

Program Logic

OS OPEN/CLOSE/EOV Logic

Release 21

Program Number 360S—DM—508

This publication, intended for system programmers and people who want to understand the Open/Close/End-of-Volume (EOV) portion of the operating system, describes the units of processing (functions) and units of code (modules) of Open/Close/EOV.

Open opens a data set by linking system control blocks so that the access methods can perform the input/output operations requested by the user. EOV applies these linkages when the end of a volume of a multivolume data set is reached and performs end-of-data processing. Close closes a data set by restoring the system control blocks to the contents they had before the data set was opened.

The book is divided into the following sections: Introduction, Method of Operation, Program Organization, Directory, Data Areas, and Diagnostic Aids. The latter section discusses how to use a set of routines called Problem Determination to find the causes of errors that occur during Open/Close/EOV processing.

The reader is expected to be familiar with OS data management, as described in *OS Data Management Services Guide*, GC26-3746.

Seventh Edition (March 1972)

This is a major revision of, and makes obsolete, GY28-6609-5. The manual has been completely rewritten to correspond to the redesign of Open/Close/End-of-Volume.

This edition applies to Release 21 of the operating system and to all subsequent releases unless otherwise indicated by new editions or technical newsletters. Changes to the information in this book may be made at any time, so before using this publication in connection with the operation of IBM systems, consult the latest *SRL Newsletter*, GN20-0360, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Forms for readers comments are provided at the back of this publication. If the forms have been removed, comments may be addressed to IBM Corporation, Programming Publications, Department D78, Monterey and Cottle Roads, San Jose, California 95114.

© Copyright International Business Machines Corporation 1968, 1969, 1970, 1971, 1972

PREFACE

This publication describes the functions (units of processing) and modules (units of code) of the Open/Close/End-of-Volume portion of the operating system. It also describes the relationship of Open/Close/EOV to the portions of the operating system that give control to and receive control from it. Open/Close/EOV modifies system control blocks, and mounts and verifies volumes so that I/O operations can be performed.

This publication is divided into seven sections:

“Section 1: Introduction.” The introduction briefly describes Open/Close/EOV processing.

“Section 2: Method of Operation.” Diagrams and supplementary text show the input, processing, and output of each function of Open/Close/EOV. The resident routine, which is used primarily to transfer control between modules, is discussed at length. The emphasis of the chapter is on the functional organization, not the modular organization, of the Open/Close/EOV portion of the operating system. The diagrams indicate the relationship between Open/Close/EOV and the other portions of the operating system that give control to and receive control from it, (the scheduler, the direct access device space management (DADSM) routines, the access method executors, and user-written routines).

“Section 3: Program Organization.” Module flow diagrams describe module processing in Open/Close/EOV, message numbers of messages issued by each module, register usage within each module, and major input and output of each module.

“Section 4: Directory.” Tables of cross-references between the functions in the “Method of Operation” section and the modules in the “Program Organization” section provide a link between the chapters when a particular module or function is being inspected.

“Section 5: Data Areas.” Illustrations describe the work areas used by Open/Close/EOV. The work areas described are the O/C/EOV work area, the JFCBMASK field of the job file control block (JFCB), the DCBOFLGS field of the data control block (DCB), the XCTL table, and the where-to-go table, which is also used by the access method executors.

“Section 6: Diagnostic Aids.” A cross-reference table of the ABEND messages issued by Open/Close/EOV and the functions and modules that issue them provides debugging information. Information is provided on how to use the Problem Determination routines and the diagnostic facilities of the Open/Close/EOV resident routine to provide additional debugging information for Open/Close/EOV.

How to Use This Book

Before using this publication, read the introduction to establish perspective. After that, use of the publication depends on the nature of the information you need.

If you want a general understanding of Open/Close/EOV, read the center block of each figure in the “Method of Operation” section. This center block describes the processing performed.

If you want a more comprehensive understanding, read the center block of each figure and study the right and left sides as the arrows direct you. The left side describes the input or conditions

for performing processing steps. The right side describes the output produced by processing steps. Remember that you begin reading each figure by reading the center, or processing block, not by reading the left side where the input is shown. Do not assume that the left and right sides show every field used as input or output; they show only the setting of fields that significantly affect continued processing.

If you look in the “Diagnostic Aids” section for the message number of a message you have received, you will find the names of the function and module that issued the message. Then turn to the index or the “Directory” to find where that function and module are discussed in the “Method of Operation” and “Program Organization” sections.

If you need further debugging help, consult “How to Use Problem Determination for Debugging” in the “Diagnostic Aids” section to see if a trace will help. If the traces provided are not adequate, consult “Resident Routine Trace” in the “Diagnostic Aids” section for information on how to create your own trace.

Prerequisite Reading

Before reading this publication, you should consult the following sections of *OS Data Management Services Guide*, GC26-3746:

“Opening and Closing a Data Set” for a summary of Open/Close/EOV processing.

“Magnetic Tape Labels and Direct-Access Labels” to familiarize yourself with the information contained in volume labels.

“Exits to Special Processing Routines” to familiarize yourself with the kinds of and purposes of these routines.

You should be generally familiar with how the operating system works, as discussed in *OS MVT Guide*, GC28-6720, and *OS MFT Guide*, GC27-6939.

You should also be familiar with the options of the OPEN, CLOSE, and FEOV macro instructions as described in *OS Data Management Macro Instructions*, GC26-3794.

Related Reading

Information on the portions of the operating system that transfer control to and receive control from Open/Close/EOV appears in the following publications:

OS BDAM Logic, GY28-6617

OS BTAM Logic, GY30-2001

OS DADSM Logic, GY28-6607

OS GAM Logic, GY27-7113

OS I/O Supervisor Logic, GY28-6616

OS ISAM Logic, GY28-6618

OS QTAM Logic, GY30-2002

OS SAM Logic, GY28-6604

OS TCAM Logic, GY30-2029

Other publications that may be of interest in conjunction with this manual are:

OS Data Management for System Programmers, GC28-6550, which discusses how to use OPENJ and RDJFCB.

OS Job Control Language Reference, GC28-6704, which describes JCL options that affect Open/Close/EOV processing.

OS System Control Blocks, GC28-6628, which describes fields used in Open/Close/EOV processing.

CONTENTS

iii	Preface
iii	How to Use This Book
iv	Prerequisite Reading
iv	Related Reading
xi	Summary of Changes for Release 21
xi	Reorganization
xi	Module Names
xi	Problem Determination
1	Section 1: Introduction
1	The Resident Routine
1	Open/Close/EOV Data Areas
1	Opening a Data Set
2	Closing a Data Set
2	End-Of-Volume Processing
5	Section 2: Method of Operation
5	Introduction to This Section
5	The Resident Routine
5	WAIT Condition
6	XCTL Condition
6	Synchronize Condition
6	Parallel Processing
8	Open Contents (Figure 1)
9	Open—Overview of Processing (Figure 2)
10	Open Initialization Processing and RDJFCB (Figure 3)
12	Open Direct-Access—Begin Direct-Access Processing (Figure 4)
14	Open Direct-Access—Continue Direct-Access Processing (Figure 5)
16	Open Tape—Mount Correct Volume and Determine Label Type (Figure 6)
18	Open Tape—Volume Label Processing (Figure 7)
20	Open Tape—Standard Input Label Processing (Figure 8)
22	Open Tape—No Label/Bypass Label Processing and Nonstandard Label Input and Output Interfaces (Figure 9)
24	Open SL Output Processing (Figure 10)
26	Open Security (Figure 11)
28	Open Merge (Figure 12)
30	Open—Access Method Executor Processing (Figure 13)
32	Open—Final Termination (Figure 14)
35	Close Contents (Figure 15)
37	Close—Overview of Processing (Figure 16)
38	Close Initialization (Figure 17)
40	Close Tape—Label Handling (Figure 18)
42	Close Tape—Volume Disposition Processing (Figure 19)
44	Close Direct Access Processing (Figure 20)
46	Close—SMF Record Building (Figure 21)
48	Close Final Processing (Figure 22)
51	TCLOSE—Overview of Processing (Figure 23)
52	TCLOSE Initialization (Figure 24)
54	TCLOSE Tape—Label Processing and Volume Positioning (Figure 25)
56	TCLOSE Direct Access—Input and Output Processing (Figure 26)
58	TCLOSE Final Processing (Figure 27)

61	EOV Contents (Figure 28)
63	EOV–Overview of Processing (Figure 29)
64	EOV Initialization (Figure 30)
66	EOV Tape Input–Process Current Volume (Figure 31)
68	EOV Tape Input–Process New Volume (Figure 32)
70	EOV Tape Output–Process Current Volume (Figure 33)
72	EOV Tape Output–Verify New Volume (Figure 34)
74	EOV Tape Output–New Volume Processing (Figure 35)
76	EOV Direct-Access Input–Mount New Volume(s) (Figure 36)
78	EOV Direct-Access Input–New Volume Processing (Figure 37)
80	EOV Direct-Access Output–Get More Space (Figure 38)
82	EOV Direct-Access Output–Processing With New Space (Figure 39)
85	Problem Determination Contents (Figure 40)
87	Problem Determination–Overview of Processing (Figure 41)
88	ABEND Interpretation and Recovery (Figure 42)
90	Optional Module Trace (Figure 43)
93	Legend
95	Section 3: Program Organization
95	Introduction to This Section
96	Open Processing
96	Open and OPENJ Initialization Processing (Chart 1 of 1) (Figure 44)
97	Open Tape Processing (Chart 1 of 6) (Figure 45)
98	Open Tape Processing (Chart 2 of 6) (Figure 46)
99	Open Tape Processing (Chart 3 of 6) (Figure 47)
100	Open Tape Processing (Chart 4 of 6) (Figure 48)
101	Open Tape Processing (Chart 5 of 6) (Figure 49)
102	Open Tape Processing (Chart 6 of 6) (Figure 50)
103	Open Direct-Access Processing (Chart 1 of 3) (Figure 51)
104	Open Direct-Access Processing (Chart 2 of 3) (Figure 52)
105	Open Direct-Access Processing (Chart 3 of 3) (Figure 53)
106	Open Security Processing (Chart 1 of 2) (Figure 54)
107	Open Security Processing (Chart 2 of 2) (Figure 55)
108	Open Merge Processing (Chart 1 of 2) (Figure 56)
109	Open Merge Processing (Chart 2 of 2) (Figure 57)
110	Open Access Method Executor Processing (Figure 58)
111	Open Final Processing (Figure 59)
112	Close Processing
112	Close Initialization Processing (Figure 60)
113	Close Tape Processing (Chart 1 of 2) (Figure 61)
114	Close Tape Processing (Chart 2 of 2) (Figure 62)
115	Close Direct-Access Processing (Chart 1 of 2) (Figure 63)
116	Close Direct-Access Processing (Chart 2 of 2) (Figure 64)
117	Close Final Processing (Figure 65)
118	SMF Data Set Processing (Figure 66)
119	TCLOSE Initialization Processing (Figure 67)
119	TCLOSE Tape Processing (Chart 1 of 1) (Figure 68)
120	TCLOSE Direct-Access Processing (Chart 1 of 1) (Figure 69)
121	TCLOSE Final Processing (Chart 1 of 1) (Figure 70)
122	EOV Processing
122	EOV Initialization Processing (Chart 1 of 3) (Figure 71)
123	EOV Initialization Processing (Chart 2 of 3) (Figure 72)
124	EOV Initialization Processing (Chart 3 of 3) (Figure 73)

125	EOV Tape Input Processing (Chart 1 of 3) (Figure 74)
126	EOV Tape Input Processing (Chart 2 of 3) (Figure 75)
127	EOV Tape Input Processing (Chart 3 of 3) (Figure 76)
128	EOV Tape Output Processing (Chart 1 of 4) (Figure 77)
129	EOV Tape Output Processing (Chart 2 of 4) (Figure 78)
130	EOV Tape Output Processing (Chart 3 of 4) (Figure 79)
131	EOV Tape Output Processing (Chart 4 of 4) (Figure 80)
132	EOV Direct-Access Input Processing (Chart 1 of 3) (Figure 81)
133	EOV Direct-Access Input Processing (Chart 2 of 3) (Figure 82)
134	EOV Direct-Access Input Processing (Chart 3 of 3) (Figure 83)
135	EOV Direct-Access Output Processing (Chart 1 of 4) (Figure 84)
136	EOV Direct-Access Output Processing (Chart 2 of 4) (Figure 85)
137	EOV Direct-Access Output Processing (Chart 3 of 4) (Figure 86)
138	EOV Direct-Access Output Processing (Chart 4 of 4) (Figure 87)
139	Shared Modules
139	Open, Close, and EOV Resident Routine (Figure 88)
140	Write-to-Operator Module (Figure 89)
141	Problem Determination Processing
141	Problem Determination (Chart 1 of 2) (Figure 90)
142	Problem Determination (Chart 2 of 2) (Figure 91)
143	Section 4: Directory
143	Function-to-Module Directory
143	Open (Including Open TYPE=J) Operation
146	Close (Including Close TYPE=T) Operation
148	EOV Operation
151	Common Functions
153	Section 5: Data Areas
153	O/C/EOV Work Area
154	Open, Close, and EOV Obtain an O/C/EOV Work Area for each DCB to be Processed (Figure 92)
157	Fields of the O/C/EOV Work Area Associated with the JFCB, ECB, IOB, DEB, and CCWs (Figure 93)
160	Fields in the O/C/EOV Work Area Used to Manipulate Other Fields (Figure 94)
161	JFCBMASK Field
164	DEBDCBMK Field
166	DCBOFLGS Field
166	Transfer Control (XCTL) Table
166	Format of the XCTL Table (Figure 95)
167	Where-To-Go (WTG) Table
167	Each Open or Close Operation Sets Up Its Own WTG Table (Figure 96)
168	WTGPATH Field
169	Section 6: Diagnostic Aids
169	ABEND Function-to-Module Cross-Reference
177	DCB Error Handling
177	Control Given to First Module of the Operation
177	Control Given to Executors
177	Control Returned from Executors
178	Control Given to Module Whose ID is in the Last Entry of WTG Table
178	Internal ABEND Codes
178	Open Operation
181	Close Operation

182	Close (TYPE=T) Operation
183	EOV Operation
185	Message-to-Function Cross Reference
195	Register Usage
196	How to Use Problem Determination for Debugging
196	ABEND Trace
196	Optional Work Area Trace
196	Formatting Traced Output
196	Resident Routine Module Trace
199	Section 7: Appendix
199	Abbreviations
203	Glossary
207	Index

SUMMARY OF CHANGES FOR RELEASE 21

Reorganization

Most of the changes to Open/Close/EOV for this release consist of repackaging the existing code to eliminate redundancy and needless transfers of control.

The Open/Close/EOV resident routine is a new module, resident in the nucleus. It waits for I/O operations to be completed, transfers control between Open/Close/EOV modules, and brings all DCBs to the same stage of processing before transferring control to a user-written routine or to another portion of the operating system. Most Open/Close/EOV modules branch to the resident routine when any one of these operations is to be performed.

Open/Close/EOV operations have been divided into functions, which are units of processing, not units of code. In some cases, a function has been put into a separate module, rather than made part of several modules, to reduce the number of load operations and to simplify the module flow logic.

Where similar processing is performed for different devices, the modules have been organized so that each module contains code applicable to only one device.

Module Names

All Open/Close/EOV module names have been changed and begin with IFG. Alias names beginning with IGG are retained where Open/Close/EOV transfers control to or receives control from other portions of the operating system.

Problem Determination

The new problem determination routines provide ABEND information, such as a trace of important control blocks, to help you diagnose errors that occur during Open/Close/EOV processing. Instructions for how to use problem determination are in the “Diagnostic Aids” section. The internal logic of problem determination is treated in the “Method of Operation” and “Program Organization” sections.

SECTION 1: INTRODUCTION

The Open, Close, and End-of-Volume operations are supervisor call (SVC) routines that reside in the SVC library (SYS1.SVCLIB) and operate out of the SVC transient areas. A program generally uses these routines by issuing a macro instruction (such as OPEN, CLOSE, FEOV, or RDJFCB) whose expansion includes an SVC instruction. Execution of this SVC instruction causes control to be passed to the SVC interruption handler, which determines the appropriate SVC routine to receive control.

The Resident Routine

The Open/Close/EOV resident routine, resident in the nucleus, waits for I/O operations to be completed, transfers control between Open/Close/EOV modules, and brings all DCBs to the same stage of processing before transferring control to a user-written routine or to another portion of the operating system. Most Open/Close/EOV modules branch to the resident routine when any one of these operations is to be performed. The resident routine is discussed in detail at the beginning of the “Method of Operation” section.

The resident routine can also be used for diagnostic purposes, as discussed in the “Diagnostic Aids” section.

Open/Close/EOV Data Areas

In addition to system control blocks, Open/Close/EOV obtains and uses three data areas.

An O/C/EOV work area is obtained for each DCB to be processed. This work area contains portions of system control blocks that apply to each DCB.

A where-to-go table is obtained for each OPEN or CLOSE macro instruction. This table identifies the next module needed to process each DCB being opened or closed. The access method executors obtain this work area for EOV processing.

A resident routine work area is obtained for each OPEN, CLOSE, or FEOV macro instruction. This work area is mainly a list of addresses of event control blocks (ECBs) that is used by the resident routine when it waits for I/O operations to be completed.

Opening a Data Set

A data set is opened by the execution of an OPEN macro instruction, with or without a specification of TYPE=J. Execution of an OPEN macro instruction that does not specify TYPE=J includes an SVC interruption (SVC 19) that causes the interruption handler to pass control to the Open routines. Execution of an OPEN macro instruction that specifies TYPE=J includes an SVC interruption (SVC 22) that causes the interruption handler to pass control to the OPENJ routine.

Open processing (performed by the Open and OPENJ routines) creates linkages between system control blocks so that the access method executors can perform the I/O operations requested by the processing program.

Much of Open processing is device-dependent, that is, it depends upon whether the data set is on magnetic tape or a direct-access device. Open processing mounts and verifies the volume supplied by the user, fills in the fields of the UCB that apply to the volume mounted on the unit it represents, and merges information from the tape label or DSCB and the JFCB into the DCB.

Both Open and OPENJ are type 4 SVC routines. The major difference between them is that Open reads the JFCB from the SYS1.SYSJOBQE data set into the O/C/EOV work area, while OPENJ moves the JFCB into the O/C/EOV work area from the user's region or partition after the user has gotten it there by issuing a RDJFCB macro instruction. This allows the user to modify the JFCB before the data set is opened. Execution of the RDJFCB macro instruction includes an SVC interruption (SVC 64) that causes the interruption handler to pass control to the Read JFCB routine in Open/Close/EOV. The OPEN TYPE=J and RDJFCB macro instructions are discussed in *OS Data Management for System Programmers*.

Closing a Data Set

The user closes a data set when he is finished reading or writing records from or to it. Execution of the CLOSE macro instruction includes an SVC interruption (SVC 20) that causes the interruption handler to pass control to the Close routines.

The Close routines restore the fields of the DCB associated with the data set to the contents they had before the DCB was opened (Open modifies several DCB fields); process labels and DSCBs; reposition the volume; and release the DEB and access method executors.

A data set can be temporarily closed if BSAM is used. Temporary closing requires the execution of a CLOSE macro instruction that specifies TYPE=T. Execution of this macro instruction includes an SVC interruption (SVC 23) that causes the interruption handler to pass control to the TCLOSE routine.

TCLOSE differs from Close in that TCLOSE does not restore the DCB fields, or release the DEB or access method executors. TCLOSE performs only label processing and volume repositioning. Therefore, when TCLOSE closes a data set, the user may resume input and output operations on that data set without reopening it.

End-Of-Volume Processing

Open processing creates linkages between system control blocks so that the access method executors can perform the I/O operations requested by the processing program. Close processing undoes these linkages when the processing program is finished with I/O operations. EOVS processing applies the linkages created by Open processing when the end of a volume of a multivolume data set is reached.

EOVS performs the final processing on a data set and mounts and verifies additional volumes, checking or building a data set label for each one. EOVS processes only one DCB at a time.

The user can force an end-of-volume condition by issuing an FEOV macro instruction. The expansion of the FEOV macro instruction includes an SVC interruption (SVC 31) that causes the interruption handler to pass control to EOVS. The operating system issues an SVC 55 instruction (for EOVS) when either the Check routine of BSAM or a synchronizing routine of QSAM determines that the volume has ended but the data set has not, or that the data set has ended.

The first module loaded for an SVC 31 instruction is the FEOV executor. The first module loaded for an SVC 55 instruction is the SYNAD/EOV executor. Both of these executors are discussed in *OS SAM Logic*. The next module loaded for either SVC is the first module of EOV.

SECTION 2: METHOD OF OPERATION

Introduction to This Section

In this section, Open/Close/EOV is divided into functions. A function is a unit of processing, while a module is a unit of code. The emphasis in this chapter is on the flow of control between functions.

The size of a function varies. It might be 1K, which is the size of modules that execute in the SVC transient area under OS MVT, or it may be less or greater than 1K.

The functions of Open/Close/EOV are illustrated by diagrams. Input to the functions and output from the functions provide information for a detailed understanding of Open/Close/EOV. How to use these diagrams and how these diagrams relate to the rest of the book is discussed in "How to Use This Book," which is included in the Preface. A legend that shows the meaning of symbols used in this section's diagrams appears on a foldout page following the last diagram.

This chapter is intended for people who want an understanding of how all or portions of Open/Close/EOV work and how Open/Close/EOV relates to the rest of the operating system.

The Resident Routine

The resident routine of Open/Close/EOV is a 504-byte module located in the nucleus. It is included in the nucleus by the system generation inner macro SGIEA3IC and is called by Open/Close/EOV under three conditions:

- A WAIT macro is to be issued for an I/O operation associated with one or more DCBs being processed.
- An Open/Close/EOV module has finished processing a particular DCB and an XCTL macro is to be issued to transfer control to the next Open/Close/EOV module required.
- A point is reached in the code at which an Open/Close/EOV module must transfer control to a module outside Open/Close/EOV. This situation arises when control is passed to the access method executors, DADSM routines, or user-written routines. Each module must finish processing all DCBs for an OPEN or CLOSE macro instruction before the module outside Open/Close/EOV can be given control.

Three entry points, at offsets of 0, 4, and 8 bytes from the beginning of the resident routine, are provided for these three situations. A module calls the resident routine by issuing an ICERES macro instruction. This macro generates a branch to the resident routine at the appropriate offset. The resident routine saves the contents of the registers of the calling Open/Close/EOV module in the last part of the O/C/EOV work area.

WAIT Condition

When an Open/Close/EOV module issues an IECRES macro instruction for a WAIT condition, a branch is taken to the resident routine, which saves the contents of registers 9 to 14, 0 and 1 in the O/C/EOV work area. The resident routine sets a bit in the O/C/EOV work area ($DXRESIND=X*80'$) to indicate that the DCB has a WAIT pending.

If there are any DCBs being processed by the module in control without a WAIT pending, and if two or more DCBs are being processed, the resident routine restores the register contents of a DCB that does not have a WAIT pending and branches back to the module that issued the IECRES macro instruction. A DCB is being processed by the module in control if the module ID in that DCB's entry in the where-to-go table matches the current module ID in the where-to-go table. See the "Data Areas" section for a description of and a discussion on the use of this table.

If an ECB is posted complete before the resident routine can issue a WAIT for it, no WAIT is issued. Thus, an SVC trace table can show EXCP entries for which no WAITs have ever been issued.

The resident routine builds an ECBLIST in the Resident routine work area and, if all DCBs being processed by the current module have WAITs pending and have not had their ECBs posted complete, issues a multiple WAIT. As soon as one ECB is posted complete, the resident routine restores the register contents of the module that called it, resets the wait-pending bit (DXRESIND) in the O/C/EOV work area, and branches back to the module that called it.

XCTL Condition

When a module has finished processing a DCB, it issues the IECRES macro instruction to branch to the resident routine. The IECRES macro moves the next module's relative track address ID and (TTR) into the where-to-go table entry for that DCB, and control is transferred to that module. When control is returned, the resident routine searches for other DCBs that need to be processed by that module. When such a DCB is found, processing is similar to WAIT condition processing, described under "WAIT Condition".

If no other DCBs need to be processed by that module, the resident routine examines the where-to-go table for the module with the lowest remaining ID in collating sequence, and moves the ID and TTR of that module into the current where-to-go table entry. (A module ID of binary 0 indicates that the DCB is not to be processed.)

After the register contents for the first DCB to be processed by the next module have been restored, the resident routine issues an XCTL macro instruction to transfer control to the module whose ID and TTR appear in the current where-to-go table entry.

Synchronize Condition

Before control can be transferred to a module that is not in Open/Close/EOV, such as an access method executor, a DADSM routine, or a user-written routine, the DCBs being processed by a module must be brought to the same stage of completion. To do this, a module branches to the resident routine using the IECRES macro instruction. The resident routine sets the synch bit in the O/C/EOV work area (DXRESIND=X'40'). The resident routine then continues in the same manner as for the WAIT condition with one exception: when all DCBs being processed by the current Open/Close/EOV module have the synch bit on, the resident routine returns directly to the calling module. The calling Open/Close/EOV module then transfers control to the outside module.

Parallel Processing

The resident routine allows a module to perform the same I/O operation on different DCBs at the same time; for example, a module can read tape labels for all tape DCBs. The resident routine

also allows a module to perform different I/O operations on different DCBs; for example, a **REWIND** can be issued for one DCB and a forward space file for another.

A module can overlap I/O operations with transfers of control (using the **XCTL** macro instruction) by issuing the **IECRES** macro instruction. It causes a transfer of control to the next module but causes that module to wait for the I/O to be completed. I/O operations are also overlapped between tape and direct-access DCBs.

