1 label
IFG0552F	Write HDR2 label
IFG0551L	SAM EOV executor

Return to the issuer of the READ/WRITE or GET/PUT macro.

Close Trace Examples

CLOSE Example 1—SYSIN

Close a SYSIN data set.

The problem program contains the following DD statement and DCB associated with the data set:

```
//DD1     DD      *

DCB1      DCB     DDNAME=DD1,DSORG=PS,EROPT=ABE,                  X
                  MACRF=GM,EODAD=EOF1
```

The Close modules executed when the `CLOSE DCB1` macro is issued:

IGC00020	Close initial
IFG0200V	Determine DSORG
IFG0200W	Gives control to the access method executors

SAM Close executor IGG0201W is given control. This module issues a CLOSE macro against the ACB opened to process the data set (see Open Example 1), which causes a recursion through the Close routines:

IGC01020	Close initial
IFG0200V	Determine DSORG, read JFCB
IFG0200W	Gives control to access method executor

JES Close executor IGG0203K is given control to process the ACB; IGG0203K returns control to:

IFG0200Y	Access method executor return
IFG0202K	Delete DEB
IFG0202L	Free Close work area used for DCB
IFG0202L	Free Close work area used for ACB

Returns to the issuer of the CLOSE macro.

Note: The same modules are executed when a SYSOUT data set is closed.

CLOSE Example 2—DA Input

Close an input data set on a direct-access device.

When the data set was opened, it was identified by the following DD statement and DCB in the problem program:

```
//DD1     DD      UNIT=2314,DISP=OLD,VOL=SER=QQLIB,DSN=QFILE1

DCB1      DCB     DDNAME=DD1,BLKSIZE=80,LRECL=80,                 X
                  DSORG=PS,EROPT=ABE,MACRF=GM,RECFM=F,            X
                  EODAD=EOF1
```

When the CLOSE DCB1 macro was issued, the following Close modules were executed.

IGC00020	Close initial
IFG0200V	Read JFCB
IFG0200W	Give control to access method executor close executors

Control is passed to the SAM Close executor IGG0201Z, which returns control to:

IFG0200Y	Access method executor return

If this were an output data set being closed, the the following module would be executed:

IFG0201R	Update F1 DSCB
IFG0202E	Close DA final
IFG0202J	Restore DCB to pre-open status
IFG0202K	Delete DEB
IFG0202L	Close final

Returns control to issuer of the CLOSE macro.

CLOSE Example 3—SL Tape Input

Close an input tape data set with standard labels.

When the data set was opened, it was identified by the following DD statement and DCB:

```
//DD1     DD     UNIT=2400,DISP=OLD,VOL=SER=PS6342,               X
//               DSNAME=PARTJURN,LABEL=SL

DCB1      DCB    DDNAME=DD1,BLKSIZE=80,LRECL=80,                  X
                 DSORG=PS,MACRF=GM,RECFM=F,                       X
                 EODAD=EOF1
```

When the CLOSE DCB1 macro was issued, the following Close modules were executed:

IGC00020	Close initial
IFG0200V	Read JFCB
IFG0200W	Give control to the Close access method executors

Control is given to the SAM Close executor IGG0201A. When the executors complete their processing, they return control to:

IFG0200Y	Access method executor return
IFG0202F	Position tape
IFG0202J	Restore DCB to pre-open status
IFG0202K	Delete DEB
IFG0202L	Close final

Returns control to the issuer of the CLOSE macro.

Note: The same modules are executed when closing an output data set on tape.

APPENDIX A: OPEN AND CLOSE PROCESSING OF SPOOLED SYSIN/SYSOUT DATA SETS

During Open and Close processing of SYSIN and SYSOUT data sets, the data sets are associated with both a DCB and an ACB. The OPEN and CLOSE macros are issued by the problem program against the DCB; the Open and Close access method executors issue OPEN and CLOSE macros against an ACB that has the same attributes, as described in "OPEN Example 1—SYSIN" and "CLOSE Example 1—SYSIN" in the "Diagnostic Aids" section. In processing the ACB, Open calls a special access method executor, IGG0193K, that performs the functions required to prepare the DEB for use by a job entry subsystem (JES).

When the ACB is closed, a special access method executor, IGG0203K, is called to reverse the Open processing.

Descriptions of these modules follow.

IGG0193K: Open Access Method Executor for Spooled Data Sets Associated with an ACB

Function: Gets storage for the DEB, the subsystem identification block (SSIB), and the subsystem option block (SSOB). This routine initializes (1) the fields of the DEB, (2) the SSIB to identify the job entry subsystem being used, and (3) the SSOB to indicate that an Open is in process. With this initialization done, the IEFSSREQ macro is issued to pass control to the job entry subsystem. When control is returned from the JES, the DEB is chained from the TCB-DEB chain and its address is added to the DEB table. The ACB is then moved to the user's storage, and if there are no more ACBs to process for this Open request, control is passed to the next access method executor identified in the WTG table or returned to IFG0196W (IGG0190S).

Input: R5 ▶ Open parameter list, R6 ▶ WTG table, R7 ▶ current Open parameter list entry, and R8 ▶ the WTG table entry representing this ACB.

Output: A DEB that points to the ACB, with a valid address in the job step's DEB table. On error conditions encountered by the subsystem routines, the DEB, SSIB, and SSOB are freed and the busy bit in the ACBOFLGS field is set to zero.

IGG0203K: Close Access Method Executor for Spooled Data Sets Associated with an ACB.

Function: This routine builds a parameter list indicating that Close is in process and passes control to the job entry subsystem identified in the SSIB, via the IEFSSREQ macro. If there are no more ACBs to process for this Close request, control is passed to the next access method executor identified in the WTG table or returned to the Close access method executor return module, IFG0200Y (IGG0200G).

Input: R5 ▶ Open parameter list, R6 ▶ WTG table, R7 ▶ current Open parameter list entry, and R8 ▶ the WTG table entry representing this ACB.

Output: R5 and R6 are unchanged, R7 ▶ the next active entry in the parameter list, and R8 ▶ the first entry in the WTG table whose module ID is not zero.

INDEX

For additional information about any subjected listed in this index, refer to the publications that are listed under the same subject in the *OS/VS2 Master Index*, GC28-0693.

OS/VS2 Open/Close/EOV Logic (Fi ., S370-30) Printed in U.S.A. SY26-3827-2

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

OS/VS2 Open/Close/EOV Logic
SY26-3827-2

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

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

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ____________________

Previous TNL ________________

Previous TNL ________________

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SY26-3827-2

Reader's Comment Form

Fold and Tape

First Class Permit
Number 6090
San Jose, California

Business Reply Mail

No postage necessary if mailed in the U.S.A.

Postage will be paid by:

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and Tape

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

OS/VS2 Open/Close/EOV Logic (File No. S370-30) Printed in U.S.A. SY26-3827-2

OS/VS2 Open/Close/EOV Logic
SY26-3827-2

Reader's Comment Form

Note: Staples can cause problems in automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

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

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL __________________

Previous TNL __________________

Previous TNL __________________

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Reader's Comment Form

Fold and Tape

First Class Permit
Number 6090
San Jose, California

Business Reply Mail

No postage necessary if mailed in the U.S.A.

Postage will be paid by:

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and Tape

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601