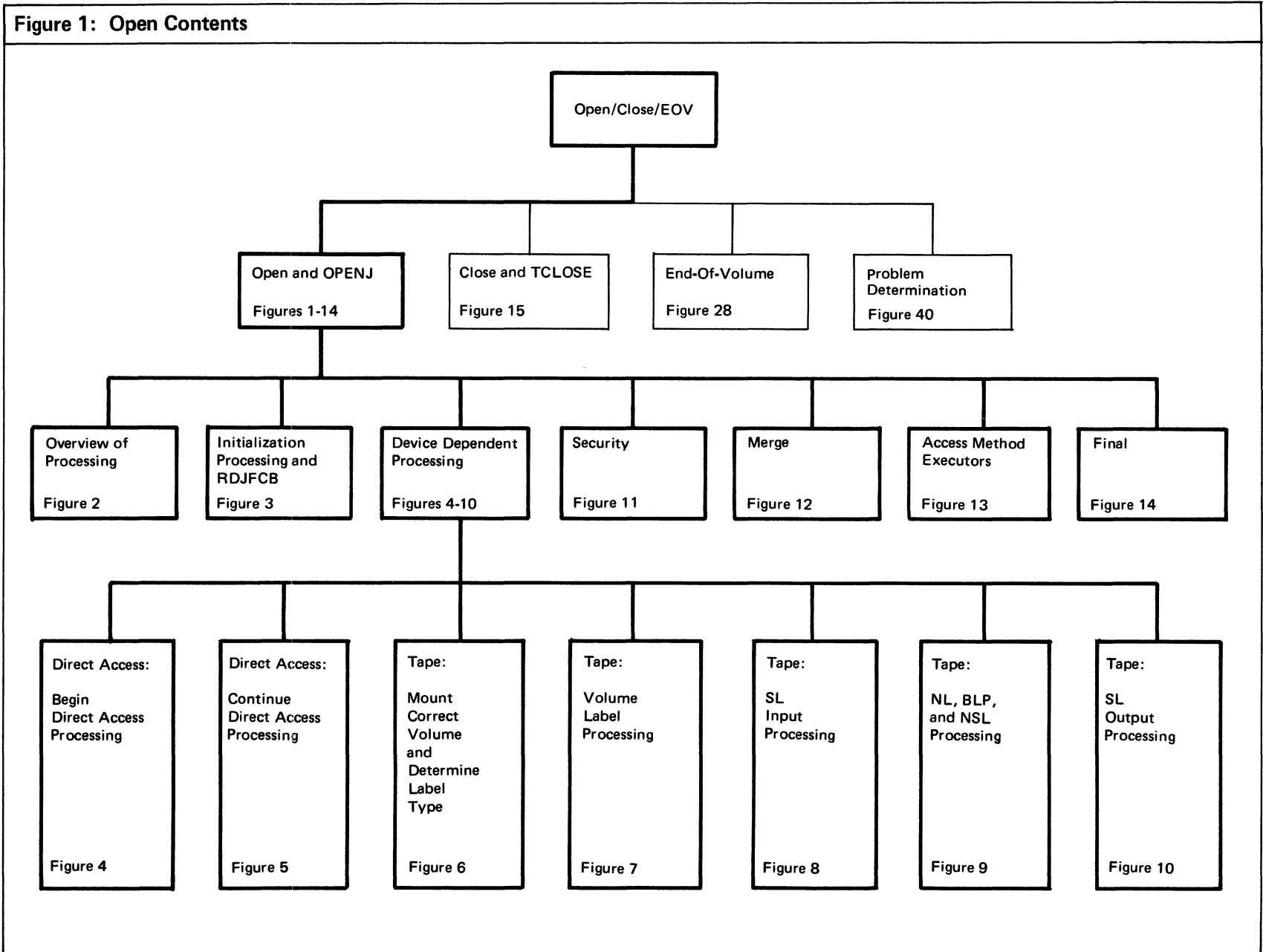


Figure 2: Open – Overview Of Processing

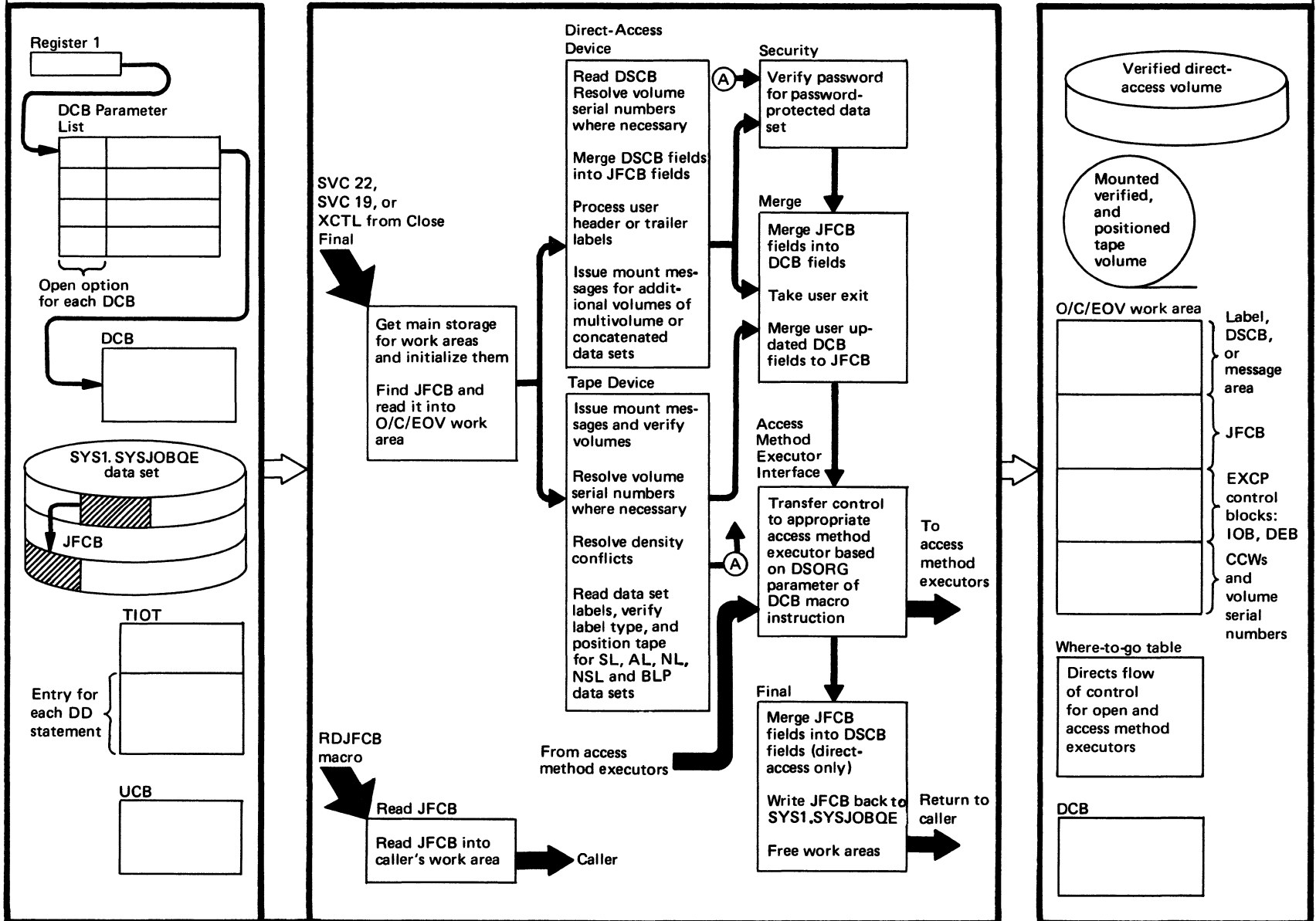


Figure 3: Open Initialization Processing and RDJFCB

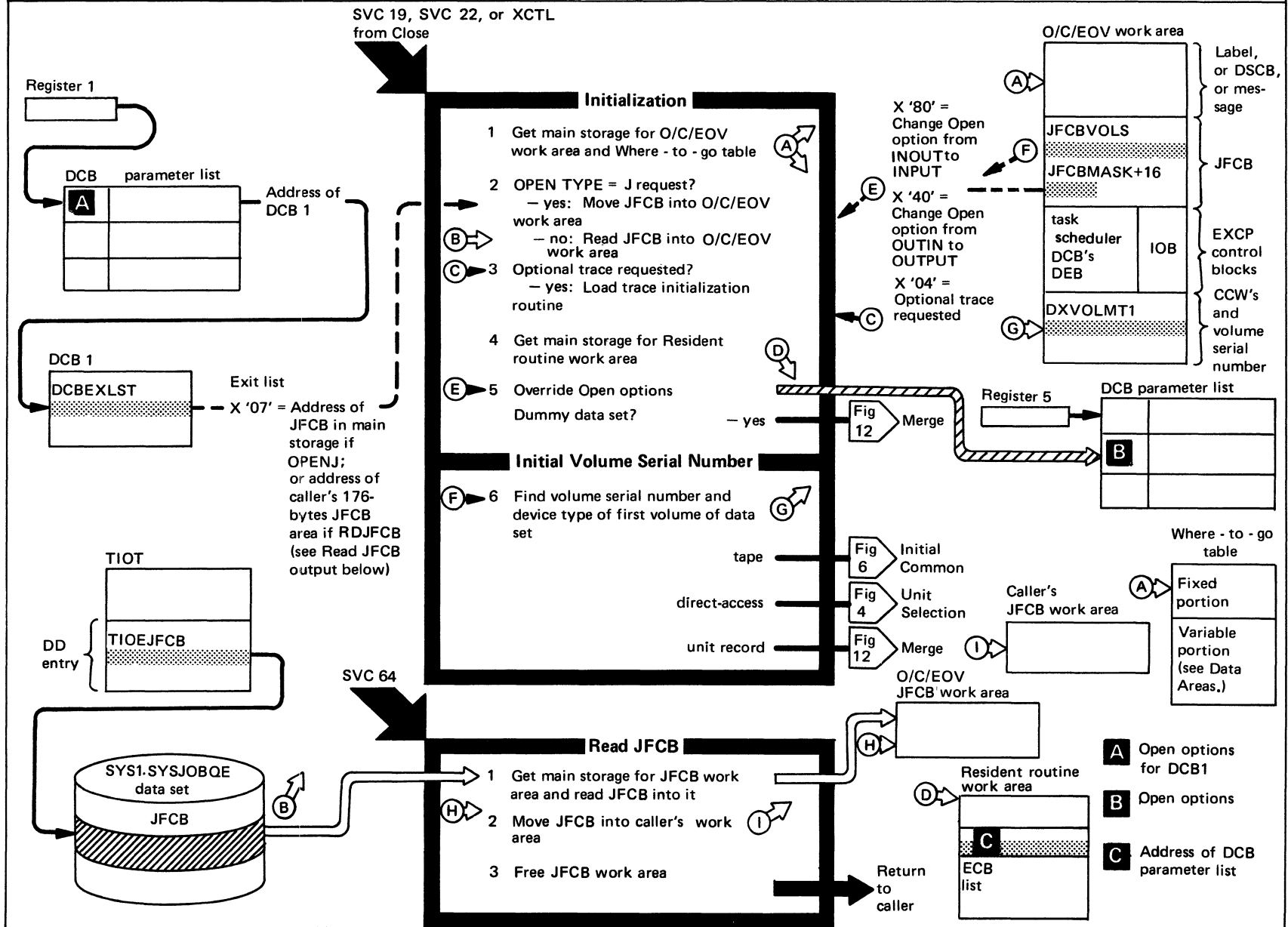


Figure 3. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 When Open is entered from the SVC interruption handler, interruptions are disabled. Open enables interruptions and continues to operate with interruptions enabled.</p> <p>Register 1 points to the DCB parameter list. Each entry in the list contains an Open option in the high-order byte and a DCB address in the three low-order bytes. The valid options are:</p> <pre> ...00.... DISP ...01.... REREAD ...11.... LEAVE ...0000 INPUT ...1111 OUTPUT ...0011 INOUT ...0111 OUTIN ...0001 RDBACK ...0100 UPDAT </pre> <p>The lock bit in DCBOFLGS is tested to see if the DCB has already been processed by Open/Close/EOV. If the lock bit is on, the next DCB in the parameter list is processed.</p> <p>Register 12 is zeroed and used to count the number of DCBs in the user's DCB parameter list. This number is used to calculate the Where-to-go table length before a GETMAIN macro instruction is issued for that area.</p> <p>A GETMAIN macro instruction is issued for 488 (1E8) bytes for the O/C/EOV work area. The control blocks needed by Open for I/O operations, primarily the DEB, IOB, and ECB, are set up in this work area. Open/Close/EOV used the EXCP access method for I/O operations</p>	<p>IGC00011 for Open</p> <p>IGC0002B for OPENJ</p> <p>IGC00011</p> <p>IGC00011</p> <p>IGC00011</p>	<p></p> <p>0IN10200</p> <p>0IN10200</p> <p>0IN11600</p>

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>2 Open obtains the JFCB by reading it in from the SYS1.SYSJOBQE data set, but OPENJ obtains the JFCB by moving it from an area pointed to by the address that follows a X'07' entry in the DCB exit list.</p> <p>3 If the optional trace was requested, a bit is set in the WTGPATH field of the Where-to-go table when the routine is loaded.</p> <p>4 The resident routine work area is two words larger if the optional trace was requested. From this point on, the resident routine is used to supervise XCTL, WAIT and SYNCH operations. (See "Resident Routine" earlier in this chapter for a discussion of these operations.)</p> <p>5 The INOUT or OUTIN option is overridden in the user's parameter list and restored at the end of successful Open processing.</p> <p>A dummy data set is indicated if the UCB address in the TIOT DD entry is zero or the DSNAME in the JFCB is NULLFILE. If the DCB is for a dummy data set, JFCBMASK+4 is set to X'20'. Open merge is then given control since there are no volumes or labels to process for dummy data sets.</p> <p>6 If a volume sequence number is specified for a physical sequential data set, it must be less than or equal to the number of volume serial numbers specified in the DD statement. This requirement prevents Open from reading JFCB extensions that do not exist and also prevents nonspecific requests for any but the first volume.</p>	<p>IGC00011 for Open</p> <p>IGC0002B for OPENJ</p> <p>IGC00011</p> <p>IFG0193A</p> <p>IFG0193A</p> <p>IFG0193A</p>	<p>0IN13200</p> <p>0IN22000</p> <p>0IN13200</p> <p>0IN31600</p> <p>0IN33800</p> <p>0IN34600</p>

Figure 4: Open Direct-Access – Begin Direct-Access Processing

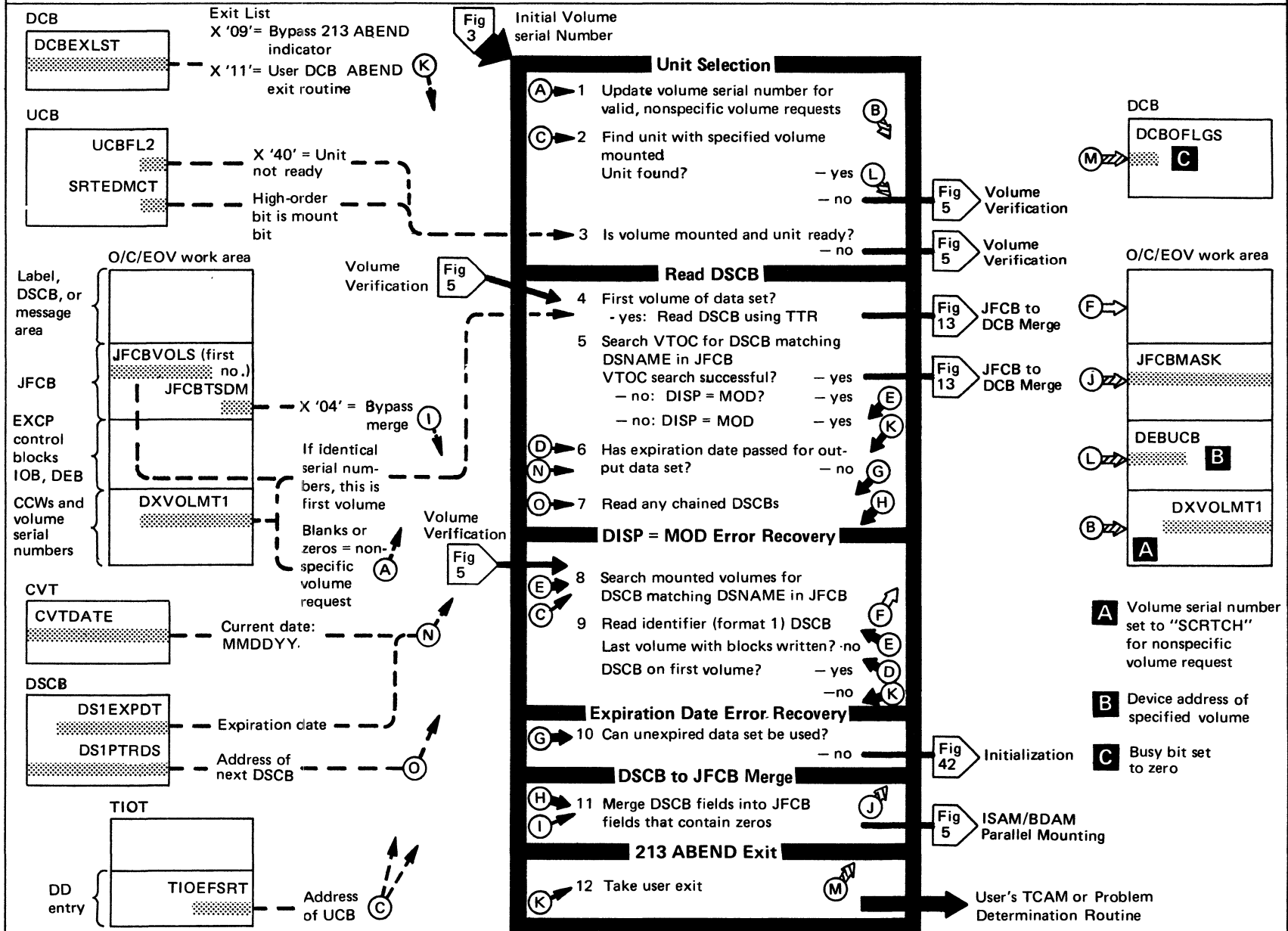


Figure 4. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 An Open option of INPUT, INOUT, or RDBACK is invalid for a nonspecific volume request because the data set could not yet exist on an unspecified volume. Each DD statement in the job step has a DD entry in the TIOT. The UCB pointers in each DD entry in the TIOT are scanned to find a unit that already has the specified volume mounted on it.	IFG0194E	ODA00600
	IFG0194E	ODA01600
3 The mount bit in the UCB is tested to determine whether the volume is mounted. The UCB not-ready bit (UCBFL2=X'40') is tested to see if the unit is ready. This check verifies that the volume was not changed before the unit was ready. If any of the following conditions are met, the Volume Verification function on Figure 5 is given control:	IFG0194E	ODA02000
	IFG0194E	ODA02000
	IFG0194E	ODA02000
<ul style="list-style-type: none"> ● volume serial number not found during UCB scan ● mount bit on ● UCB not-ready bit on 		
4 The DSCB TTR in the JFCB is checked to make sure it is larger than the beginning of the VTOC.	IFG0195A	ODA10800
5 If this volume is the second volume of a multi-volume data set, the entire VTOC is searched for a key equal to the DSNAME in the JFCB.	IFG0195A	

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
7 Each chained DSCB is read into its own 144-byte work area. The first DSCB work area is chained from DSCCORE in the O/C/EOV work area.	IFG0195A	ODA12000
8 If DISP=MOD is specified for a physical-sequential data set and the volume serial number selected was for the last volume of the data set, the data set might not extend that far yet. This function searches all the volumes assigned to the data set, beginning with the first volume, to find the end of the data set.	IFG0195E	ODA24100
10 A message is sent to the operator's console requesting permission to use the unexpired data set.	IFG0195G	ODA90400
11 The merge is done only for format 1 DSCBs and is bypassed if the inhibit bit (JFCBTSDM=X'04') is on. The DSCB fields merged into zero JFCB fields are: RECFM OPTCD KEYLEN DSORG BLKSIZE (except for ISAM load mode) LRECL RKP BFALN The JFCBMASK field contains a record of the fields merged.	IFG0195A	ODA12600

Figure 5: Open Direct-Access – Continue Direct-Access Processing

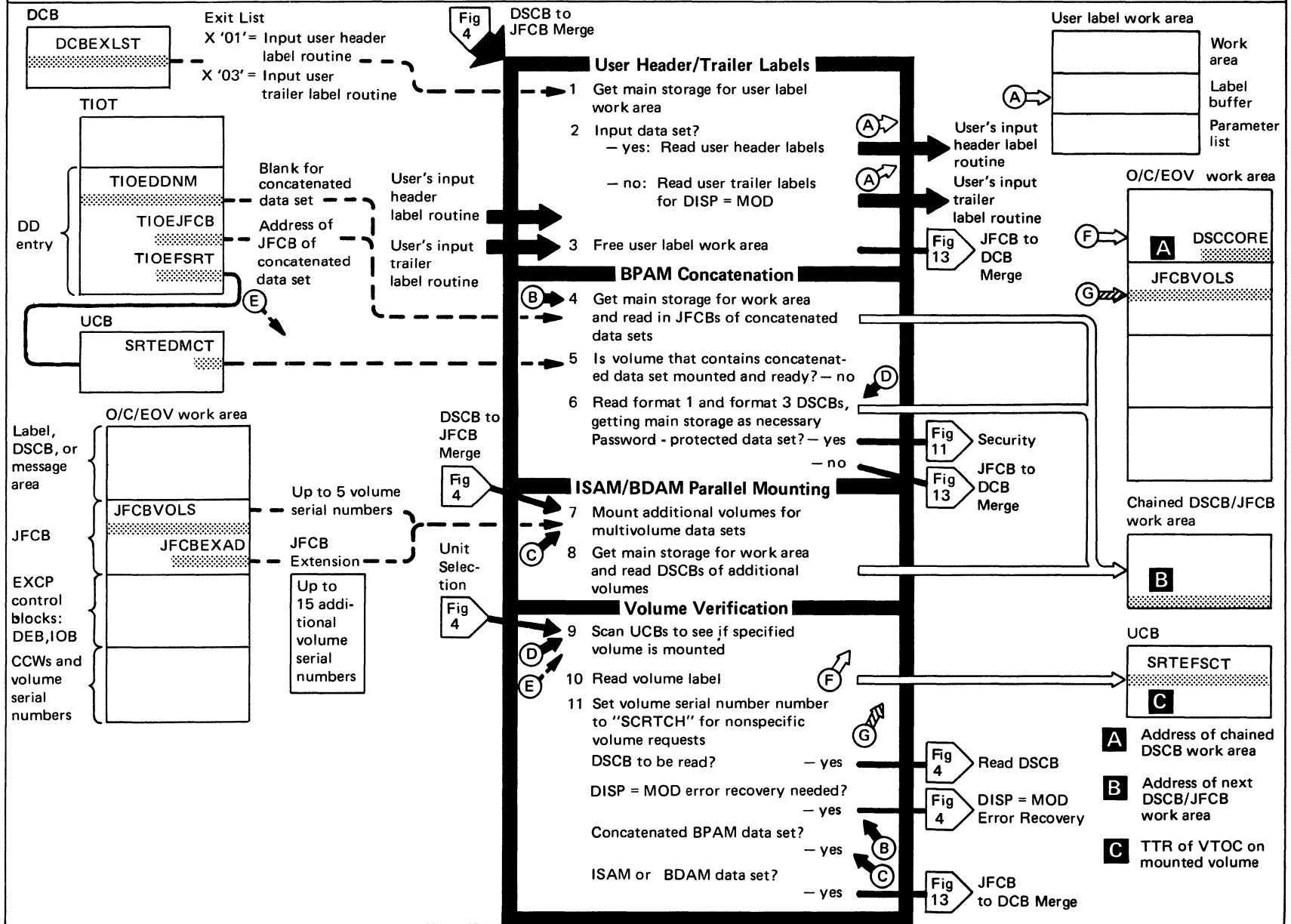


Figure 5. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>2 If the user has specified user header or trailer labels for an input data set, Open reads an input label into the label buffer and passes a parameter list (which includes a pointer to the label buffer) to the user's input label routine. This routine processes the input label and passes back a return code to indicate whether it wants more input labels read.</p> <p><u>Data Management Services Guide</u> describes how to write user label routines</p> <p>4 The TIOT contains a DD entry for each DD statement. This figure illustrates only one entry.</p>	IFG0554L	ULSYNCH

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>6 If JFCBIND2 in the JFCB is set to X'10', a password is required to read from or write to the data set. If the field is set to X'30', a password is required to write to the data set, but not to read from it.</p> <p>12 This exit gives the user the option of changing a DSNNAME or volume serial number and resuming processing for this DCB in the Unit Selection function (Figure 4).</p>	IFG0195M IFG0195G	 ODA92600

Figure 6: Open Tape --- Mount Correct Volume And Determine Label Type

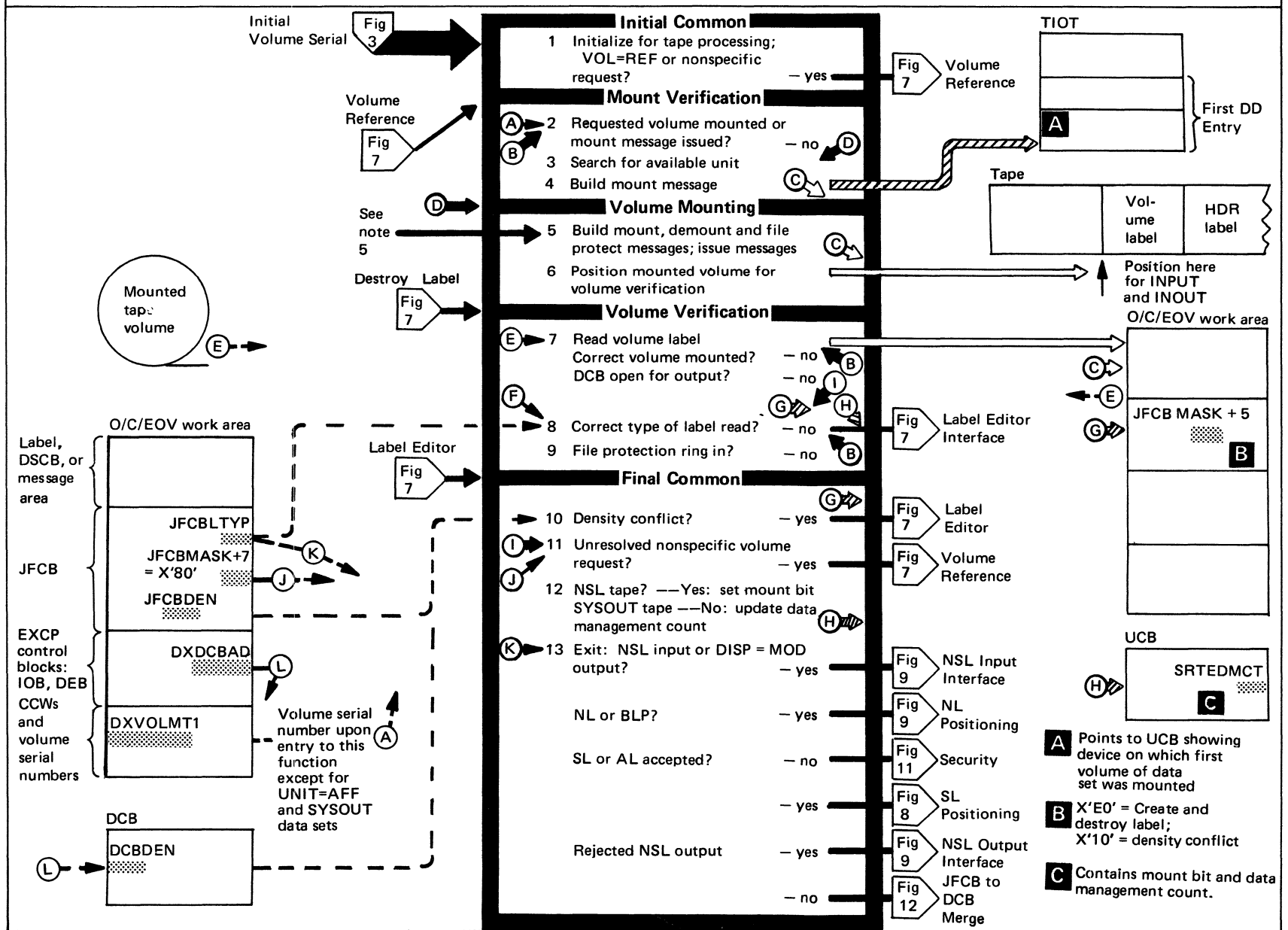


Figure 6. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 If the file sequence number in the JFCBFLSQ field has not been specified, it is set to the default value of 1.	IFG0193B	OTA00050
If volume statistics are to be kept, the Systems Management Facility (SMF) counters and flags are cleared.	IFG0193B	OTA00300
If an NL tape was specified, the reduced error recovery bit DXDEBOFG=X'01' is set in the DEB of the O/C/EOV work area.	IFG0193B	OTA00400
If the tape density in the JFCBDEN field is different from the tape density specified in the DCBDEN field, the DCBDEN value is used. Subsequent tape processing may reveal that the tape density specified in the DCB is wrong. This situation produces the density conflict discovered during the Final Common function.	IFG0193B	OTA00800
If American National Standard labels are specified, option code Q is added to the JFCBOPTCD field.	IFG0193B	OTA01500
If an error is detected for an American National Standard tape, a demount message is built. If a nonspecific request was made, the volume is remounted. Otherwise, the tape is unloaded, UCB fields are reset, the previous mount message is deleted, the demount message is issued, and the appropriate internal ABEND code is passed to the ABEND Interpretation and Recovery functions of Problem Determination	IFG0193B	OTA02600

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 An available unit has these characteristics: it is not reserved, does not have a permanently resident volume mounted on it, and has a data management count in the UCB (SRTEDMCT field) of zero. The data management count is a record of how many DCBs have been opened on that device.	IFG0194F	OTA07000
5 This function is entered from the following functions: NSL Ouptut Interface Figure 9 Label Editor Figure 8 SL Output Security Figure 10 SL Output Rewrite Volume Label Figure 10	IFG0194G	IFG0194G
11 A nonspecific volume request will not be resolved at this point if a referenced JFCB contained a pseudo volume serial number (indicated by (FFTR) instead of a real volume serial number.	IFG0194I	OTA16200

Figure 7: Open Tape – Volume Label Processing

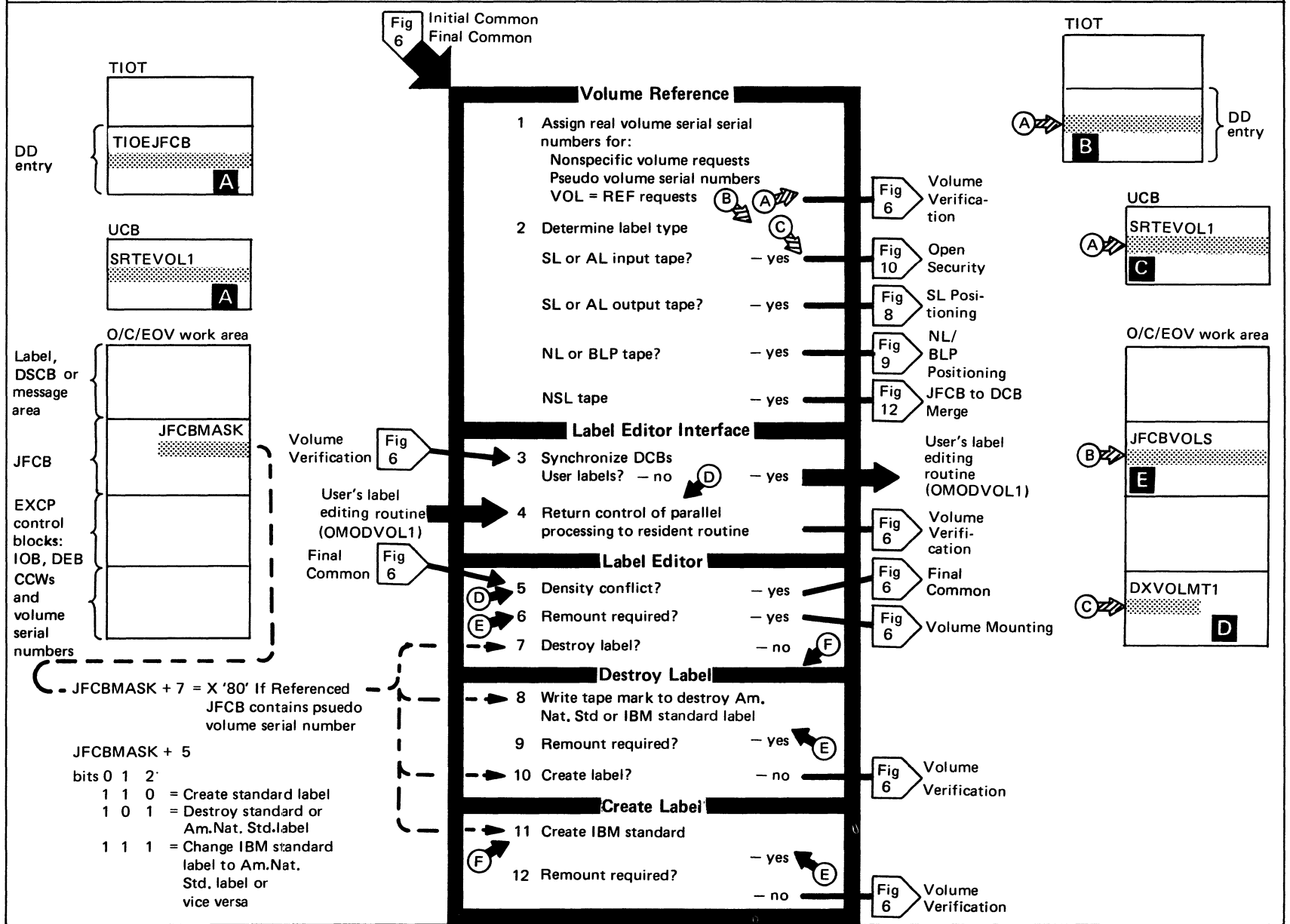


Figure 7. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 A pseudo volume serial number is X'FF' followed by the TTR for the JFCB of the data set being opened.	IFG0149D	OTA03800
After a referenced JFCB has been read for a VOL=REF request, a test is made to determine whether it contains a pseudo volume serial number or a real volume serial number. If it contains a pseudo volume serial number, JFCBMASK+7 is set to indicate that this function must be executed again.	IFG0194D	OTA03900
If the referenced JFCB contains real volume serial numbers, the last volume serial number is placed in the original JFCB at JFCBVOLS and in the O/C/EOV work area at DXVOLMT1. JFCB extensions are read in, if necessary, to find the last volume serial number.	IFG0194D	OTA04500
This function is entered a second time if a referenced JFCB contains a pseudo volume serial number. The referenced JFCB is read and the volume serial number for the volume that was just verified is placed in the JFCB at JFCBVOLS. The JFCB is then written back to the SYS1.SYSJOBQE data set.	IFG0194D	OTA03200
3 After the DCBs are synchronized, that is, all DCBs being processed in parallel are processed up to the same point, the resident routine pointer is saved in register RD. The DCB parameter list pointer is loaded into register RPAR, and this function transfers control to OMODVOL1.	IFG0194J	OTA18800

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
4 This step is only executed if the user has provided a label editor (OMODVOL1).	IFG0194J	OTA19100
5 This step is only executed if the user did not provide a label editor.	IFG0193C	OTA19500
JFCBMASK+5 is checked to determine if entry is due to a label error or density conflict.	IFG0193D	OTA20400
6 A read is issued to obtain a data set label. If a data cannot be read, a remount is required.	IFG0193D	OTA20400
11 If a label has been destroyed, another must be created to take its place. If an American National Standard label is destroyed, for instance, an IBM standard label is created.	IFG0193D	IFG0193D
If NL or NSL was requested, the tape is rewound and a tapemark is written. If there is an I/O error while writing the tapemark, an error message is written and an exit is taken to the Label Editor function for a remount.	IFG0193D	OTA21100
After the tapemark is written, a message giving the tape attributes is issued. An exit is then taken to the Volume Verification function to reprocess the DCB.	IFG0193D	OTA21200

A Same pseudo volume serial number means unit assigned to scratch data set

B Pseudo volume serial number from UCB

C First unit address from.TIOT

D = "SCRATCH" for nonspecific requests
 = Last volume serial number for VOL=REF requests

E For nonspecific volume requests and pseudo volume serial numbers, the volume serial number assigned to the data set is that of a scratch volume.

For VOL=REF requests where the referenced JFCB contains volume serial numbers, the volume serial number assigned to the data set is the last volume serial number in the referenced JFCB or JFCB extension.

Figure 8: Open Tape – Standard Input Label Processing

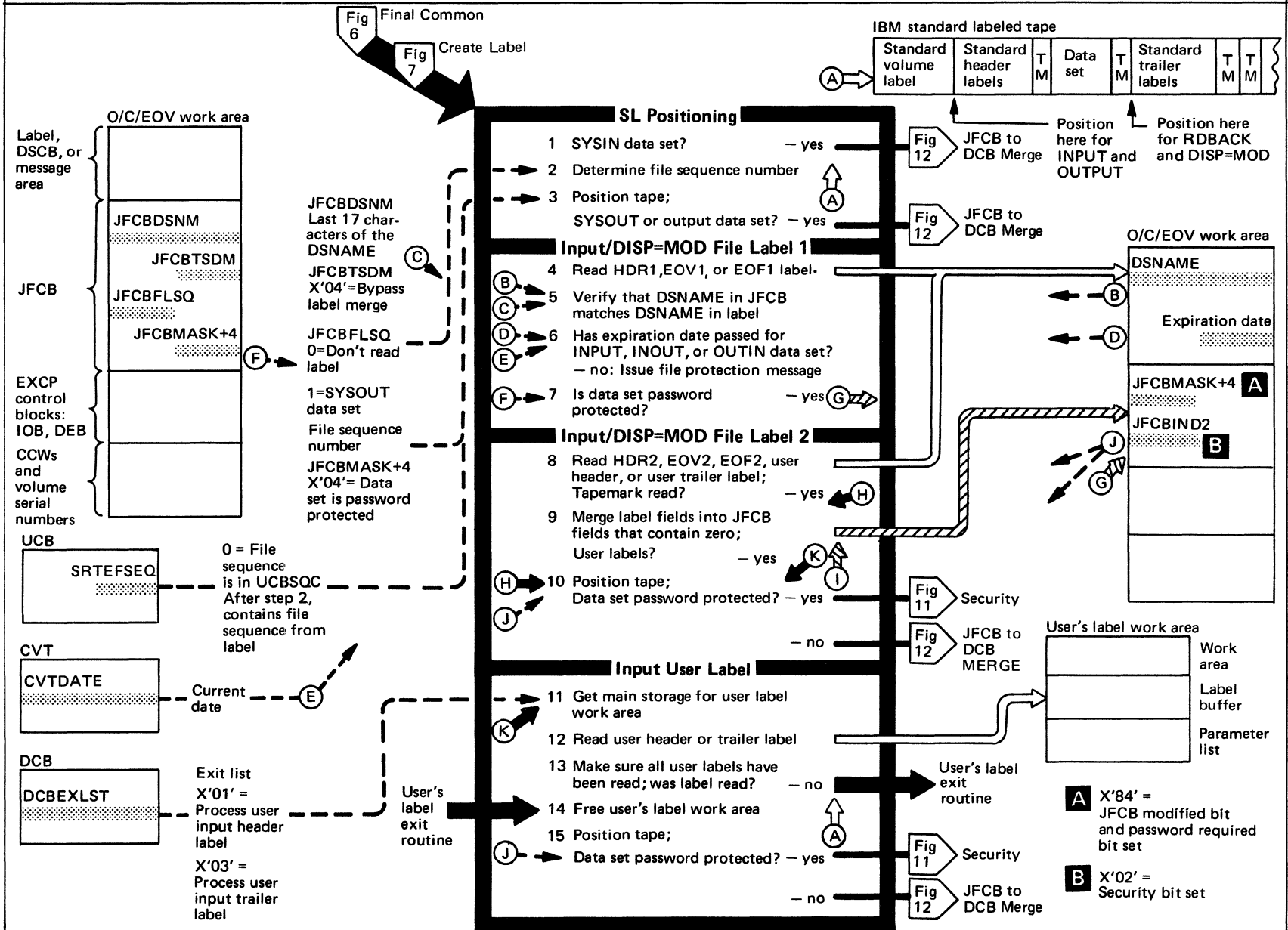


Figure 8. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 No positioning is required for SYSIN data sets. 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.	IFG0195B	OTA25000
2 The JFCB file sequence (JFCBFLSQ) field is set to one for SYSOUT data sets, because there is one SYSOUT data set opened for DISP=MOD that is continually being modified. If the file sequence number in the UCB (SRTEDMCT) is not zero, the file sequence must be obtained from the first header label. If the file sequence number in the UCB is zero, the file sequence number and count are set to one. If the JFCB file sequence number is greater than one, the header label for the first data set is read in and the file sequence number from the label is placed in the UCB.	IFG0195B IFG0195B	OTA25200 OTA25400 OTA25500
3 For Input or Output data sets, the UCB file sequence number is set equal to the JFCB file sequence number For RDBACK INPUT data sets or DISP=MOD OUTPUT data sets, the UCB file sequence number is set equal to the JFCB file sequence number plus one.	IFG0195B IFG0195B	OTA26600 OTA26900
5 The last seventeen characters of the DSNNAME in the JFCB are compared to the DSNNAME in the label. If they do not match and the data set is a generation data set, the tape may have been created under DOS. After the DSNNAME has been verified, the block count is moved from the label to the DCB After the DSNNAME 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).	IFG0195H IFG0195H	OTA28800 OTA29300

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
6 If the data set is open for output with DISP MOD and the expiration date has not passed, a message asks the operator to decide whether to accept the tape or not. If the operator replies "M," an exit is taken to Problem Determination to terminate processing. If the operator replies "U," the data set is checked to see if it is password protected.	IFG0195H	OTA29500
7 If the security byte on an ASCII volume is not blank and does not indicate read and write security, the volume cannot be used. An ASCII accessibility error message is issued. If the security byte in the label indicates a protected data set with American National Standard or IBM Standard labels, the security bit in the JFCB (JFCBIND2=X'10') is set, as are the JFCB modified bit (JFCBMASK+4=X'80') and the password required bit (JFCBMASK+4=X'04'). An exit is taken to the Input/MOD File Label 2 function if a data set is password protected.	IFG0195K IFG0195K	OTA30800 OTA31200
8 This step merges the second file label with the information in the JFCB If a tapemark is read instead of a second file label, the tape was created under DOS and control is passed to the Input User Label function. If the label read is a user header or user trailer label, the tape was created under DOS. The tape is backspaced over the user label.	IFG0195K IFG0195K IFG0195K	OTA32300 OTA32300 OTA31500
9 After a second file label has been read, a test of the JFCB full switch (JFCBTSDM) is made. If it is on (X'04'), the merge is bypassed. The merge is also bypassed for an ASCII volume not created under OS.	IFG0195K	OTA32100
10 If the password-required bit (JFCBMASK+4=X'04') is set, an exit is taken to the Security function to read the password.	IFG0195K	OTA34500

Figure 9: Open Tape – No Label/Bypass Label Processing and Nonstandard Label Input and Output Interfaces

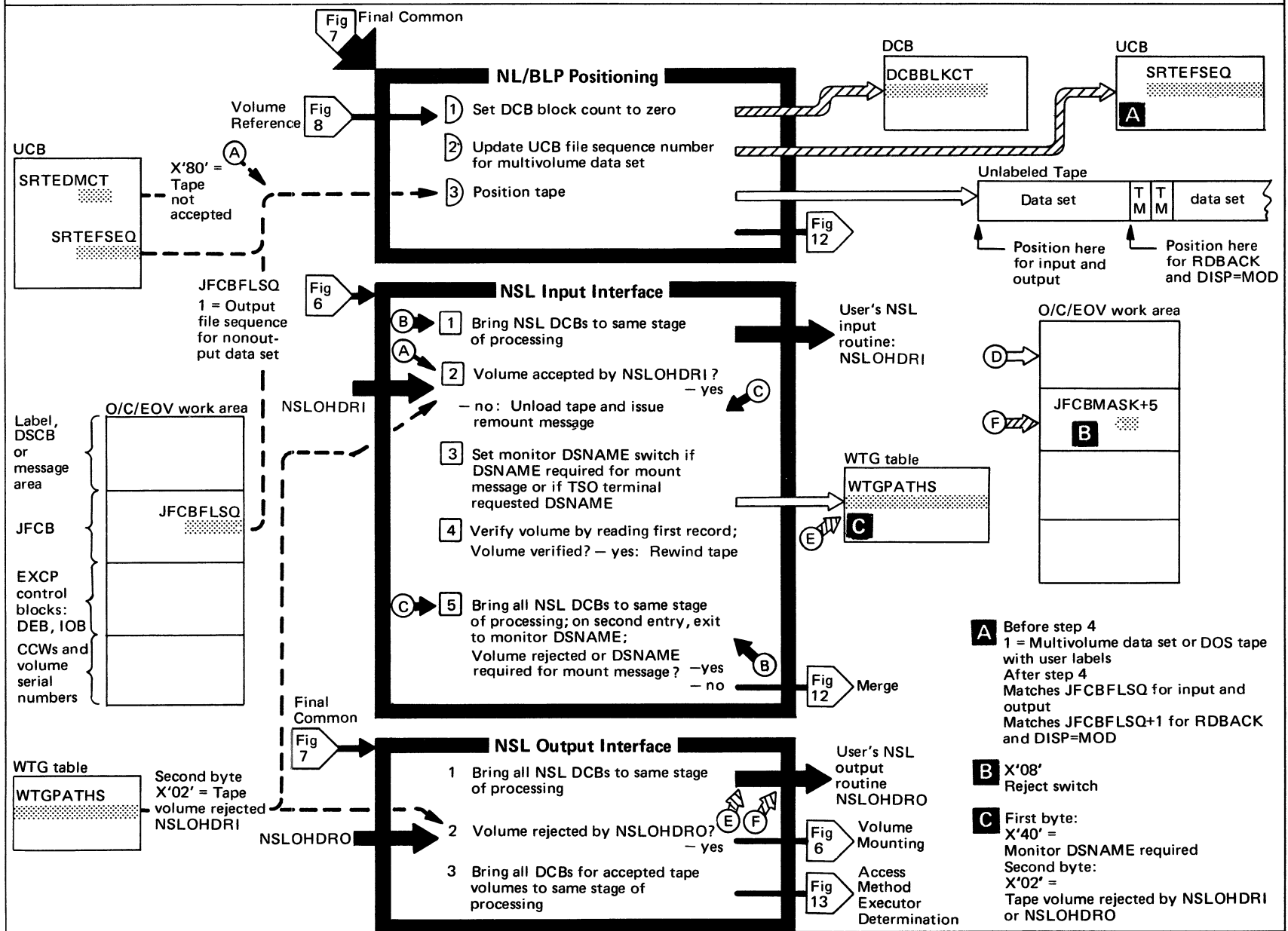


Figure 9. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>③ For INPUT and OUTPUT data sets, the UCB file sequence number is set equal to the JFCB file sequence number.</p> <p>For RDBACK and DISP=MOD data sets, the UCB file sequence number is set equal to the JFCB file sequence number plus 1.</p>	IFG0195C	OTA50700
<p>① If this is the first entry, this function saves registers RTIOT and RUCB in the O/C/EOV work area at DXXCTL. It then issues an IECRES macro instruction to synchronize NSL DCB processing. After DCBs have been synchronized, register 5 is set to point to the DCB parameter list before control is transferred to NSLOHDR1.</p> <p>The NSL routine is given the full parameter list for processing. It is required that the NSL routine save and restore registers.</p>	IFG0195D	OTA60100
<p>② If the NSL routine rejects the volume, the tape is unloaded. A mount message is constructed in the O/C/EOV work area to remount the volume. A rejection switch is set in the Where-to-go table (WTGPATH+1=X'02') to indicate a tape was rejected.</p>	IFG0195D	OTA60600

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>④ If the first record contains the first four characters "VOL1," the tape is unloaded and the message issued again. The tape is rewound if it is an acceptable NSL tape. If a channel error or a unit check other than a data check occurs, an exit is taken to problem determination.</p>	IFG0195D	OTA62100
<p>1 This function is also entered from the Final Common function (Figure 6) after reject processing has been performed.</p> <p>If this is the first entry, registers RUCB and RTIOT are saved in the O/C/EOV work area at DXXCTL. The IECRES macro instruction is issued to synchronize NSL DCBs. After DCBs have been synchronized, control is transferred to NSLOHDRO.</p>	IFG01950	OTA70100
<p>2 The mount bit in the UCB (SRTEDMCT=X'80') is tested to determine if the tape was accepted. If the mount bit is set to one, the tape was not accepted, the rejection switches in the Where-to-go table (WTGPATH+1=X'02') and the JFCB (JFCBMASK+5=X'08') are set. The data management count in the UCB (SRTEDMCT) is set to zero and an exit is taken to the Volume Mounting function.</p>	IFG01950	OTA70800

Figure 10: Open – SL Output Processing

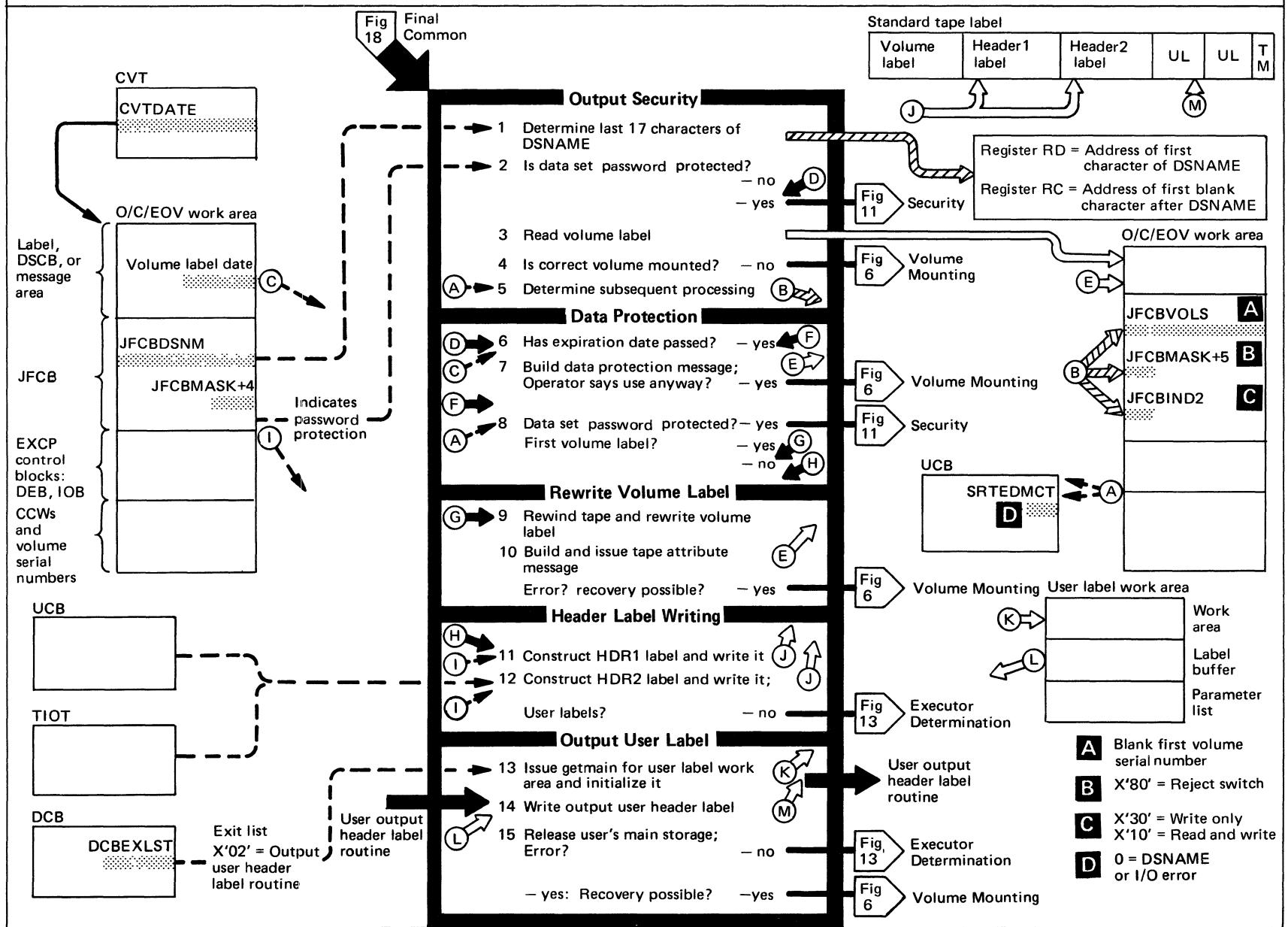


Figure 10. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 The security bit (JFCBMASK+4=X'04') is tested to determine whether the password was read, which indicates that security processing was previously performed. If so, this function is skipped.	IFG0196N	OTA38100
3 A read is issued for header label 1.	IFG0196N	OTA38500
If a tapemark is read, the security bit (see 2 above) is set to bypass the date protection function, which follows.	IFG0196N	OTA39800
5 If a DSNAME or I/O error occurs, the data management count in the UCB (SRTEDMCT) is set to zero, the volume serial number in DXVOLMT1 is set to SCRTCH, and the first volume serial number in the JFCB (JFCBVOLS) is blanked out. This sets the conditions for a scratch volume request. The rejection switch (JFCBMASK+5=X'80') is set and an exit is taken to the Volume Mounting function to reprocess this DCB.	IFG0196N	OTA38900

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
10 If an I/O error occurs while processing the first volume label on the first volume of a data set, the following steps are taken to recover from the error: <ul style="list-style-type: none"> • Set DXVOLMT1 in the O/C/EOV work area to SCRTCH • Zero data management count in the UCB (SRTEDMCT) • Blank the volume serial number in the JFCB (JFCBVOLS) An exit is then taken to the Volume Mounting function which will request that a scratch volume be mounted.	IFG0196T	OTA46000
13 If the DCB exit list does not exist, or if it does not contain a user header label exit, user label processing is bypassed. If the label exit routine exists, registers 5-12 are saved in the user label work area, and the user label count in the user label work area is set to zero.	IFG0196U	OTA64100
14 Unless American National Standard user labels are being processed, the user label count is increased by one and tested for the maximum number of 8 IBM standard user labels.	IFG0196U	OTA46600

Figure 11: Open Security

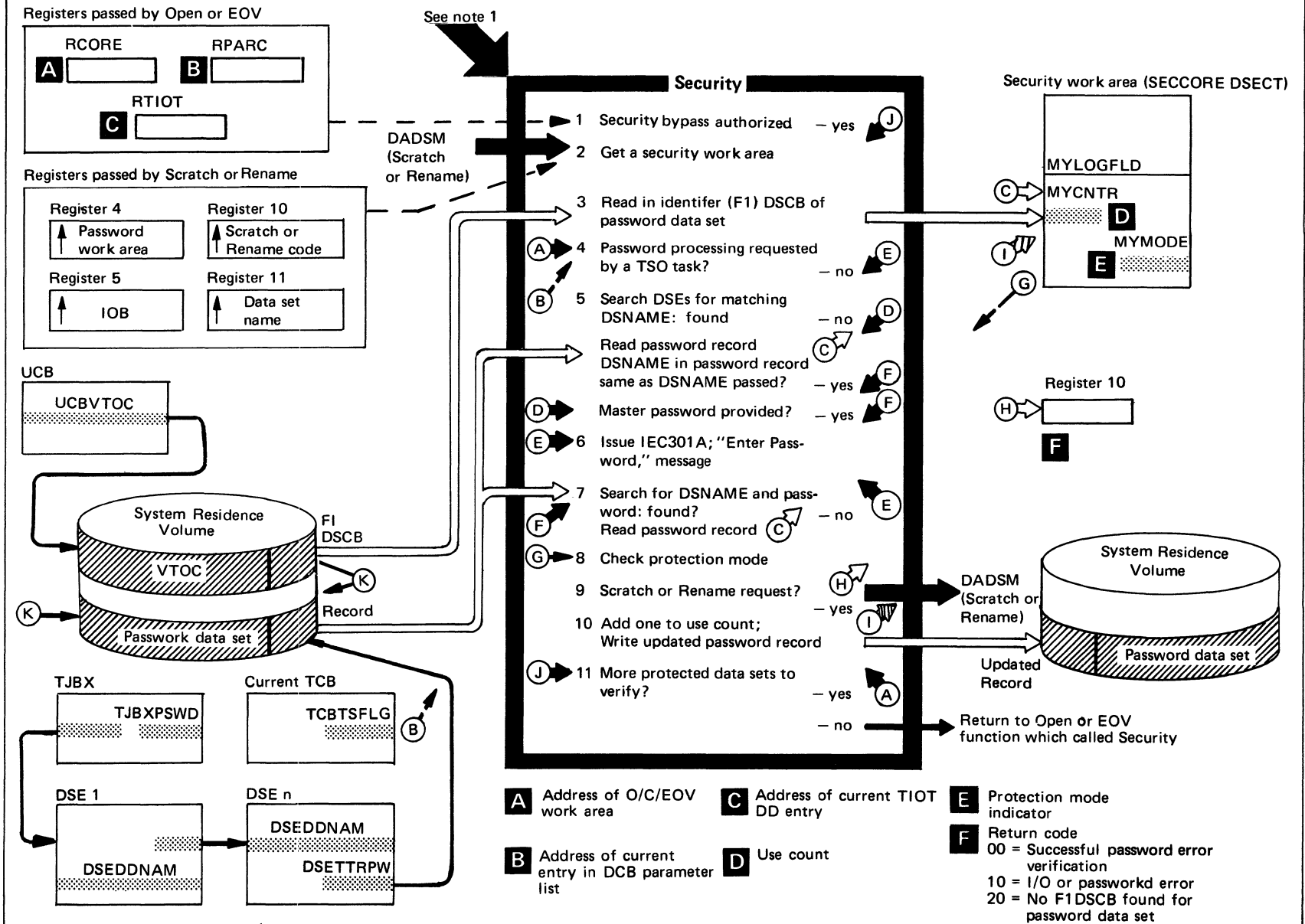
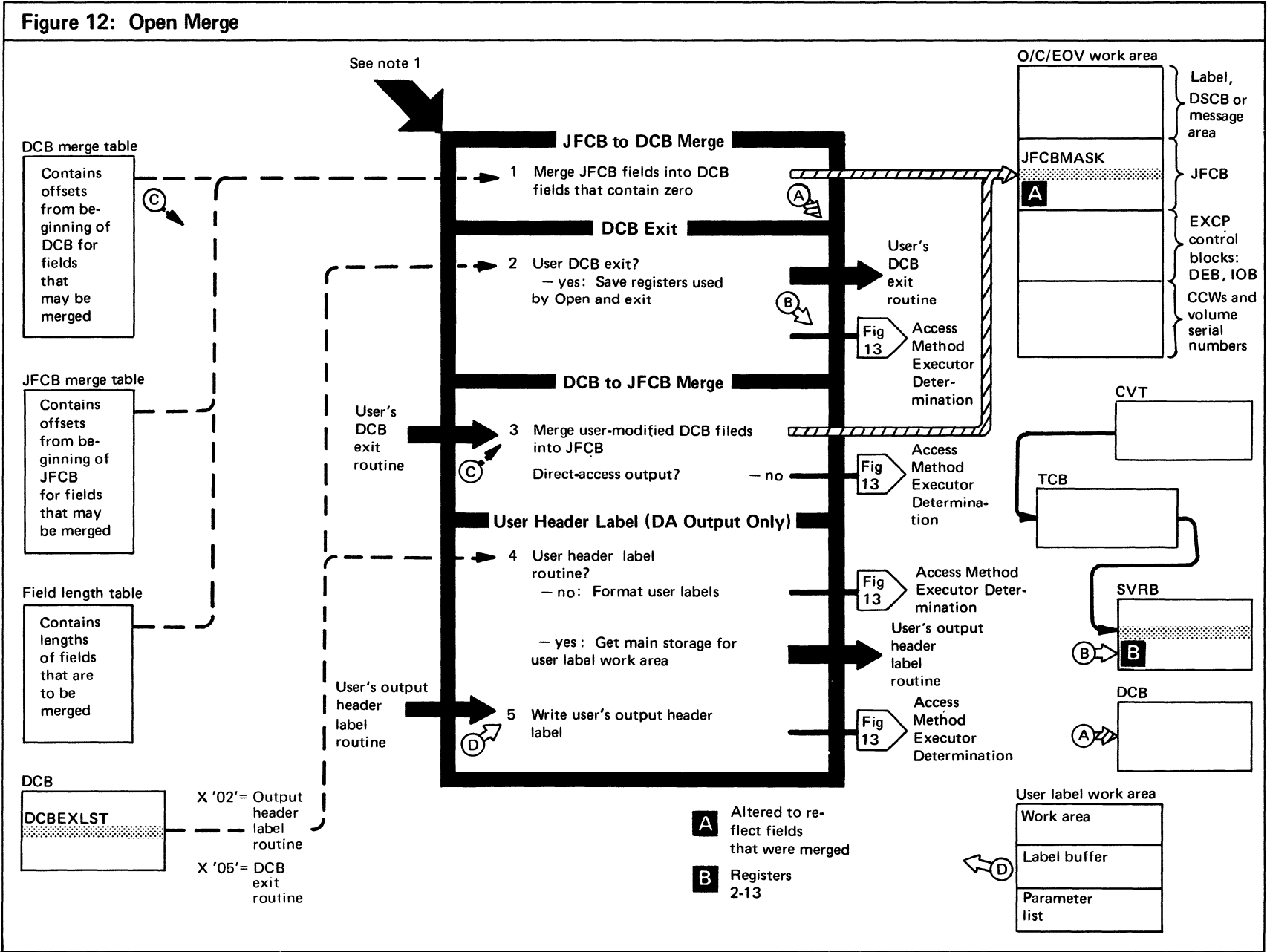


Figure 11. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 This function is entered from the following Open functions:</p> <ul style="list-style-type: none"> ISAM/BDAM Parallel Mounting (Figure 5) User Header/Trailer Labels (Figure 5) BPAM Concatenation (Figure 5) Final Common (Figure 6) SL Input DISP=MOD File Label 2 (Figure 8) SL Input User Header Label (Figure 8) <p>or the following EOVS functions:</p> <ul style="list-style-type: none"> Standard Header Label (Figure 32) Mount Ahead (Figure 36) <p>This function verifies the password of one or more password-protected data sets. A check is made to determine whether security processing can be bypassed for TSO tasks.</p>	IFG0195T	OSQ10400
<p>2 If processing Open or EOVS requests, a security work area is obtained.</p> <p>If a Scratch or Rename request is being processed, a security work area is obtained.</p>	IFG0195T SECLOADA	OSQ10600 OSQ50200
<p>3 If an Open or EOVS request is being processed, this function reads the address of the password data set into the security work area. If the identifier (format 1) DSCB cannot be located, a 213 ABEND is issued.</p> <p>If a Scratch or Rename request is being processed, this function reads the address of the password data set into the security work area. If the identifier (format 1) DSCB cannot be located, a code of X'20' is returned to Scratch or Rename in register 10.</p>	IFG0195T SECLOADA	OSQ11200 OSQ50200
<p>4 This function is entered if a task running under the time-sharing option (TSO) requires password-protection (security) processing. It may be entered as a result of an Open, EOVS, Scratch, or Rename request. Whether or not it is a TSO task is determined by examining the TCBTSFLG field of the current TCB. If a TSO task is requesting password protection processing, this section of code obtains the address of the first data set extension (DSE) block from the TJBX; the DSEs are searched for a matching DSNAME.</p>	IFG0195V	OSQ30400

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>5 The DSEs are searched for a DSNAME that matches the DSNAME of the password-protected data set being processed. (Register 1 contains the address of the DSNAME passed.)</p> <p>If a matching DSNAME is found in a DSE and a password record address (in TTR format) exists in the DSE, this function issues an ENQ macro for the DSNAME, reads the password record, and compares the DSNAME with the DSNAME passed in Register 1.</p> <p>If no matching DSNAME is found in the DSE chain, this function checks for the presence of a master password in the TJBXPSWD field of the TJBX.</p>	IFG0195V IFG0195V IFG0195V	OSQ30400 OSQ33000 OSQ33000
<p>6 This function issues a WTOR macro instruction to the operator console or 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.</p>	READPSWD	OSQ41000
<p>7 This function searches the password data set for the matching DSNAME and password. If the password record containing the matching DSNAME and password are found, the record is read into the security work area. This is performed in IFG0195U for Open or EOVS requests or in SECLOADA for Scratch or Rename requests.</p>	IFG0195U SECLOADA	IFG0195U (Open or EOVS) OSQ52000 (Scratch or Rename)
<p>8 If the search is successful for an Open or EOVS request, the protection-mode indicator in the password record (which can indicate a read-only data set or a data set that can be opened for both reading and writing) 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 incorrect (for example, the user is opening the data set for output, but the protection-mode indicator allows read-only), a 913 ABEND is issued. If the search is successful for a Scratch or Rename request, control is returned to Scratch or Rename via XCTL.</p>	IFG0195U IFG0195V	OSQ23000 (non-TSO task) OSQ31400 (TSO task)
<p>10 If the use mode (input or output) is compatible with the protection mode (read-only, read-write), the use count in the password record is increased by one and the updated record is written to the password data set.</p>		
<p>11 If all of the data sets requiring password processing have been verified, control is passed to the Open or EOVS calling function (see list of Open and EOVS calling functions in Step 1).</p>		



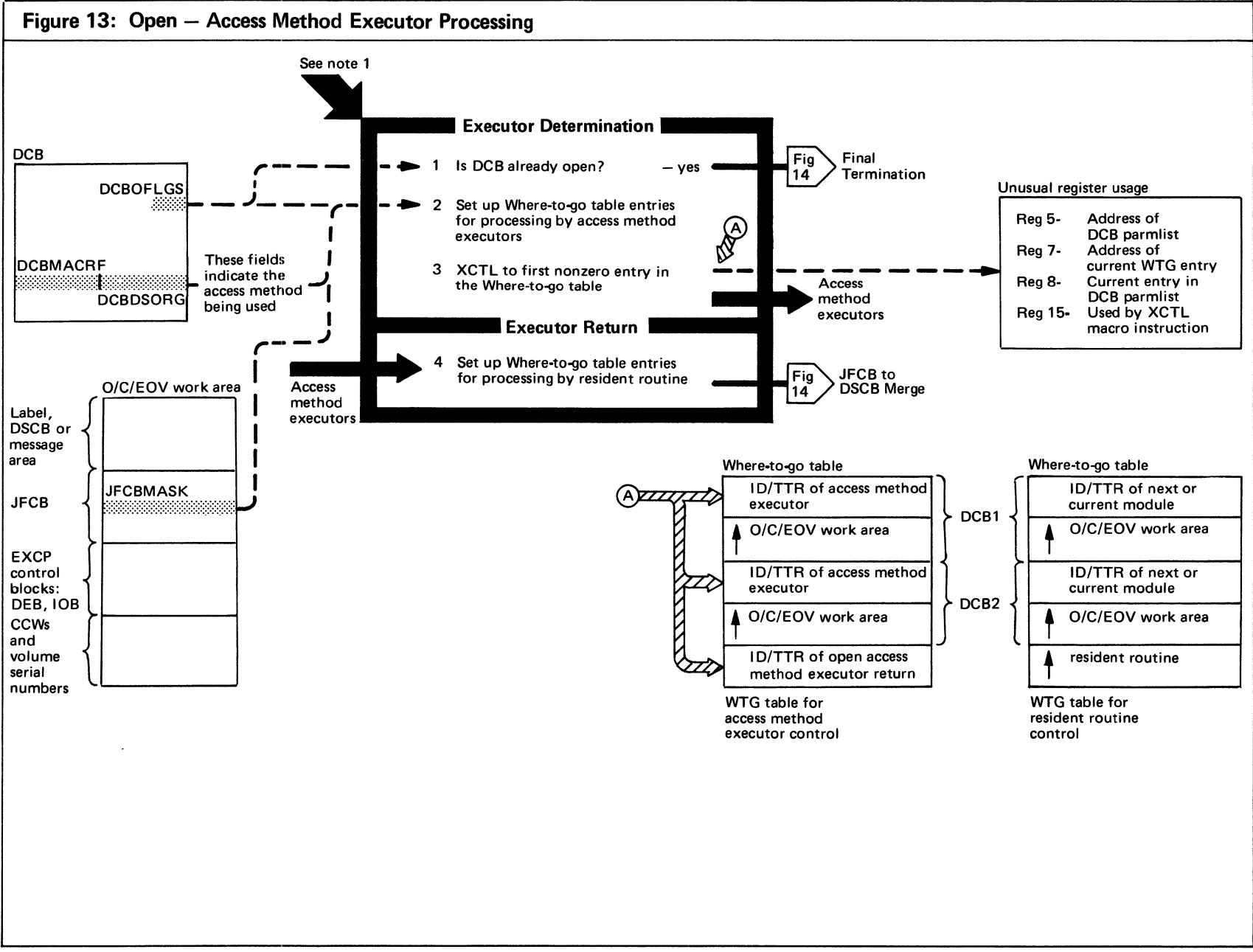


Figure 13. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 This function is entered from the following functions:</p> <p>Tape-SL Output Header Label Writing (Figure 10) Tape-SL Output User Labels (Figure 10) JFCB to DCB Merge (Figure 12) User Output Header Labels (Figure 12)</p>	IFG0196V	IFG0196V
<p>2 The Where-to-go table is used throughout Open/Close/EOV processing to identify the next or current module needed to process each DCB. After Open sets it up for the access method executors and transfers control, the access methods use the Where-to-go table and pass it back to Open when they are through.</p>	IFG0196V	OMG91400

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL																																																
<p>5 Continued</p> <table border="0"> <thead> <tr> <th><u>Access Method</u></th> <th><u>DSORG</u></th> <th><u>Executor</u></th> </tr> </thead> <tbody> <tr> <td>EXCP</td> <td>any</td> <td>IGG0191A</td> </tr> <tr> <td>BSAM, QSAM</td> <td>PS</td> <td>IGG0191A</td> </tr> <tr> <td>TSO Terminal Dummy Data Set</td> <td>PS</td> <td>IGG0191A</td> </tr> <tr> <td>BPAM</td> <td>PO</td> <td>IGG0191A</td> </tr> <tr> <td>BSAM, QSAM Dummy Data Set</td> <td>PS</td> <td>IGG0191C</td> </tr> <tr> <td>ISAM</td> <td>IS</td> <td>IGG0192A</td> </tr> <tr> <td>BDAM</td> <td>DA</td> <td>IGG0193A</td> </tr> <tr> <td>READ/WRITE</td> <td>CX</td> <td>IGG0193M</td> </tr> <tr> <td>GET/PUT</td> <td>CX</td> <td>IGG0193N</td> </tr> <tr> <td>QTAM</td> <td>CQ</td> <td>IGG0193O</td> </tr> <tr> <td>QTAM</td> <td>MQ</td> <td>IGG0193P</td> </tr> <tr> <td>GAM</td> <td>GR</td> <td>IGG0193Y</td> </tr> <tr> <td>TCAM Message Queue</td> <td>TQ</td> <td>IGG0193O</td> </tr> <tr> <td>TCAM Line Group</td> <td>TX</td> <td>IGG01935</td> </tr> <tr> <td>TCAM Dummy Data Set</td> <td>PS</td> <td>IGG01946</td> </tr> </tbody> </table>	<u>Access Method</u>	<u>DSORG</u>	<u>Executor</u>	EXCP	any	IGG0191A	BSAM, QSAM	PS	IGG0191A	TSO Terminal Dummy Data Set	PS	IGG0191A	BPAM	PO	IGG0191A	BSAM, QSAM Dummy Data Set	PS	IGG0191C	ISAM	IS	IGG0192A	BDAM	DA	IGG0193A	READ/WRITE	CX	IGG0193M	GET/PUT	CX	IGG0193N	QTAM	CQ	IGG0193O	QTAM	MQ	IGG0193P	GAM	GR	IGG0193Y	TCAM Message Queue	TQ	IGG0193O	TCAM Line Group	TX	IGG01935	TCAM Dummy Data Set	PS	IGG01946		
<u>Access Method</u>	<u>DSORG</u>	<u>Executor</u>																																																
EXCP	any	IGG0191A																																																
BSAM, QSAM	PS	IGG0191A																																																
TSO Terminal Dummy Data Set	PS	IGG0191A																																																
BPAM	PO	IGG0191A																																																
BSAM, QSAM Dummy Data Set	PS	IGG0191C																																																
ISAM	IS	IGG0192A																																																
BDAM	DA	IGG0193A																																																
READ/WRITE	CX	IGG0193M																																																
GET/PUT	CX	IGG0193N																																																
QTAM	CQ	IGG0193O																																																
QTAM	MQ	IGG0193P																																																
GAM	GR	IGG0193Y																																																
TCAM Message Queue	TQ	IGG0193O																																																
TCAM Line Group	TX	IGG01935																																																
TCAM Dummy Data Set	PS	IGG01946																																																

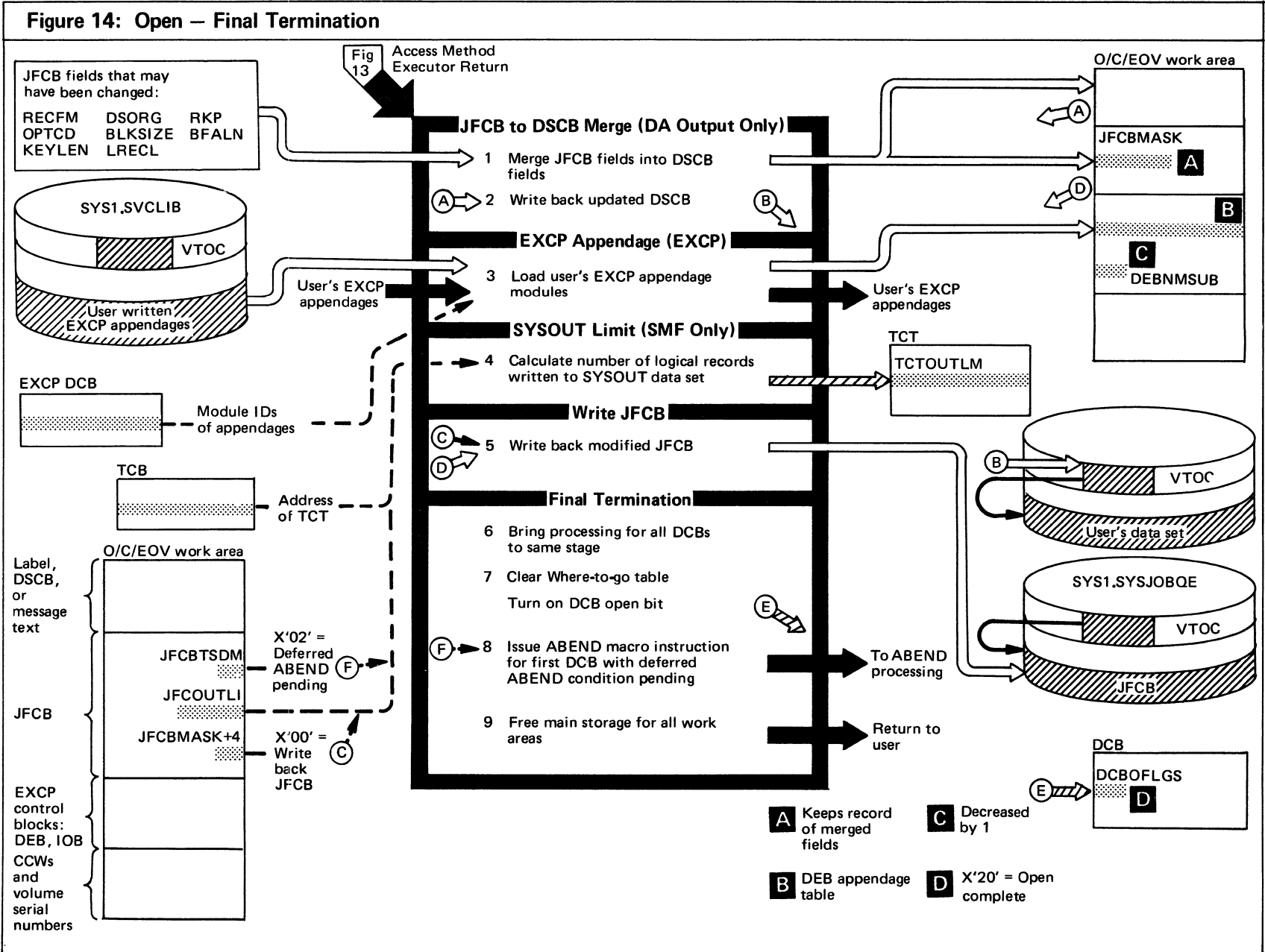


Figure 14. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 This merge is only done for output processing. The corresponding second step of the backward merge for tape is the writing of output standard labels, which occurs before control is given to the access method executors.	IFG0196W	OFN20200
2 If the write-back indicator is on, the updated DSCB is written back to its location in its VTOC. (The CCHHR was saved in DSCCW7 in the O/C/EOV work area when the VTOC was first searched.)	IFG0196X	OFN32200
3 If the EXCP access method is specified, bit 2 of the DCBMACRF field is set. The module IDs of EXCP appendages are put in the EXCP access method interface section of the EXCP DCB. Open supplies IGG019 as the leading six characters of the module names. The user must have link-edited his modules into SYS1.SVCLIB or be using standard appendages that already exist in SYS1.SVCLIB.	IFG0196X	OFN41200
The order of the appendages in the DCB does not correspond to the order in the DEB appendage vector table. The PCI and SIO appendage IDs are swapped in the DCB to correspond to the DEB during loading.	IFG0196X	OFN41200
After the EXCP appendages have been processed, the PCI and SIO module IDs are put in their original locations.	IFG0196X	OFN41400

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
4 Since records may be blocked, the number of logical records is obtained by counting the number of EXCPs. Short blocks, variable length record headers, and short variable length logical records can cause this EXCP count to be inexact.	IFG0196X	OFN30400
The number of logical records (EXCPs) is calculated by multiplying the OUTLI value in the JFCB by the contents of the DCBLRECL field, dividing by the DCB BLKSIZE field, and adding one if there is a remainder. This calculation does not consider the difference in the relationship between LRECL and BLKSIZE for F, V, and U type records.	IFG0196X	OFN30600
5 The JFCB is written back unless the write-back inhibit bit is set (JFCBTSDM=X'08'). This bit allows the Nucleus Initialization Program (NIP), which is loaded during IPL, to open system data sets, including SYS1.SYSJOBQE, before SYS1.SYSJOBQE contains valid JFCBs.	IFG0196X	OFN51200
7 If the DCBOFLGS lock and busy bits are still on, the DCB has just been successfully opened. The open bit is turned on and the busy bit off.	IFG0198N	OFN73600

Figure 15: Close Contents

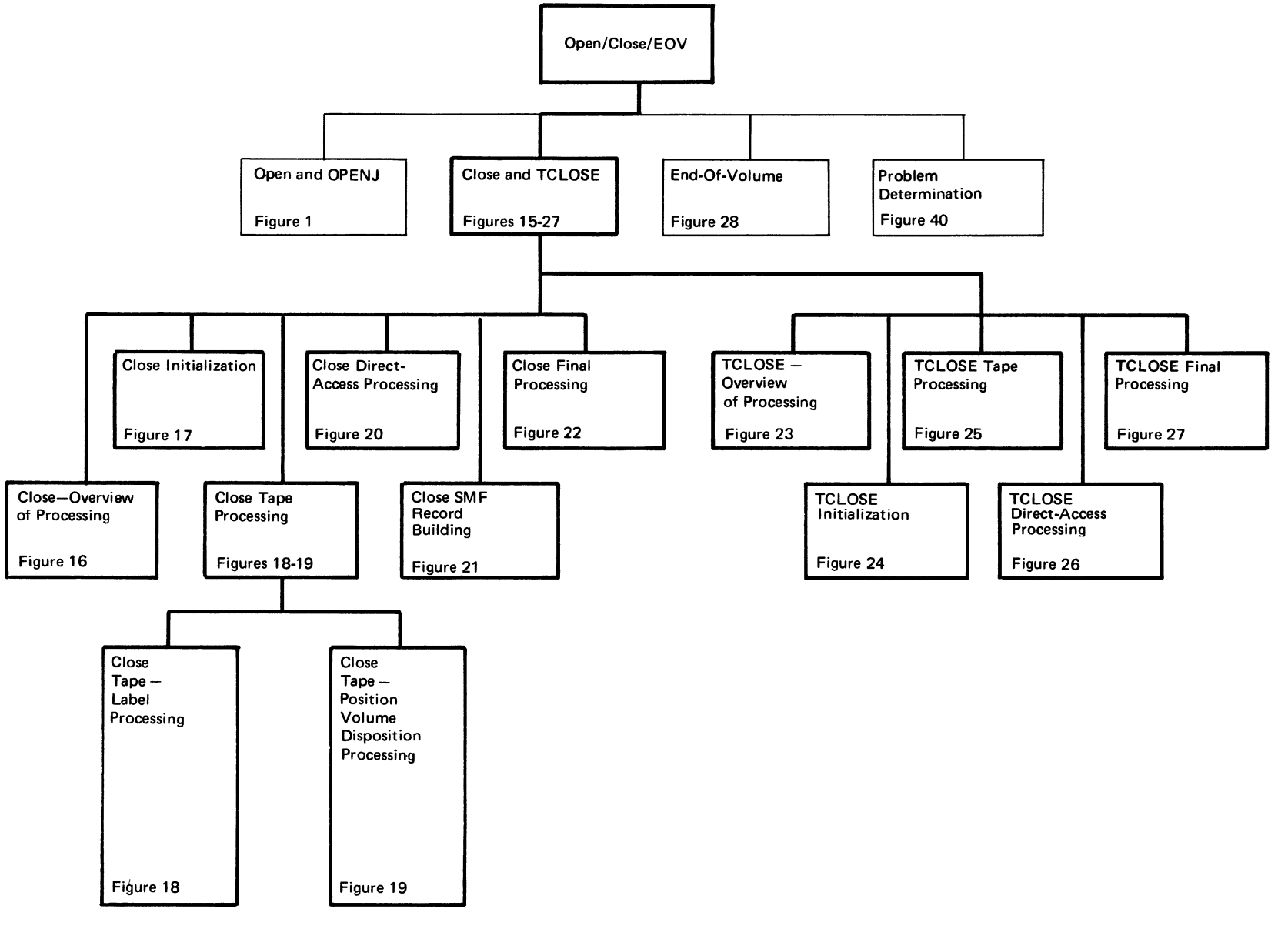
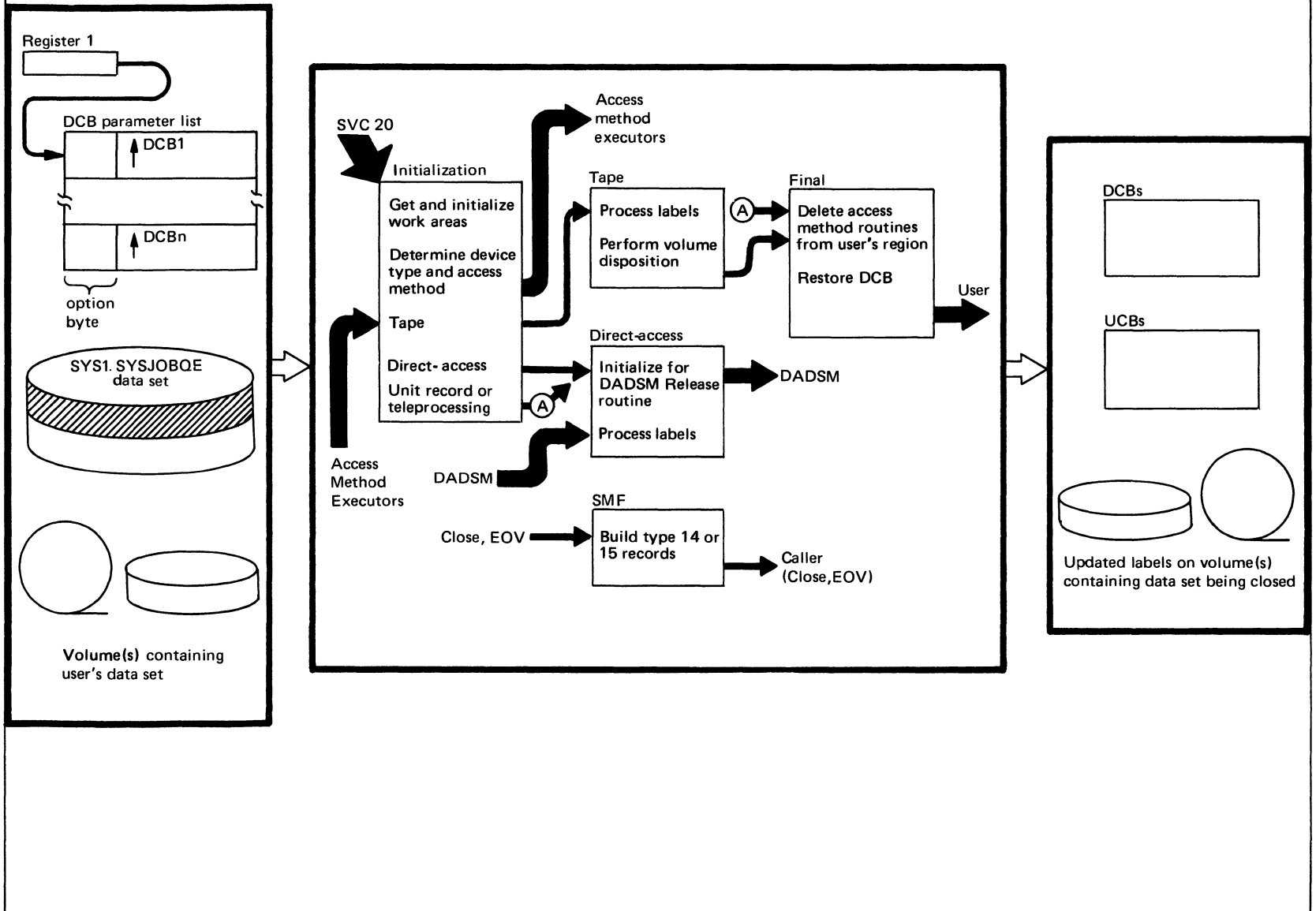


Figure 16: Close – Overview of Processing



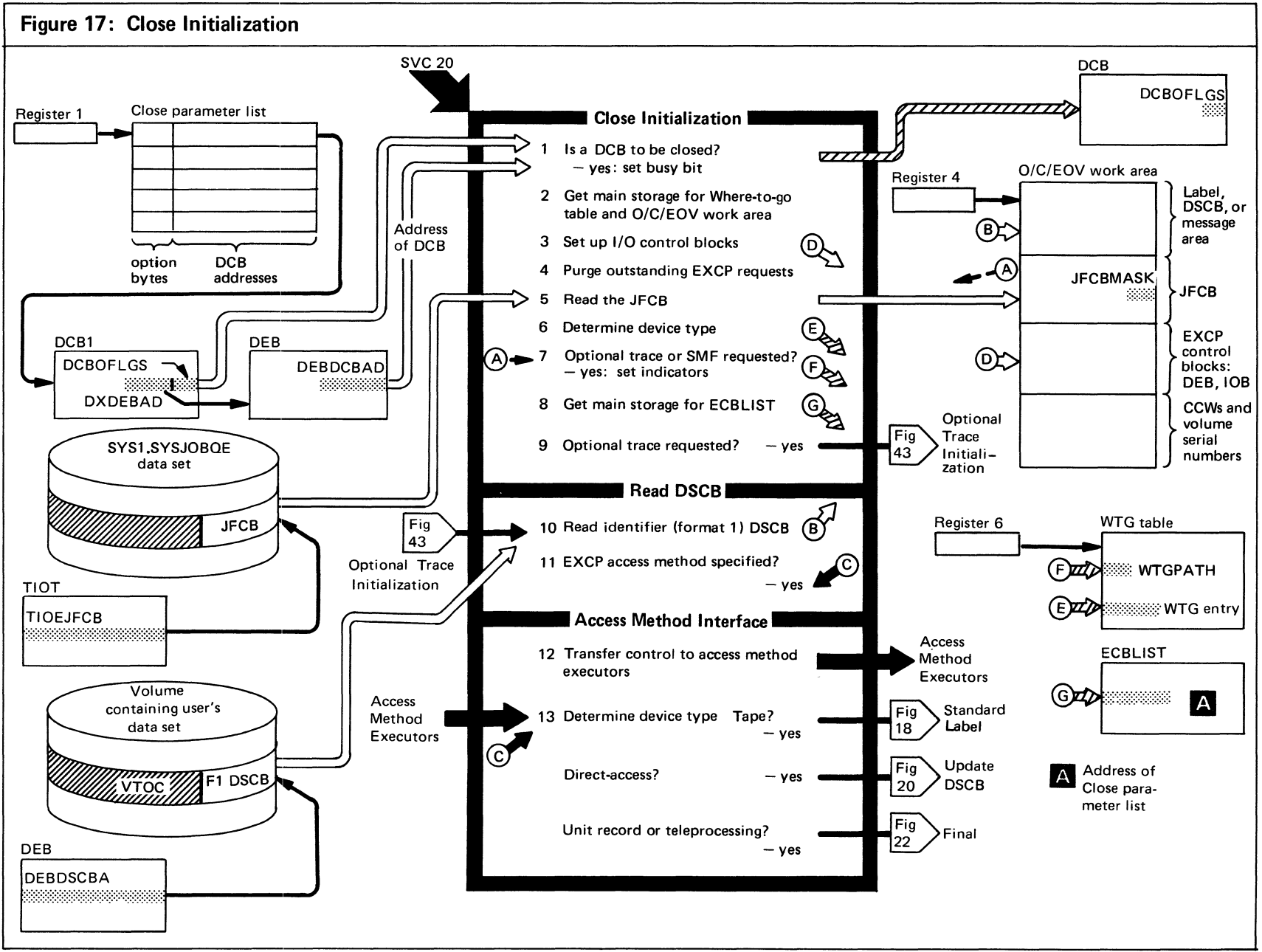


Figure 17. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 This function checks the open bit and lock bit of the DCBOFLGS field; both must be set to 1 (on) to indicate that the DCB is open and that user-exit routines are no longer in control. If these bits are both on, it verifies that the DEB on the current TCB DEB chain points to the right DCB and sets busy bit (bit 7) of the DCBOFLGS field to 1 to indicate that this DCB is currently being processed by Open, Close, or End-of-Volume.	IGC00020	CIN00100
2 The address of the resident routine is put in the WTG table (which consists of 40 bytes plus 8 bytes for each DCB to be processed) after the table has been set to zero. An O/C/EOV work area is acquired for each DCB to be closed. The O/C/EOV work area requires 488 bytes unless trailer label processing has been deferred from end-of-volume to end-of-data at the user's request, in which case 536 bytes are needed.	IGC00020	CIN00500
3 The control blocks built by this routine for EXCP I/O operations include a valid DEB, but the only part of the DCB included is a DEB pointer.	IGC00020	CIN00850
4 EXCP and ABEND I/O requests against this DCB are purged. This is normally done by the access method executors.	IGC00020	CIN00900
5 The offset of the DD entry for this data set in the TIOT for the DCB is obtained, the relative address of the JFCB is converted to an absolute track address using the resident conversion routine (IECPONVT), and the JFCB is read into the O/C/EOV work area. For rotational position sensing (RPS) devices, a channel program is built. This routine then links to the resident sector conversion routine (IECPSCRI) to convert the sector value. No JFCB is read into the O/C/EOV work area for system tasks.	IFG0200V	CIN02200

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
6 The device type is determined, and the module IDs of Close tape or direct-access routines are put in each active DCB entry of the Where-to-go table.	IFG0200V	CIN01000
7 The JFCBMASK field is tested to determine whether an optional trace or SMF information was requested by the user.	IFG0200V	CIN02500
8 Main storage for the resident routine work area is acquired. This work area is an ECBLIST, which is 4 (n+1) bytes long where n is the number of DCBs being closed. If an optional trace has been requested by the user, the resident routine work area is two words longer, and the address of the close parameter list is saved in the first word of the work area.	IFG0200V	CIN0300
10 If the data set being closed resides on a direct-access device and was open for output, or user labels have been specified, this routine reads the data portion of the identifier (format 1) DSCB into the O/C/EOV work area.	IFG0200V	CIN03800
12 Transfers control to the close access method executors using the XCTL macro instruction.	IFG0200W	CIN05800
13 When control is returned from the access-method executors, the device type is determined. Subsequent transferring of control is handled by the resident routine.		

Figure 18: Close Tape – Label Handling

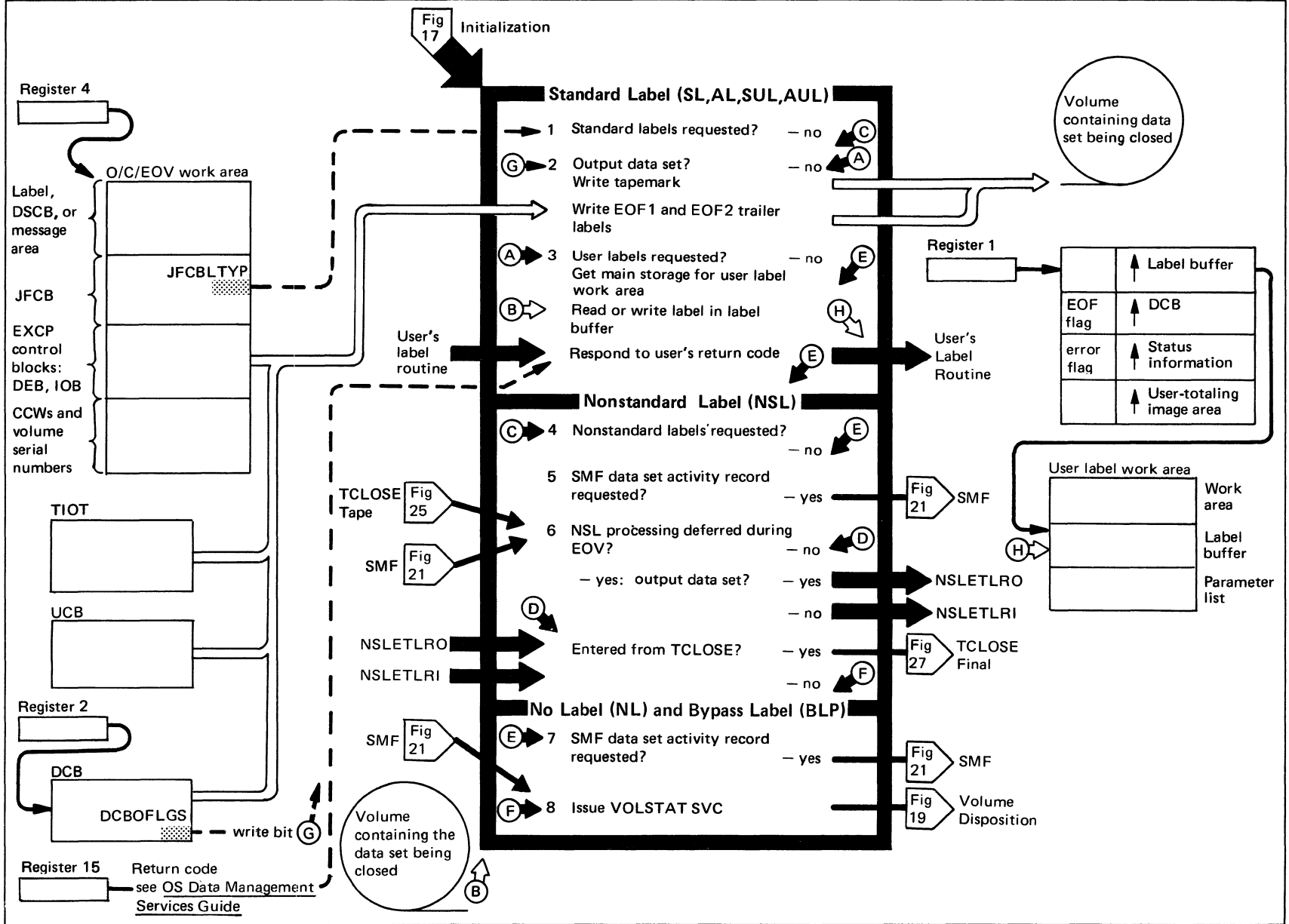


Figure 18. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 Checks the write bit of the DCBOFLGS field to determine whether the last operation was a write (indicates an output data set is being closed). Writes a tapemark and standard trailer 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 Tape Labels.	IFG0200Z IFG0200Z	CIN06600 CTA00200
3 Processes standard user labels (SUL or AUL) if JFCBLTYP field equals X'08'. If an input data set is being closed, a user label is read into the label buffer. If an output data set is being closed, a cleared buffer is passed to the user's program. Control is passed to the user's label routine via a SYNCH macro instruction (SVC 12). When control is returned from the user's label routine, the user's return code in register 15 is examined to determine whether label processing should continue.	IFG0202A	CTA03600
4 Checks to see if nonstandard labels (NSL) are requested.	IFG0202B	CCM00200

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
5 If a Systems Management Facility (SMF)=type 14 or 15 record is required (data set activity is being monitored by SMF), this routine transfers control to the SMF routines. This must be done before control is passed to the user's NSL routine because these routines alter the volume's disposition (which may change the information that is to be gathered for the SMF records).		
6 This routine is entered from Close and TCLOSE tape routines for processing nonstandard labels. If NSL processing was deferred from end-of-volume to end-of-data, control is transferred to NSLETRLI so the user can process nonstandard labels. When control is returned, the extra 48 bytes of the O/EOV work area needed for deferred label processing are freed. Transfers control to NSLCTRLO so the user can process nonstandard, output labels and perform volume disposition. Checks for entry from TCLOSE and returns control appropriately.	IFG0202B IFG0202B	CCM00500 CCM00700
8 Error statistics by volume (ESV) are written either to the operator's console or the SMF data set. See <u>OS I/O Supervisor Logic</u> for details on VOLSTAT (SVC 91).	IFG0202B IFG0202F	

Figure 19: Close Tape – Volume Disposition Processing

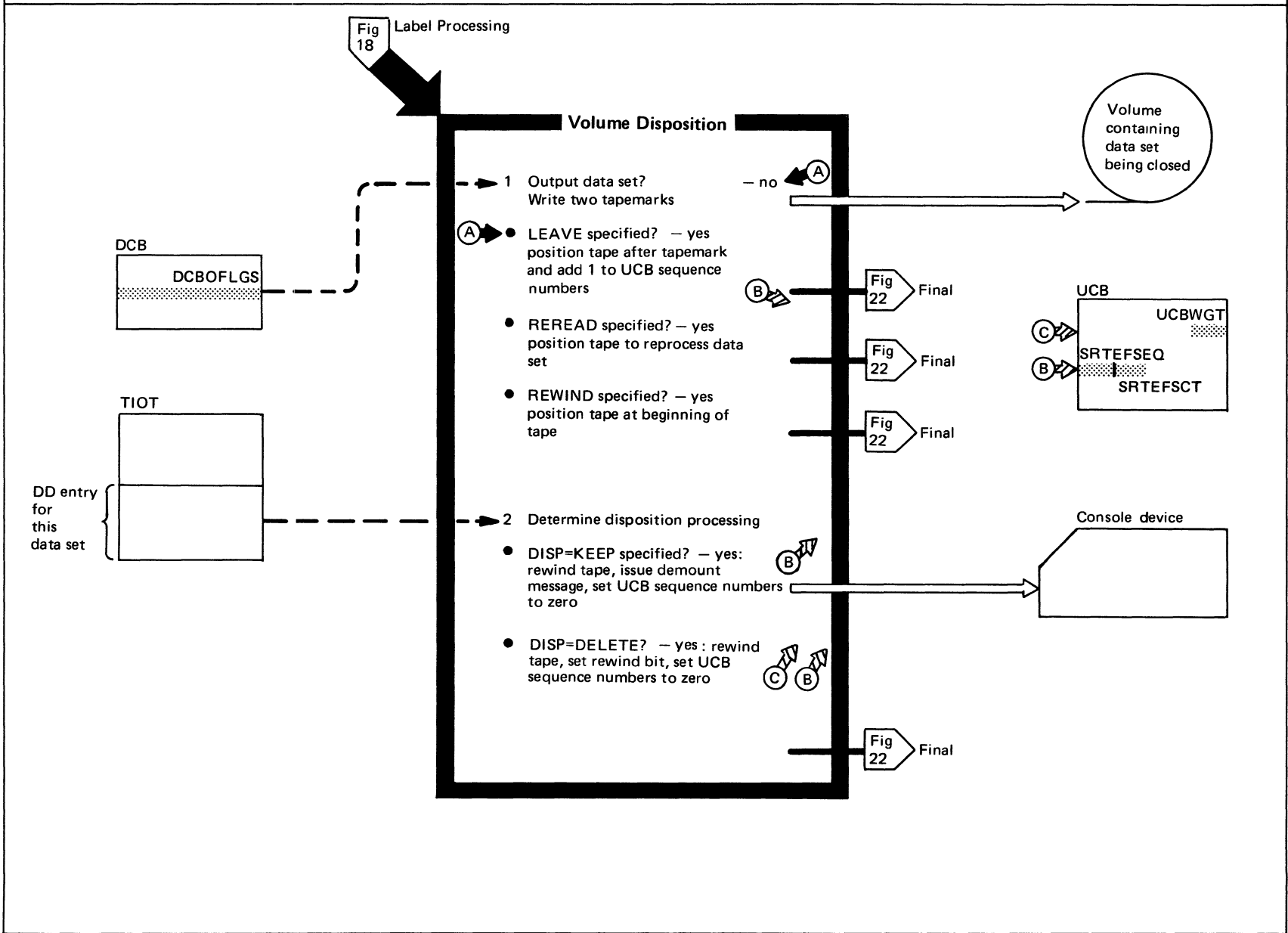


Figure 19. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 This function positions tape volumes according to the close option specified by the user and updates the UCB accordingly. For SYSIN, SYSOUT, and null data sets, no positioning is necessary. For unlabeled tape, tapemarks 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 and one forward space file are issued for labeled tape or two backspace files and one forward space file are issued for unlabeled tape.</p>	IFG0202F	CTA06850

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>2 If no options (LEAVE, REREAD, REWIND, 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 DISP action. If KEEP is requested for a reserved data set, a REWIND is issued and the rewind bit (bit 3 in the UCBWGT field) is set to indicate to the job scheduler termination routine that no rewind is necessary. The logical (SRTEFSEQ) and physical (SRTEFSC) data set sequence numbers in the UCB are cleared. No demount message is issued for reserved volumes.</p>	IFG0202F	CTA10800

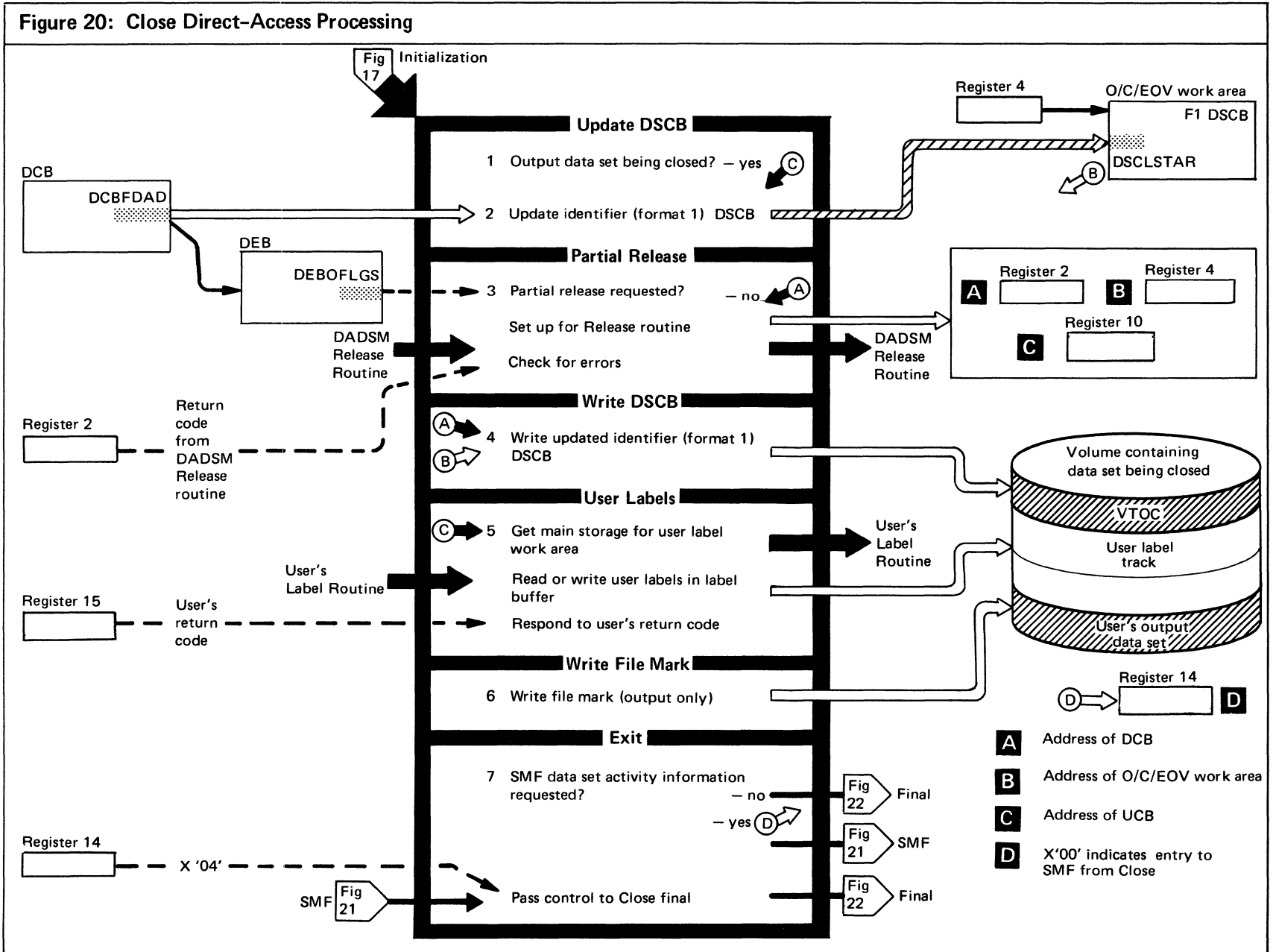


Figure 20. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 This function is entered to close output data sets on a direct-access device. If the identifier (format 1) DSCB was not modified, it is read in again. The absolute address in the DCBFDAD field is converted and written to the identifier (format 1) DSCB. The track balance field (DSCBSIND) and last volume indicator are updated. For partitioned data sets, the directory count (DSCBCDBL) is also updated.	IFG0200Y	CIN07860
3 If unused space is to be released before the data set is closed (DEBOFLGS field set to X'10'), control is transferred to the DADSM release routine (see OS DADSM Logic). Bit 0 of the WTGPATH field in the Where-to-go table is set to 1 to indicate partial release requested. When control is returned from the release routine, a check is made for errors encountered during release processing; on error conditions, this function sets internal ABEND code to 87, dequeues the VTOC, and transfers control to problem determination.	IFG0200Y	CIN07900
4 Writes updated identifier (format 1) DSCB to the VTOC; if partial release is in effect, the VTOC is dequeued.	IFG0201R	CDA00200

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
5 This function 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 DCB exit list. If updating is necessary, this routine writes the user's updated labels when control is returned from the user's label routine.	IFG0202D	CDA02500
This function is also entered if user labels are required for data sets opened for output. If the current volume of the data set is not the first volume this function must find the first volume and read in the identifier (format 1) DSCB of that volume. If the first volume is not mounted, no user label is written. Control is passed to the user's label routine to format the labels. When control is returned, the labels are written by this routine.	IFG0202C	CDA00800
6 This function is entered to write a file mark or to terminate a physical sequential, output data set. If no space remains in the data set, no file mark is written.	IFG0202E	CDA06560
7 If SMF information is required, this routine passes control to the SMF routines. If not, control is passed to the Close Final function by calling the resident routine.	IFG0202E	CDA08000

Figure 21: Close – SMF Record Building

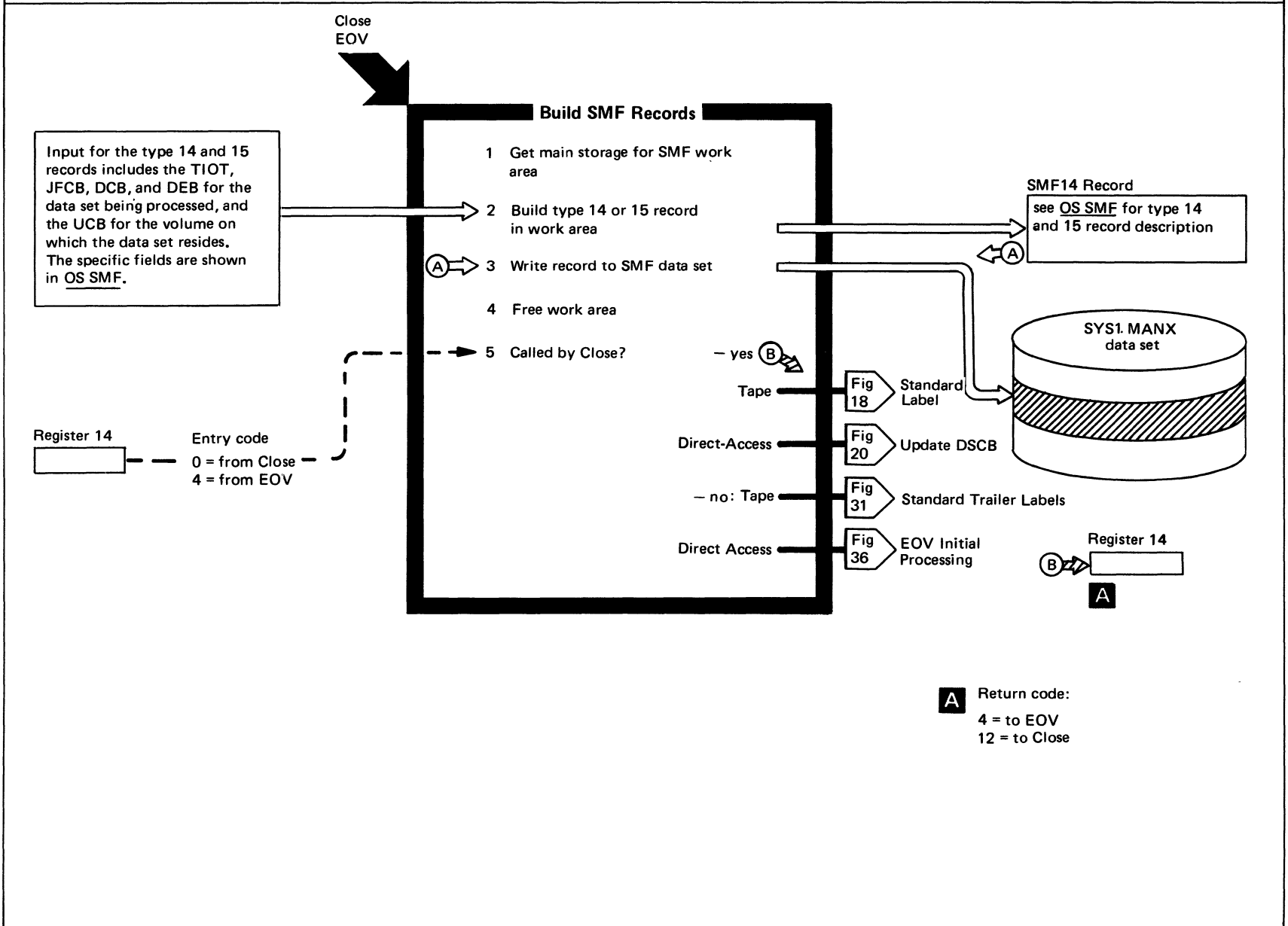


Figure 21. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>2 Builds JFCB, DEB, DCB segments of SMF record. Checks are made for device type and an appropriate UCB segment is built for the data sets residing on tape and DA devices.</p> <p>Determines the type of SMF record to be built. If the Close option in DCBOFLGS is INPUT or RDBACK, a type 14 SMF record is to be built. If the Close option in DCBOFLGS is OUTPUT, UPDATE, INOUT, or OUTIN, a type 15 record is to be built.</p>	<p>IFG0202H</p> <p>IFG0202I</p>	<p>CCM02520</p> <p>CCM05300 (DA)</p>

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>3 Issues SMFWTM (SVC 83) to write the record to the SYS1.MANX data set</p>	<p>IFG0202H</p>	<p>CCM04800 (Tape)</p>

Figure 22: Close Final Processing

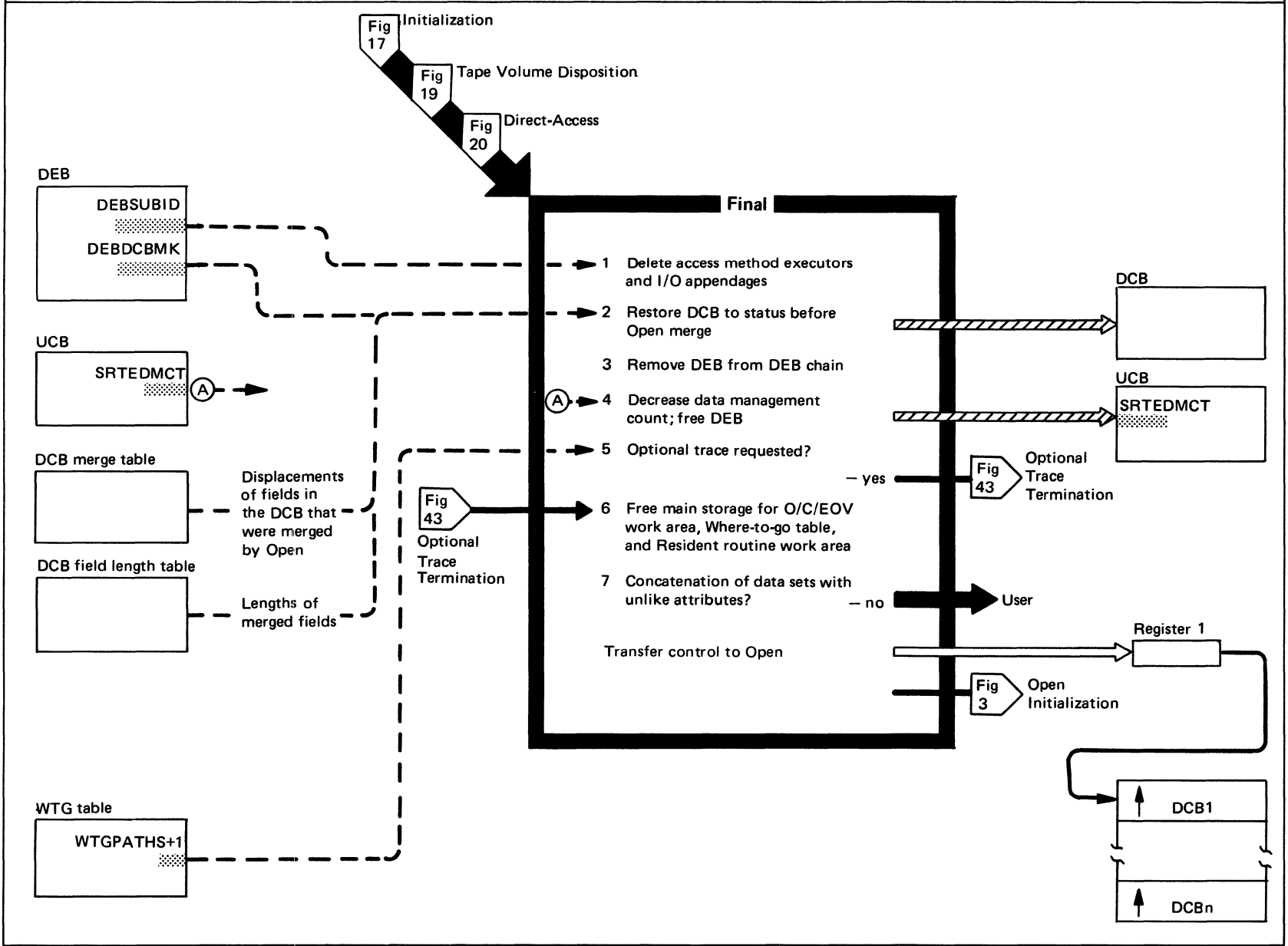


Figure 22. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 If concatenation of data sets with unlike attributes is in effect (DCBOFLGS=X'28'), the input synchronizing and error processing routine (IGG019AQ) or the check routine (GG019BB), whichever is being used, is not deleted. If user totaling is in effect, the user-totaling save area is freed.	IFG0202J	CFN00500
2 Restores fields of DCB to what they were before Open merge took place. The OPTCD, BFTEK, BFALN (HIARC bits 0 and 5 are not restored), DEVT, and DVTBL fields for direct-access DCBs are restored. For nonconcatenated data sets, the MACRF, IFLGS, and DDNAM fields 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.	IFG0202J	CFN04100

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 Removes the DEB for this data set from the TCB DEB chain.	IFG0202K	
4 The data management count (SRTEDMCT field) is the count of open DCBs. If the count is 0, it is not decreased.	IFG0202K	CFN01200
6 Releases the O/C/EOV work area and resident routine work area (ECBLIST). If there is an IRB address in the DEB, it is also released. Releases the Where-to-go table.	IFG0202L	CFN07800

Figure 23: TCLOSE – Overview of Processing

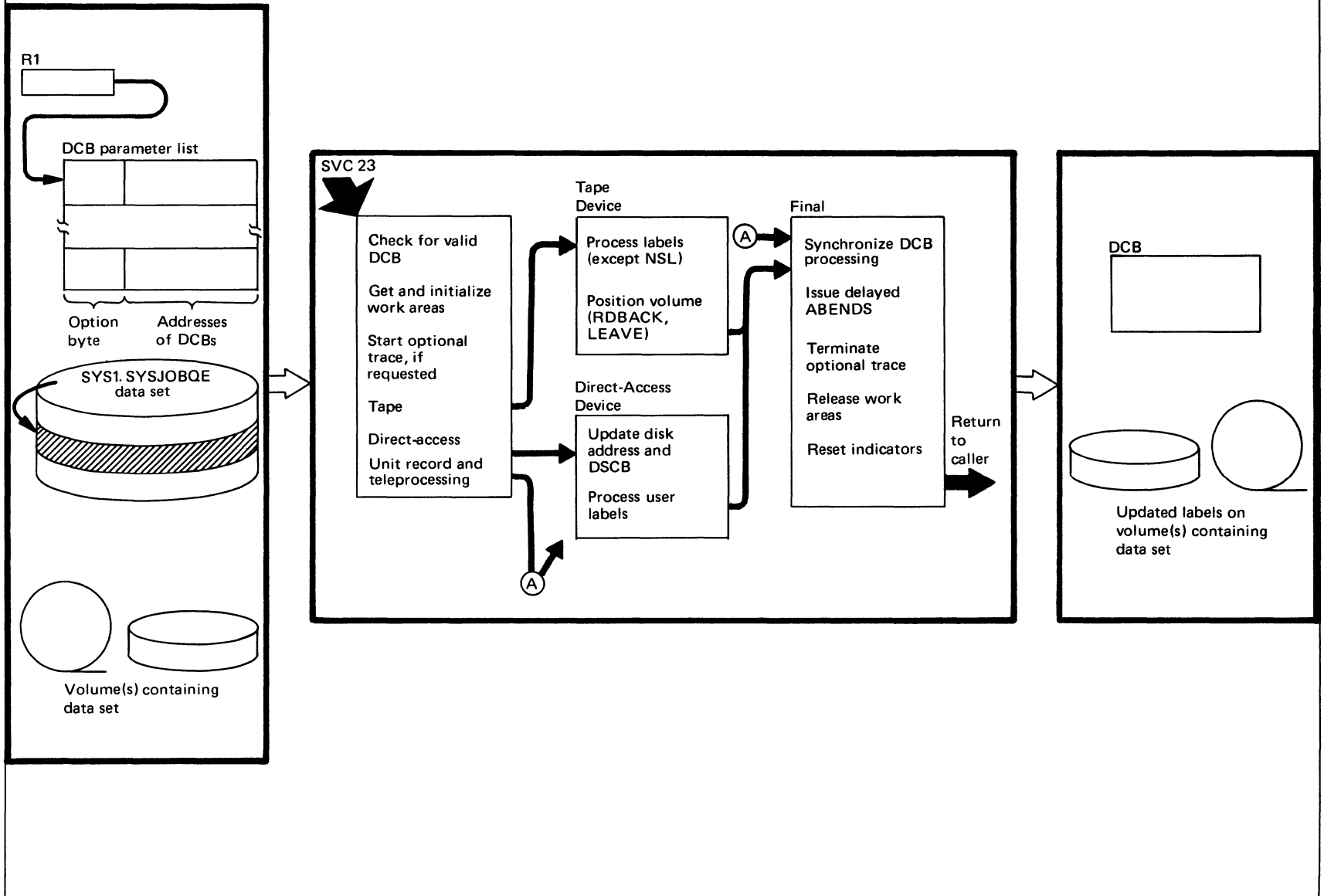


Figure 24: TCLOSE Initialization

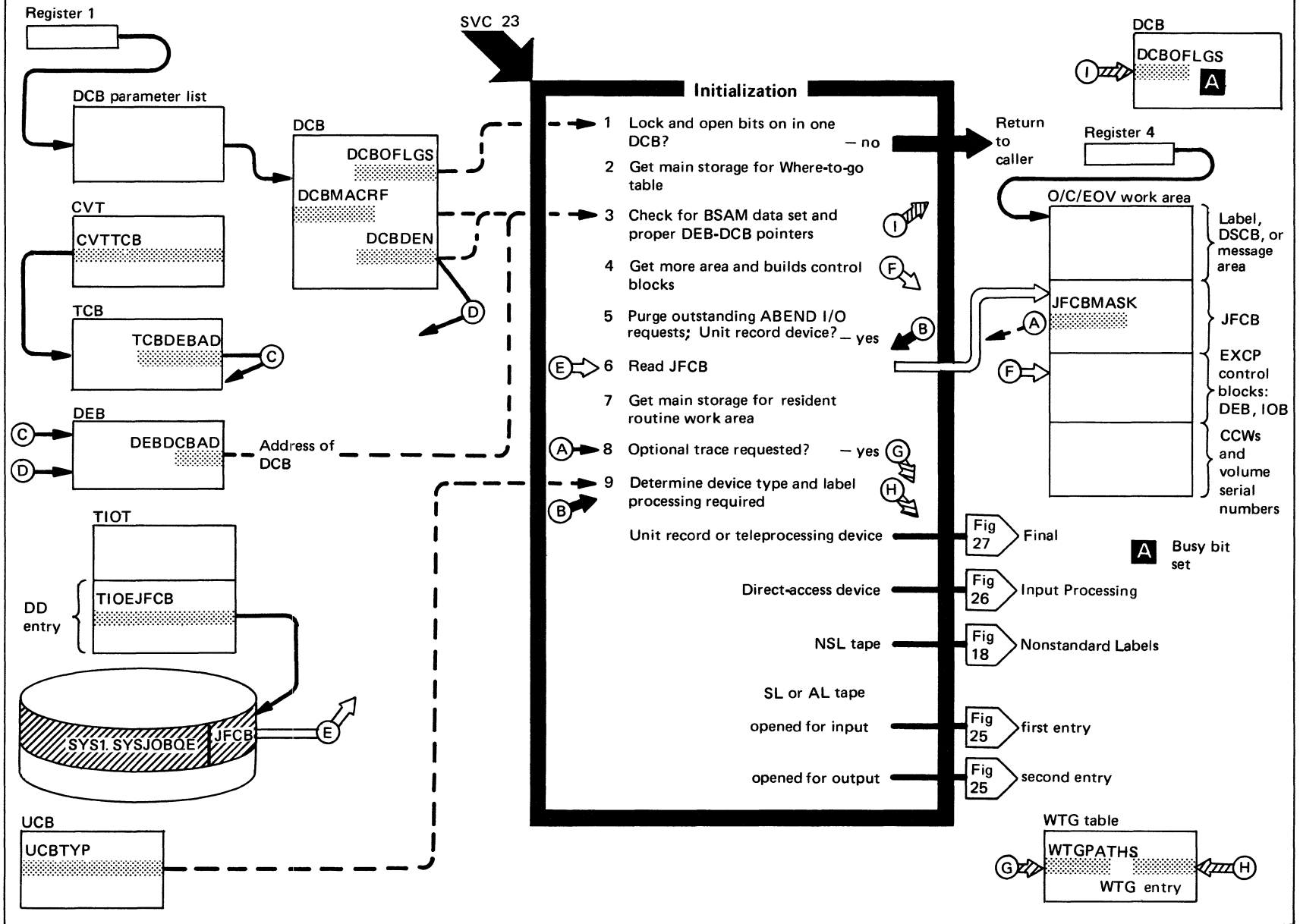


Figure 24. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 Checks the lock and open bits of DCBOFLGS; if both are on (set to 1), it is assumed that the DCB is to be temporarily closed and processing is continued. If no DCB in the parameter list has both bits on, control is immediately returned to the caller.	IGC0002C	TCM03000
2 If one or more of the DCBs represented in the parameter list passes the step 1 check, this step gets main storage for the WTG table, which is used for all DCBs being processed.	IGC0002C	TCM05000
3 To perform TCLOSE processing, the lock and open must be on in DCBOFLGS, the busy indicator in DCBOFLGS must be off, the access method must be BSAM (as indicated in the DCBMACRF and DCBMACRF + 1 fields), and the DEB pointed to by DCBDFBAD must be on the TCB DEB chain. This DEB must also contain the address of the DCB. If all of the validity tests are successful, the busy bit (in-process indicator) is turned on in the DCBOFLGS field, unless the device type is a terminal (TSO task). For TSO terminals, TCLOSE turns off the EOF indicator in the DCVIND1 field.	IGC0002C	TCM07000
4 This section of code gets a 488-byte O/C/EOV work area for each DCB to be processed by TCLOSE, unless deferred NSL processing is requested, for which a 536 byte work area is required for EOV processing.	IGC0002C	TCM16000

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
5 Issues a PURGE macro instruction specifying the QUIESCE I/O option. (See OS Data Management for System Programmers.)	IGC0002C	TCM19000
6 Reads the JFCB and determines whether the trace option is in effect (indicated by the JFCBMASK field); if so, it sets the trace bit in the WTGPATHS field of the WTG table	IGC0002C	TCM23000
7 Gets main storage for a resident routine work area (ECBLIST). This work area is (n + 1) 4 bytes (plus 8 bytes if the optional trace is requested) where n is the number of DCBs being temporarily closed. This routine saves the address of the TCLOSE parameter list in the first word of the area.	IGC0002C	TCM33000
8 After initialization has been performed for each DCB, this function transfers control to the trace initialization function if indicated by the trace bit in the WTGPATHS field.	IGC0002C	TCM35000
9 Checks the UCBTBYT3 field to determine which TCLOSE function to transfer control to: (TCLOSE final Function) for unit record or teleprocessing devices, TCLOSE direct-access function, or TCLOSE tape function.)	IGC0002C	TCM21000 TCM24000 TCM25000
If the JFCB indicates NSL processing for a tape data set, this routine transfers control to the Close Tape ---Label Handling (NSL) function. Otherwise, control is passed to the TCLOSE tape function	IGC0002C	TCM25500

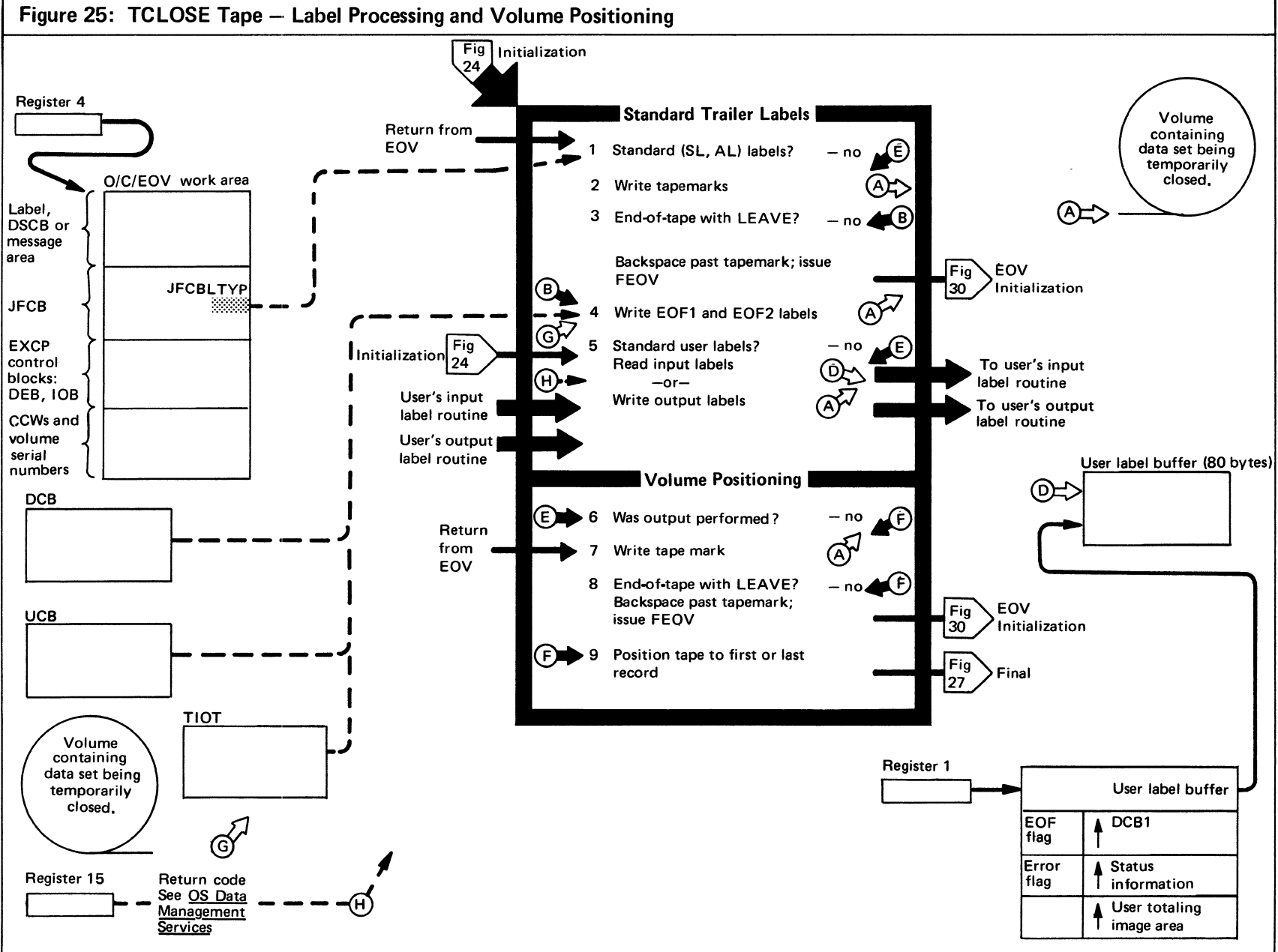
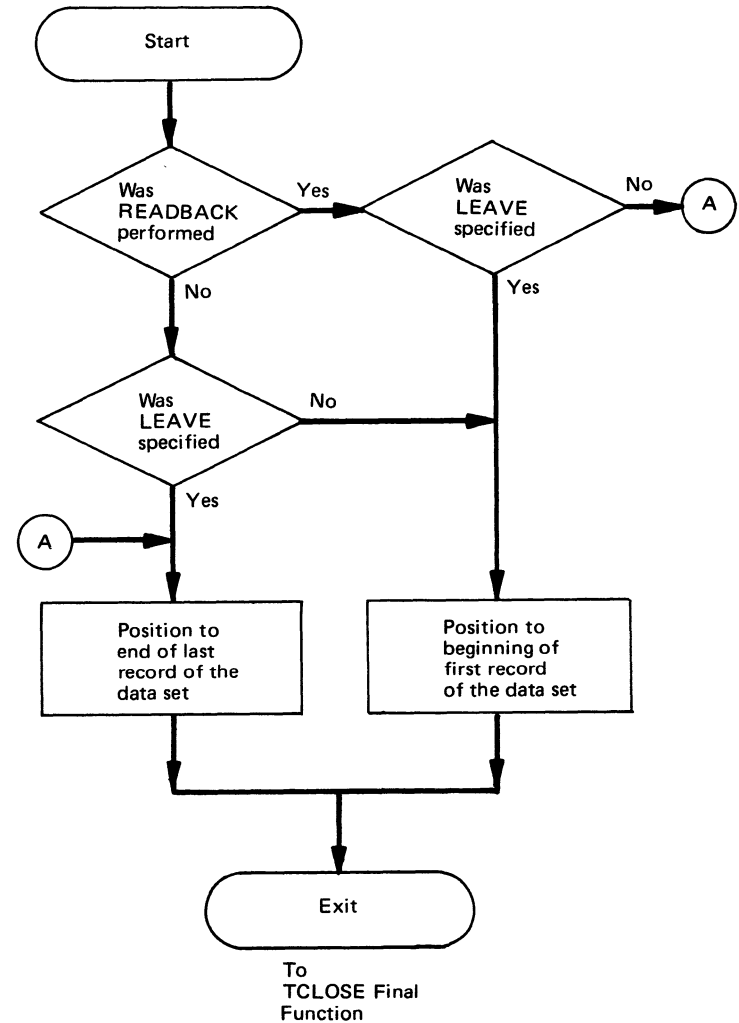


Figure 25. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 Process standard labels for output tape data sets being temporarily closed. One tapemark is written to delimit the data. This function is reentered from FEOV processing (step 3) if a new volume has been mounted.	IFG0232G	TCT00400
3 For SL and AL tape data sets, a test is made to determine whether end-of-tape was sensed when the tapemark was written, and, if so, whether LEAVE was specified; if both conditions are met, the tape is backspaced past the tapemark and an FEOV macro instruction is issued to write trailer labels, mount a new volume, and write header labels on the new volume, after which normal TCLOSE processing is continued.	IFG0232G	TCT00600
4 Writes EOF1 and EOF2 labels following the tapemark (data delimiter) using information from the DCB, UCB, JFCB, and TIOT DD entry.		
5 This function is entered from TCLOSE initialization for input data sets on tape. If user labels are requested and if the data set was opened for input, this function reads the label into the user's label buffer and passes control to the user's label routine to verify the label. 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 function to write the label.	IFG0232M	TCT07200
6 Entered after user-label processing and for NL tape data sets, this function determines whether the user wrote a user label to tape.	IFG0232S	TCT14200
7 If a label was written, two tapemarks are written to delimit either the data (NL tape) or the labels (SL or SUL tape).	IFG0232S	TCT14400
8 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 are met, the tape is backspaced past the tapemark and an FEOV macro instruction is issued to write trailer labels, mount a new volume, and write header labels on the new volume, after which normal TCLOSE processing is continued.	IFG0232S	TCT14600
9 The tape volume is positioned as follows:		



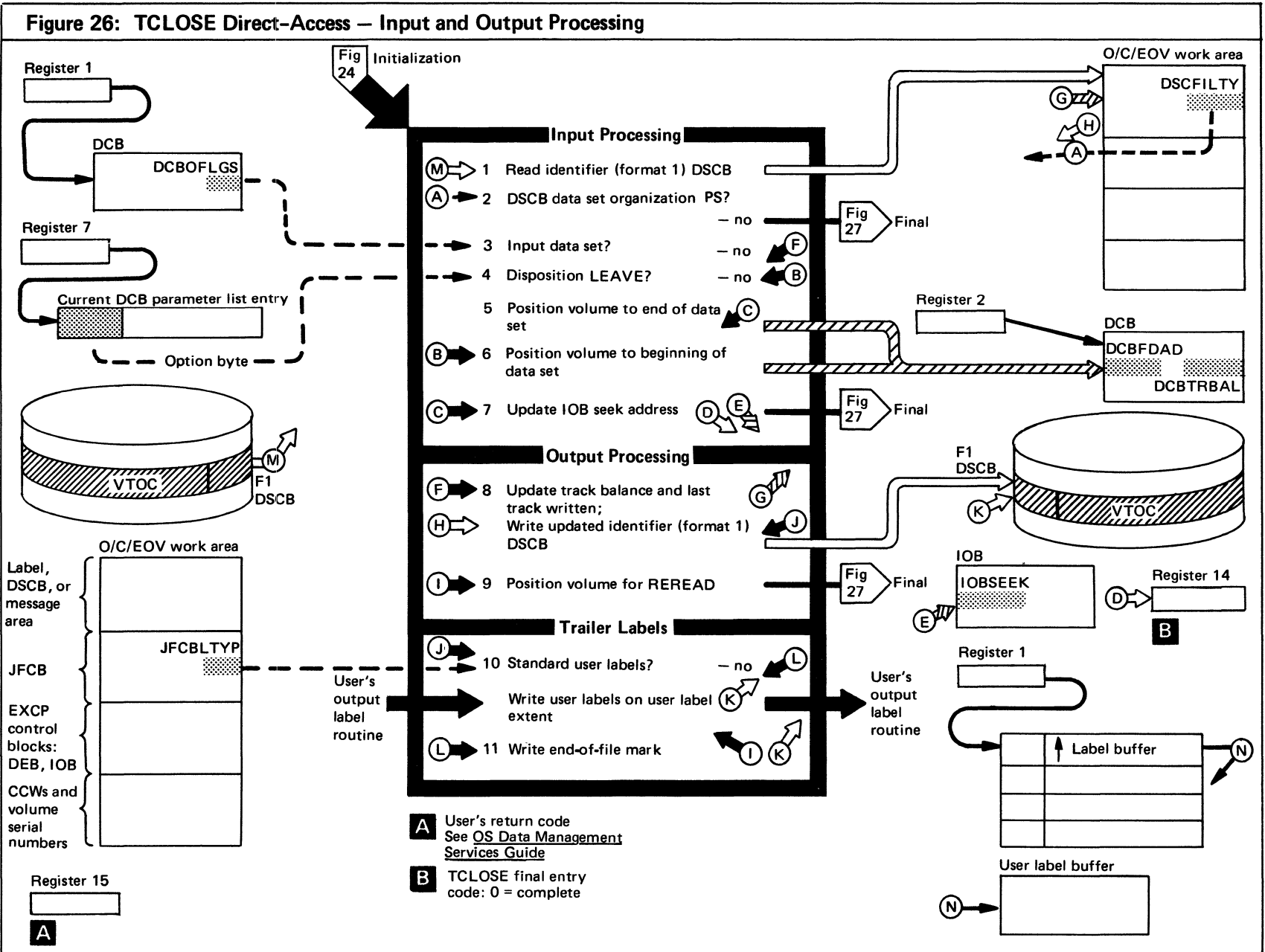


Figure 26. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 This function is entered from TCLOSE initialization for both input and output data sets on direct-access devices. This function reads the data portion of the identifier (format 1) DSCB into the O/E/EOV work area.</p> <p>2 The data set organization indicator in the identifier (format 1) DSCB is checked to verify that the data set is physical sequential (PS). The indicator is in the DSCFILTY field of the DSCB. If the data set organization is not PS, an indicator is set in the work area (the DXCCW field) to indicate that no more processing should be performed, and an exit is taken to the TCLOSE final function.</p>	IFG0232D	TCD00100
<p>4 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 DCB track balance (DCBTRBAL) to reflect the number of bytes remaining on the track.</p>	IFG0232D	TCD01500

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>6 If the disposition of an input data set is REREAD, 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.</p>	IFG0232D	TCD00200
<p>7 The full disk address in the IOB field is updated.</p>	IFG0232D	TCD00300
<p>8 If an output data set is being temporarily closed, this function updates the track balance field (DSCTRBAL) and the last-track written field (DSCSTAR) and then writes the updated DSCB back to the VTOC.</p>	IFG0232D	TCD00300
<p>9 Performs standard user label processing for output data sets on direct-access devices: builds the SUL interface, transfers control to the user's label routine, and when control is returned, writes the user trailer label to the user-label extent.</p>	IFG0232J	TCD03100

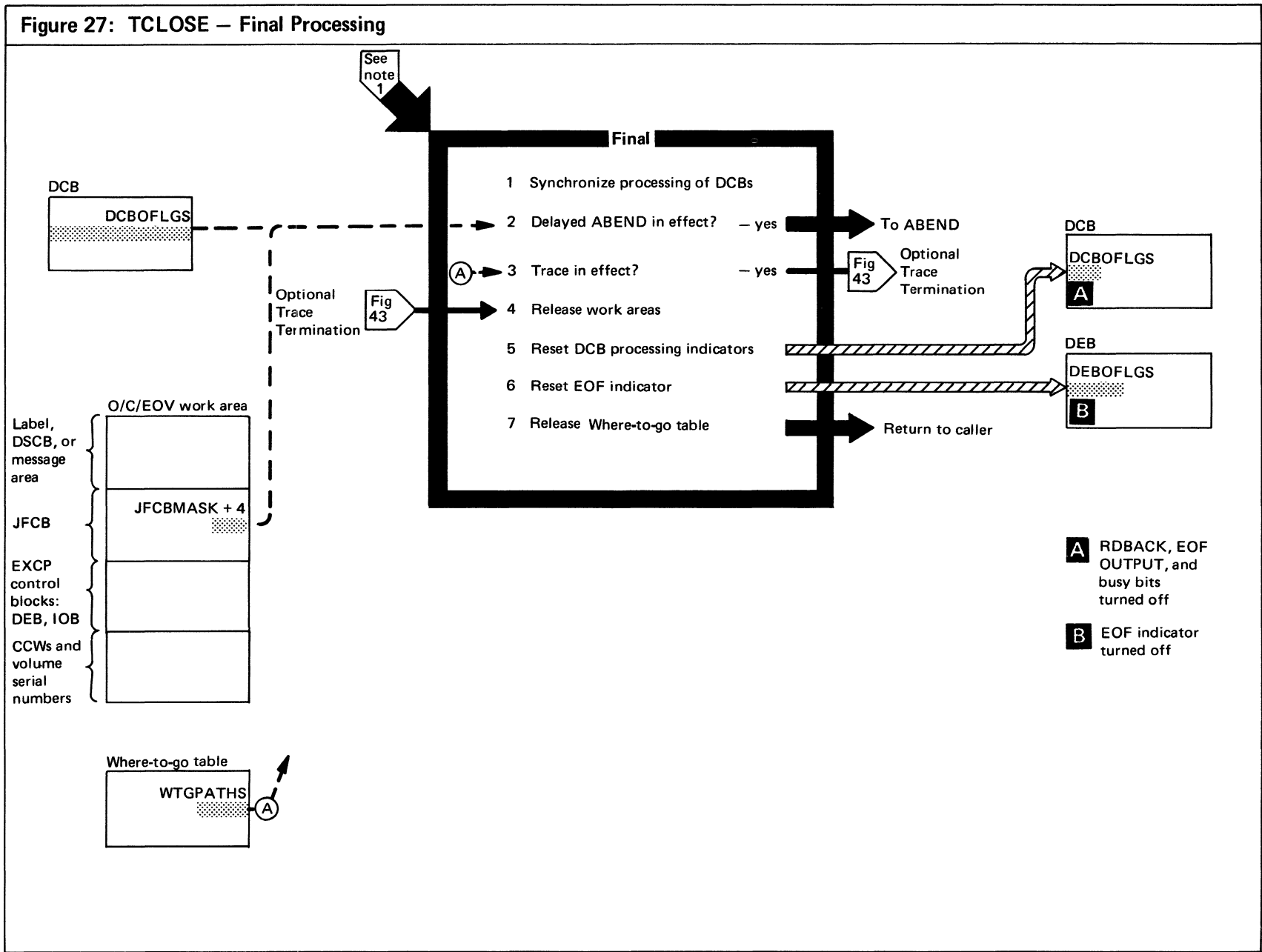


Figure 27. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 Verifies that each DCB represented in the parameter list has been processed by each of the preceding TCLOSE functions and is now ready for final processing.	IFG0232Z	TCM50000
2 Determines whether the user has requested a delayed ABEND (DSCBOFLGS=X'01' and JFCBMASK+4=X'02'); if so, issues an ABEND macro instruction to terminate processing.	IFG0232Z	TCM50500

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 Determines whether a trace is in effect; if so, transfers control to the optional Trace Termination function.		
4 Releases DCB and resident routine work areas.	IFG0232Z	TCM53000
7 When all DCBs represented in the TCLOSE parameter list have been processed, this function releases the Where-to-go table and exits to the user via an SVC 3.	IFG0232Z	TCM54500

Figure 28: EOJ Contents

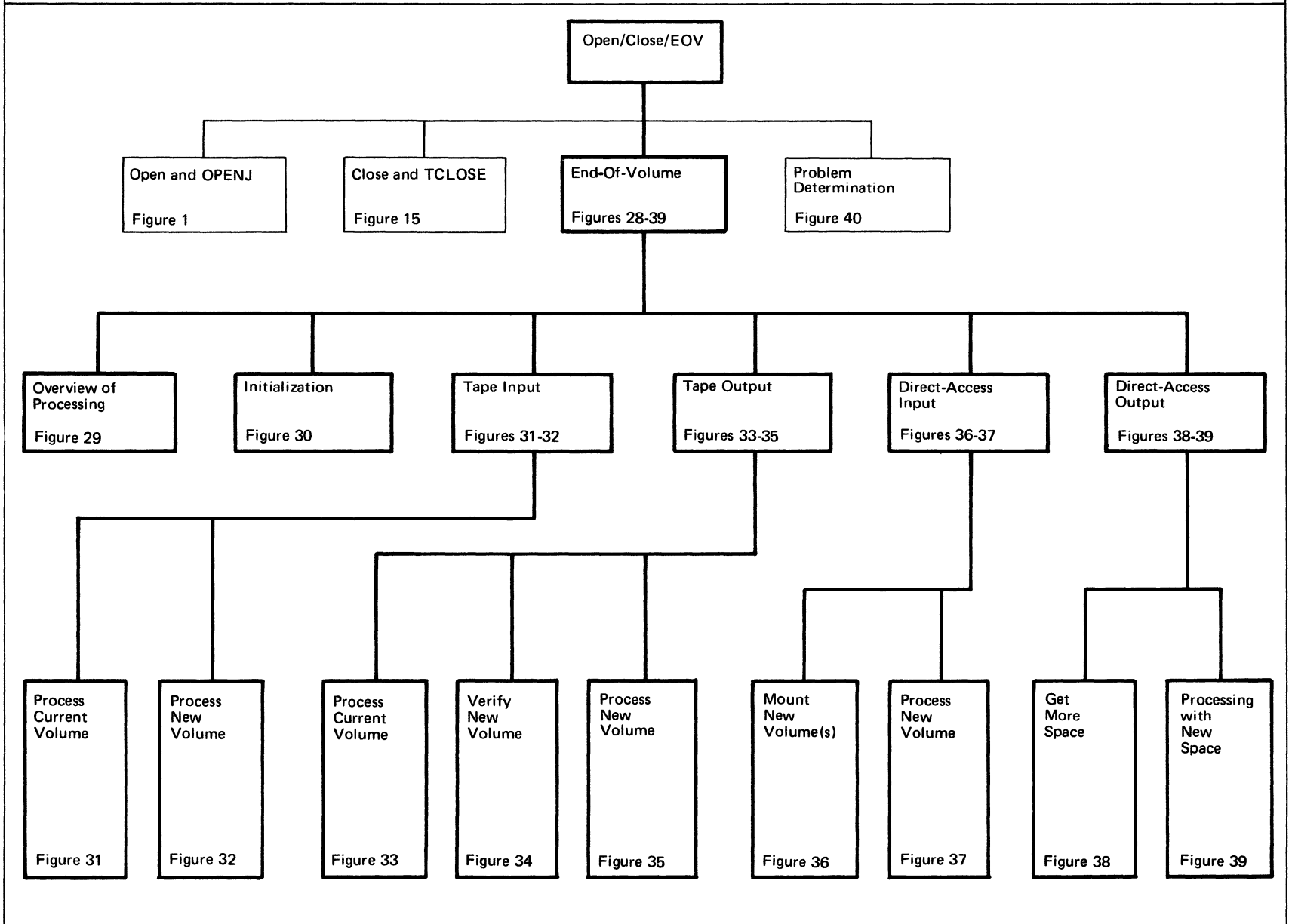


Figure 29: EOVS—Overview of Processing

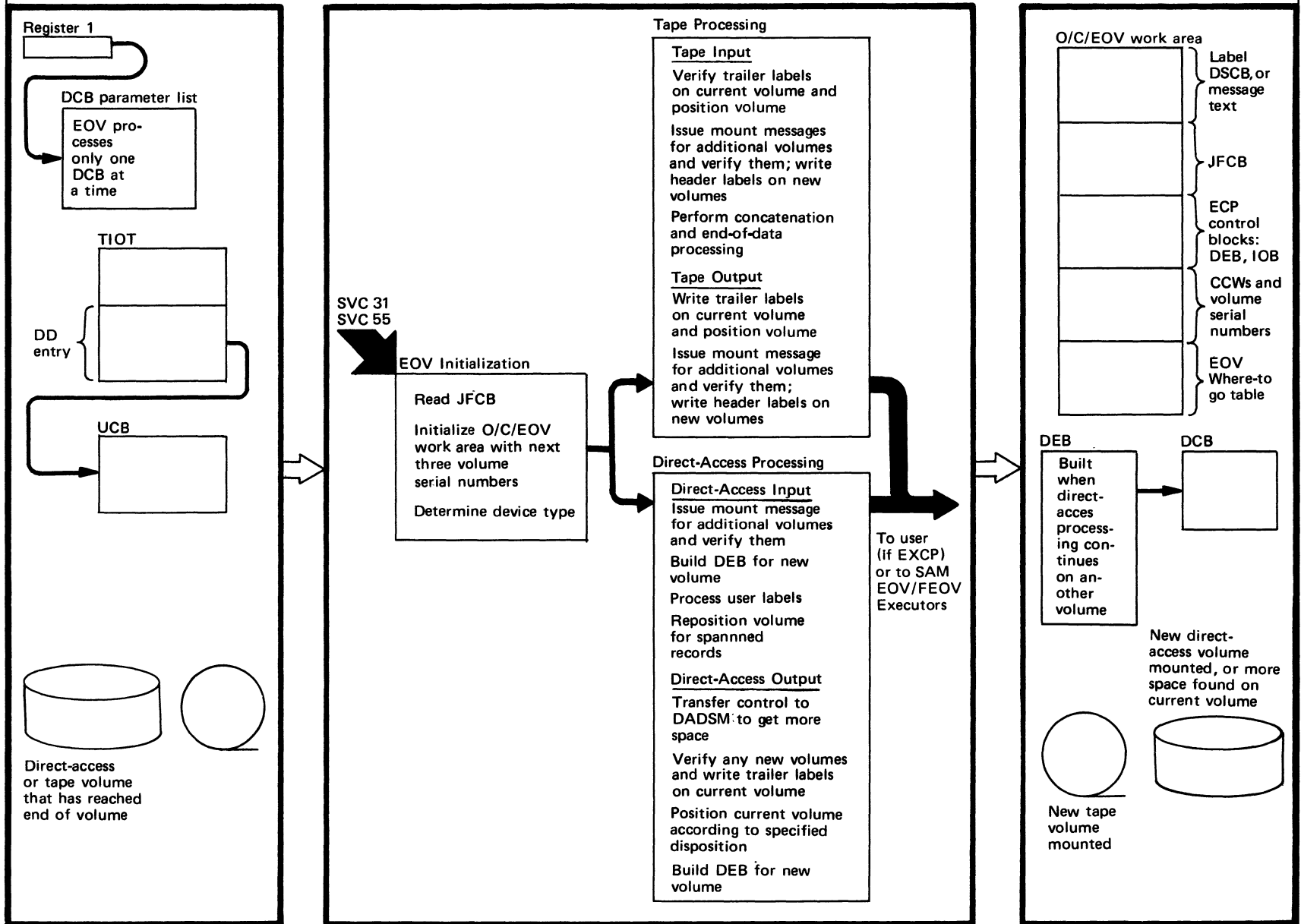


Figure 30: EOV Initialization

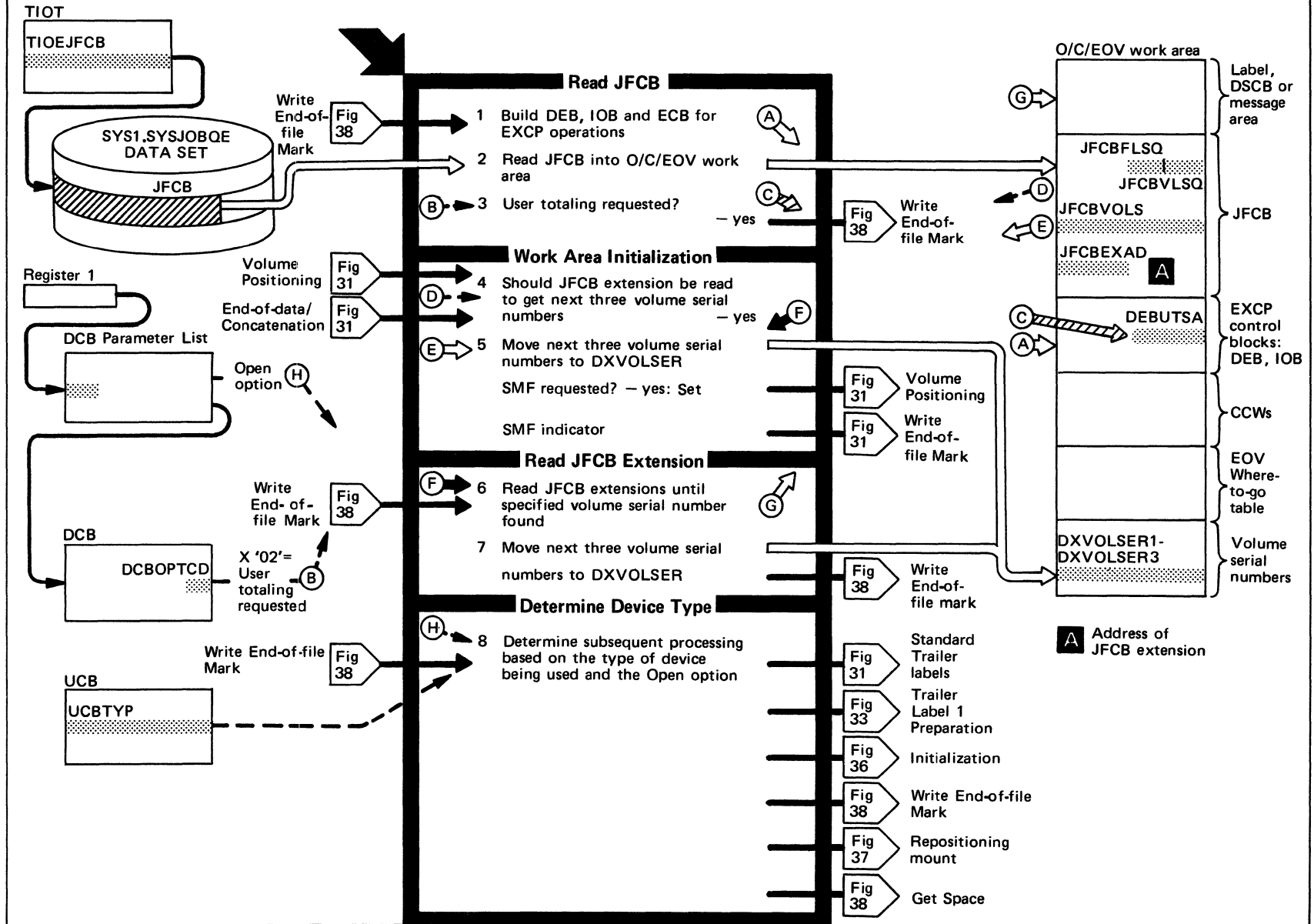


Figure 30. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 The FEOV executor or the SYNAD/EOV executor (both of which are described in <u>OS SAM Logic</u>) issues a GETMAIN macro instruction for the single DCB at end-of-volume. Although Open and Close can process many DCBs in parallel, EOv processes only one DCB at a time.	IGC0003A (FEOV)	EIN03400
	IGC0005E (SYNAD/EOV)	EIN01800
3 The last two bytes of the DEBUTSA field in the DEB now point to the last record of the last block written on the volume.	IFG0551F	EIN12300
4 The JFCBVOLS field can contain as many as five volume serial numbers. Subsequent volume serial numbers are contained in JFCB extensions, which can contain as many as fifteen volume serial numbers each.	IFG0551H	EIN14400
The JFCB volume sequence field (JFCBVLSQ) is added to the DEB volume sequence field to determine the sequence number of the volume to be mounted next.	IFG0551H	EIN14100
The System Management Facility (SMF) provides information on the use of system resources.		

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
5 The volume serial numbers of the three consecutive volumes to be processed, beginning with the volume currently at end-of-volume, are moved from the JFCB into the DXVOLSER field in the O/C/EOV work area.	IFG0551H	EIN14400

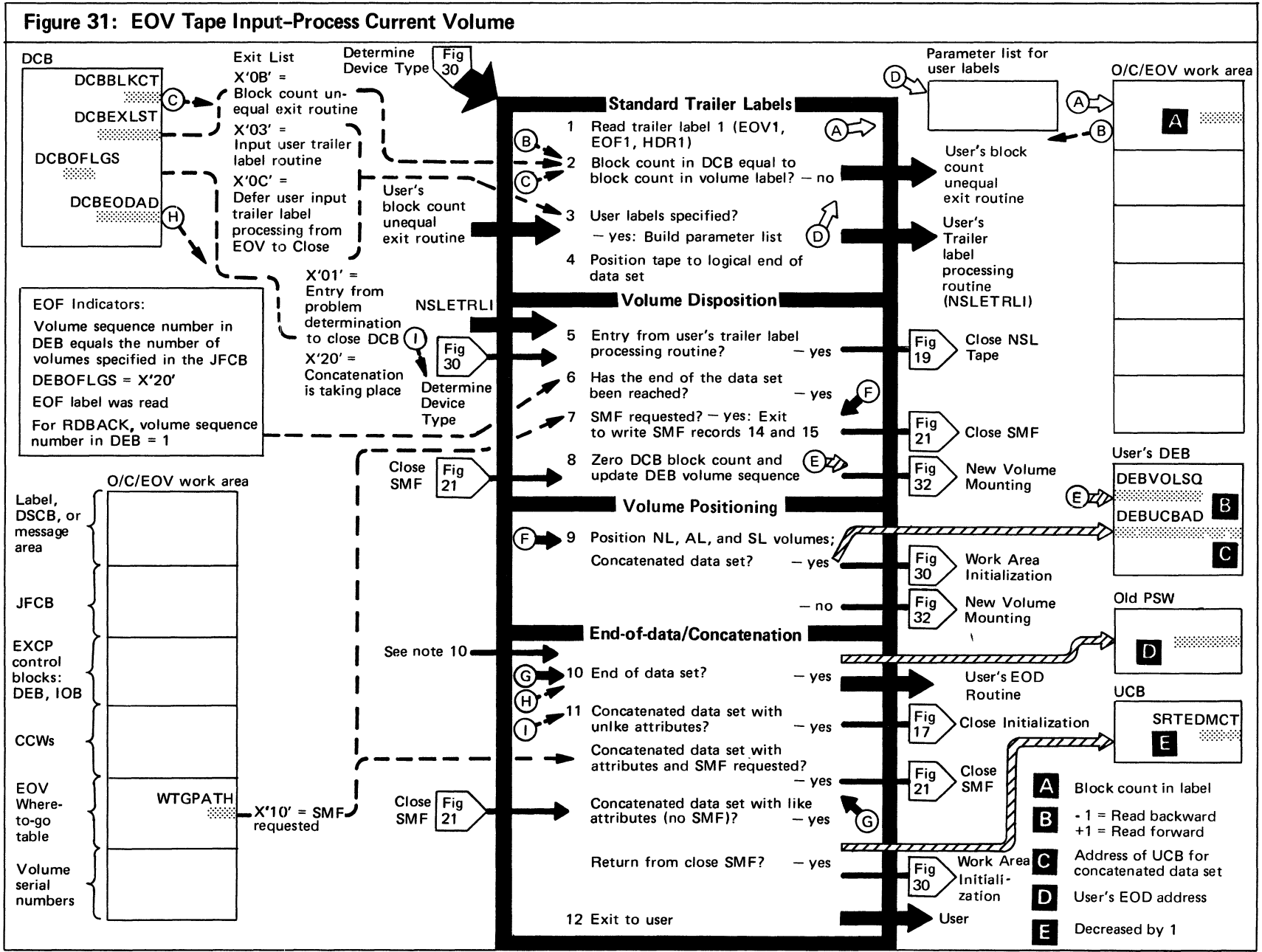


Figure 31. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 Instructions for writing user routines are documented in <u>OS Data Management Services Guide</u> .		
3 The parameter list passed to the user for trailer label processing is described in <u>OS Data Management Services Guide</u> . Before control is passed to the user, the lock bit in DCBOFLGS is set to 0; when the user returns control, the lock bit is reset to 1. The user indicates that additional user labels are to be read by specifying a return code of 0 in register 15; a return code of 4 indicates that user label processing should be stopped. User labels are not processed if deferred label processing has been specified as a DCB exit list entry.	IFG0552T	ETI04100
4 The logical end of the data set for read-forward operations is the interblock gap following the trailer labels. The logical end of the data set for a read-backward operation is the interblock gap preceding header label 1.	IFG0552R	ETI01900
	IFG0552T	ETI05900

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
9 NSL volumes are rewound and unloaded. See the program organization figure for module IFG0552V for a detailed description of how the tape is positioned.	IFG0552V	ETI08500
10 This function is entered from the following functions: Work Area Initialization (Figure 30) Standard Header Label (Figure 32) ABEND Interpretation and Recovery (Figure 42)	IFG0552X	ECM60500
11 If concatenated data sets have unlike attributes, the DCB is closed and then reopened with new attributes. Problem Determination enters this function if an error occurred in an EOVS function and the user indicated that it should be ignored.	IFG0552X	ECM60900
12 EOVS determines by process of elimination, that the data set resides on a unit record device.		

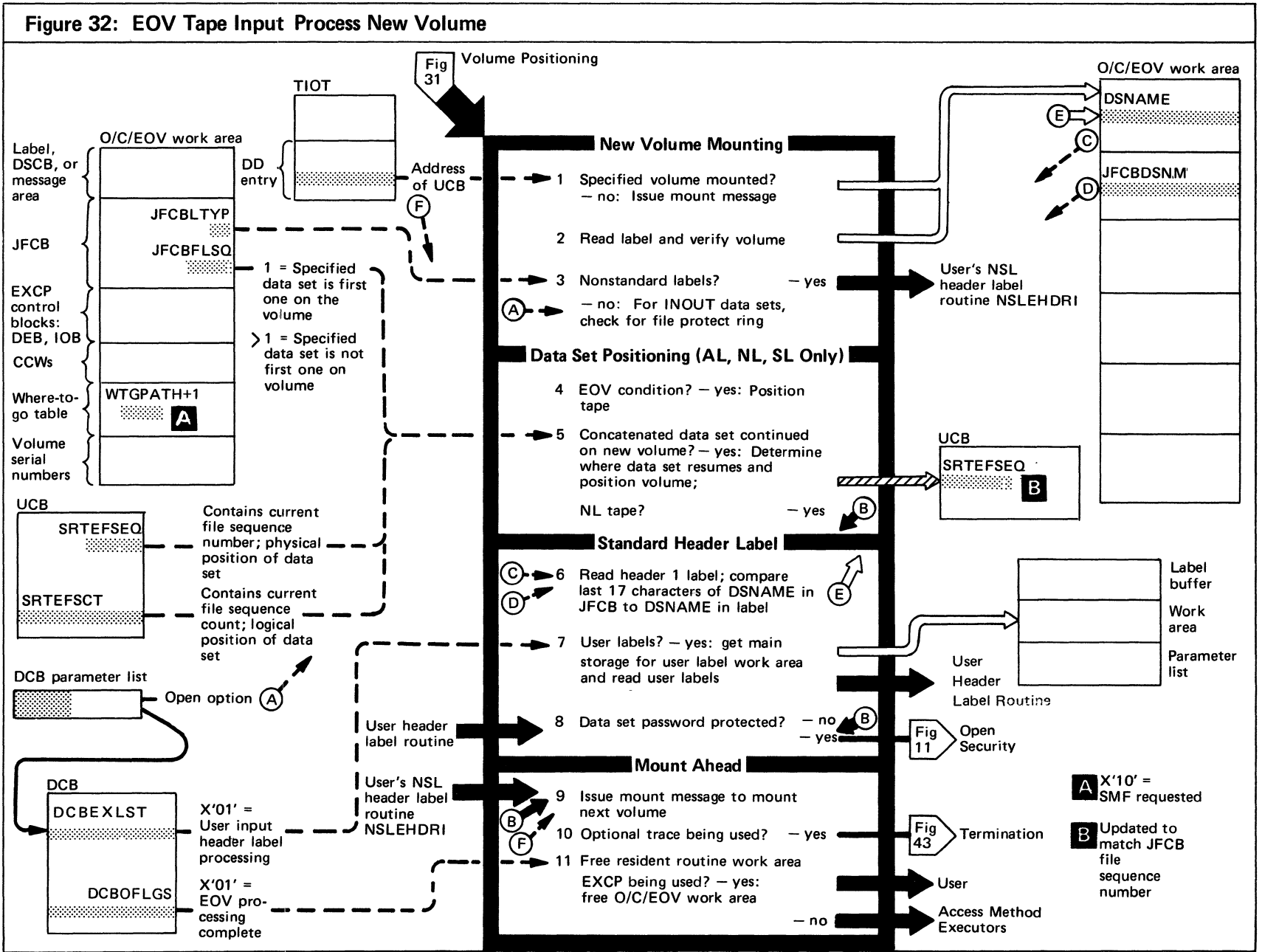


Figure 32. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 If the specified volume is not mounted and an empty unit is available, a mount message is issued for the specified volume. If an empty unit isn't available, the first unit with an unreserved volume is selected. The unreserved volume is unloaded and a message about its state is issued. A mount message is then issued for the specified volume.	IFG0552Z	ETI13100
2 The volume serial number is verified for SL and AL tapes. If a tapemark is read, the tape may be a DOS NL tape with a leading tapemark. If a volume label is not read, or if a volume serial number cannot be verified, the volume is demounted and a mount message is issued for the appropriate volume. If no volume label is read for NL or NSL tapes, it is assumed that the correct volume is mounted.	IFG0552Z	ETI15300

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 For NSL tapes, an exit is then taken to NSLEHDRI to verify the volume and position the tape to the data set. If a file-protect ring is required, the tape is unloaded, the ring is inserted, and standard volume labels are reverified.	IFG0553D	ETI18200
4 An NL tape is positioned either to the interblock gap preceding the first data record for a read-forward operation or to the interblock gap following the last data record for a read-backward operation. An AL or SL tape is positioned either to the interblock gap preceding header label 1 for a read-forward operation or to the interblock gap preceding trailer label 1 for a read-backward operation.	IFG0553D	ETI19900

Figure 33: EOV Tape Output-Process Current Volume

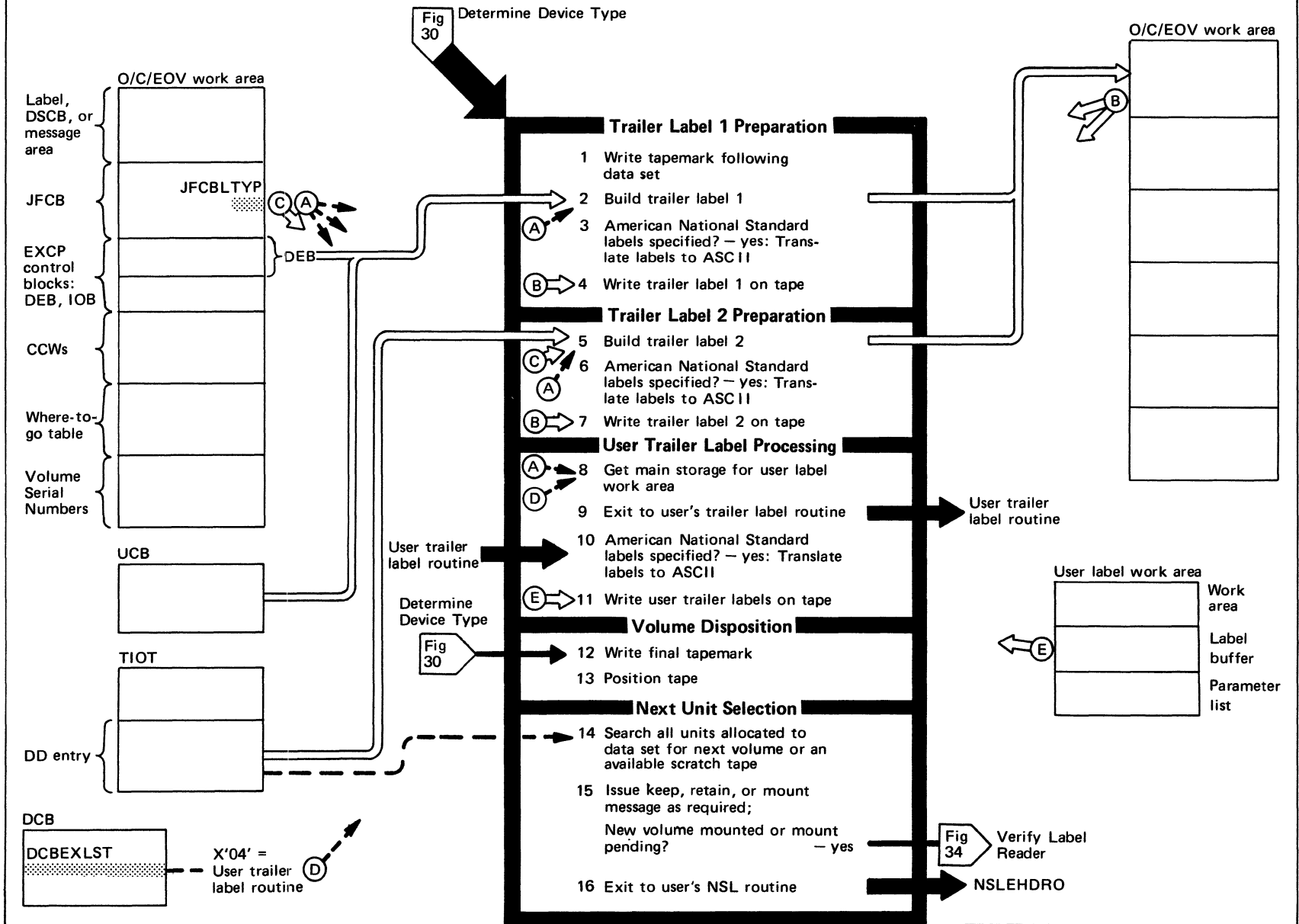


Figure 33. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
12 The tapemark is written following the data set on an NL tape or following the trailer label on SL tape. For an American National Standard or NL tape, two tapemarks are written	IFG0551T	ETO05200
13 Positioning and disposition options are processed in the following order: (a) LEAVE or REWIND (specified by the FEOV executor in byte 0 of the register pointing to the current DCB). If REWIND was specified, the tape is rewound to the load point and the logical sequence count is decreased by the number of data sets on the volume minus one. The physical sequence count is set to one.	IFG0551T	ETO05700

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
13 (b) Open options specified in DEBOPATB field of DEB. If REREAD is specified, the tape is positioned to the first data record on an NL tape or to the HDR1 label for an SL tape.	IFG0551T	ETO05700
If KEEP, CATLG, or UNCATLG is specified, the volume is rewound and unloaded, the UCB sequence and volume serial number fields are set to zero to indicate that no volume is mounted, and a "keep" or "retain" message is sent to the system operator. This is the only positioning performed for NSL tapes.	IFG0551T	ETO06300
(c) User-requested disposition (specified in the last 2 bits of the TIOESTTA field in TIOT).		
(d) If none of the above is specified, the default is LEAVE.		

Figure 34: EOVS Tape Output-Verify New Volume

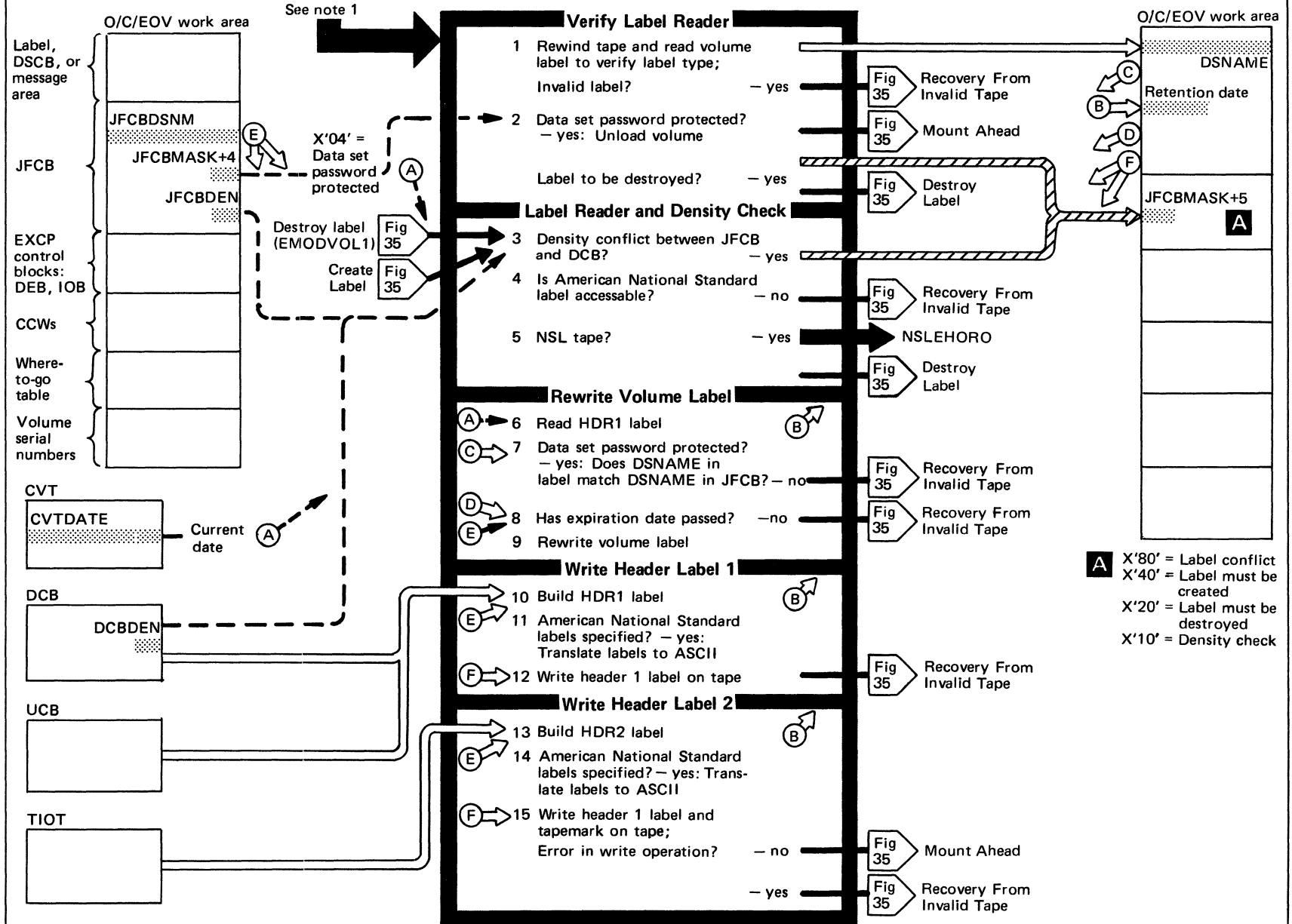


Figure 34. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 This function is entered from the following functions:</p> <p>Next Unit Selection (Figure 33) Destroy Label (EMODVOLI) (Figure 35) Create Label (Figure 35) Recovery From Invalid Tape (Figure 35)</p>	IFG0551X	ETO10100

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>3 If control returned to this function with the density check bit on in the JFCB, the return is from the Destroy Label (EMODVOL1) function. The density in the JFCB is used. Density is not checked for NL tapes.</p>	IFG0551X	ETO11400

Figure 35: EOV Tape Output--New Volume Processing, continued

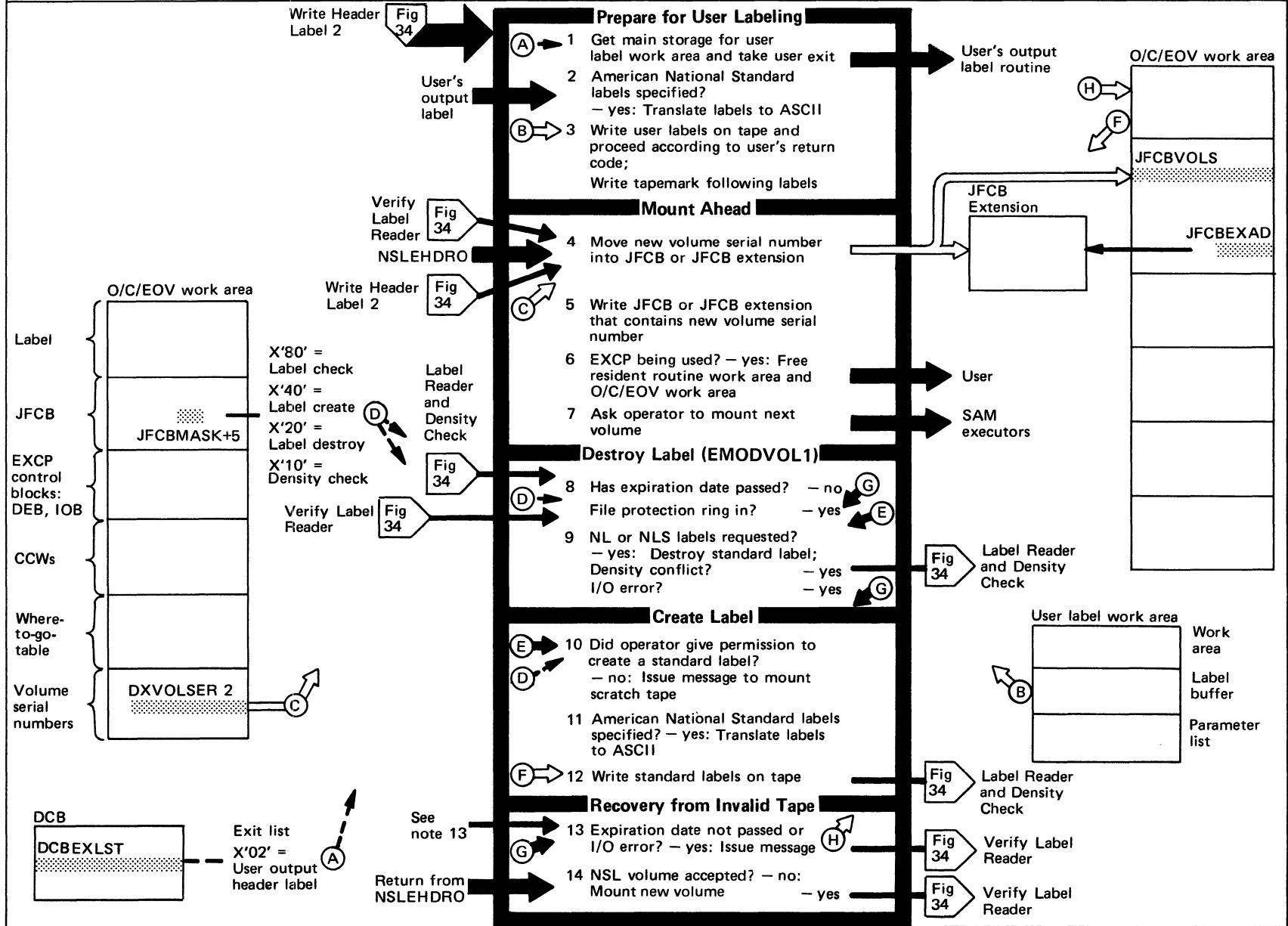


Figure 35. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 A return code of 4 means "stop user label processing." Additional user labels are read if the return code is 0. (Register 15 contains the return code.)	IFG0552F	ETO22700
9 This function receives control from the Label Reader and Density Check function if either a label error or density conflict exists. If the entry is for density conflict, control is immediately returned to the calling function. If the expiration date has passed or a volume with an American National Standard label was mounted, the operator's permission is needed to destroy the label. If permission is granted and an NL tape was requested, a tapemark is written over the volume label and control is returned to the Label Reader and Density Check function. If permission is granted and a standard or American National Standard label was requested, control is passed to the Label Create function. If the operator denies permission to destroy the label or the label indicates password protection, the serial number in the work area is replaced with "SCRTCH" and control is passed to the Recovery from Invalid Tape function	IFG0552J	ETO27800

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
13 This function is entered from the following EOV functions shown in Figure 34: Verify Label Reader Label Reader and Density Check Rewrite Volume Label Write Header Label 1 Write Header Label 2	IFG0552N	ETO32500
14 This step is entered if the new volume was rejected by the user's NSL output header label routine (NSLEHDRO); the mounting process is repeated for the specified volume.	IFG0552N	ETO33000

Figure 36: EOV Direct-Access Input-Mount New Volume(s)

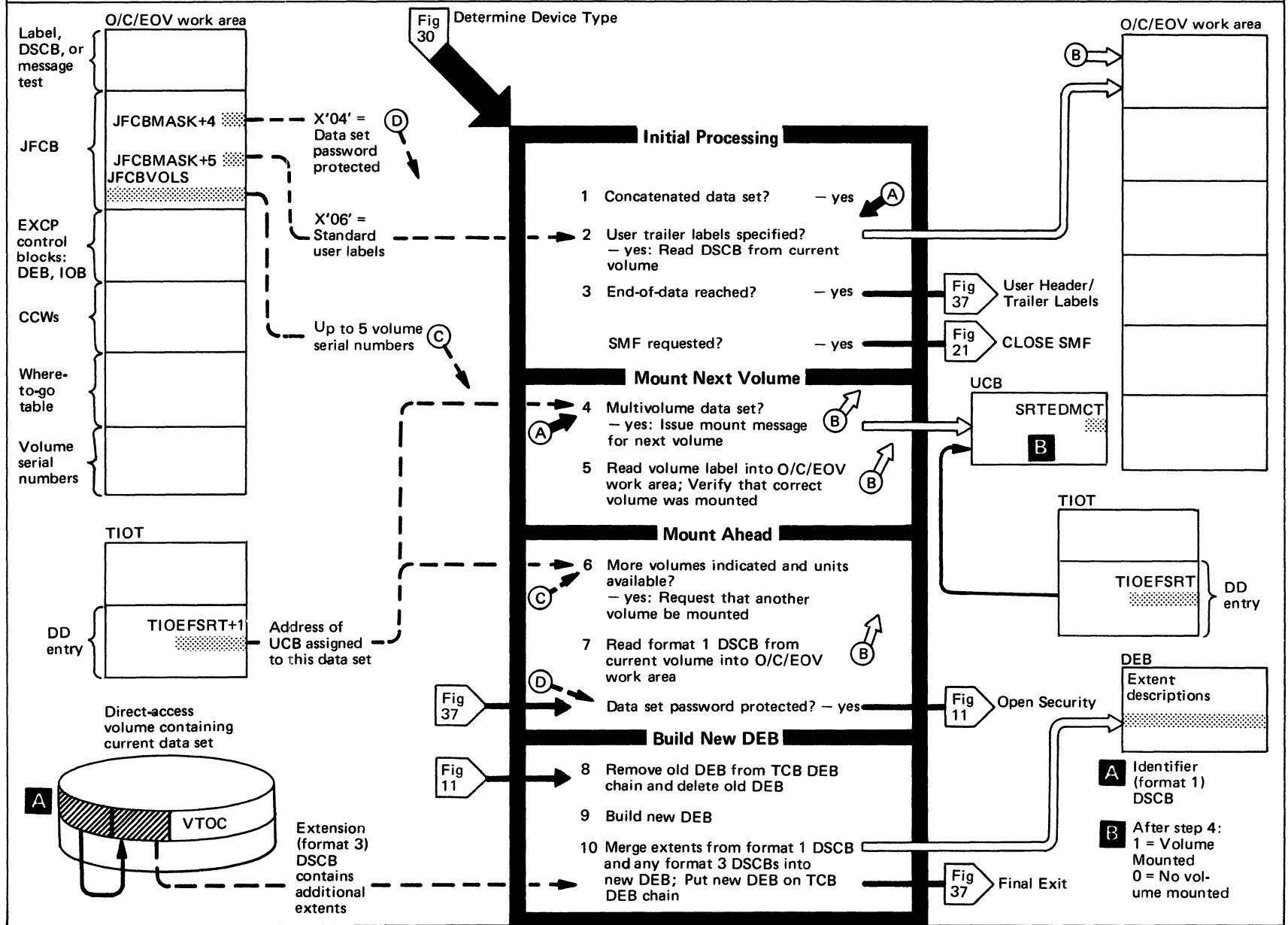


Figure 36. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
3 System Management Facility (SMF) measures the utilization of system resources.		
4 There are 2 modules for this function: one for the IBM 2321 Data Cell and another for other direct-access devices. In both cases, if the next volume of a multivolume data set is not mounted, the UCB addresses in the TIOT DD entries are searched for a unit that is either free or has a demountable volume on it. In the latter case, a demount message is issued. A mount message is then issued for a new volume on that unit.	IFG0553R (2321) IFG0553T (not 2321)	EDI02100 EDI04400

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
7 After the next volume (if any) is mounted, processing continues with the current volume. The format 1 DSCB is read into the O/C/EOV work area. If the data set is password-protected, an exit is taken to Open Security (Figure 11).	IFG0553V	EDI07700
8 A new DEB is built by EOVS every time a new direct-access volume is mounted.	IFG0553X	EDI09200

Figure 37: EOV Direct-Access Input-New Volume Processing

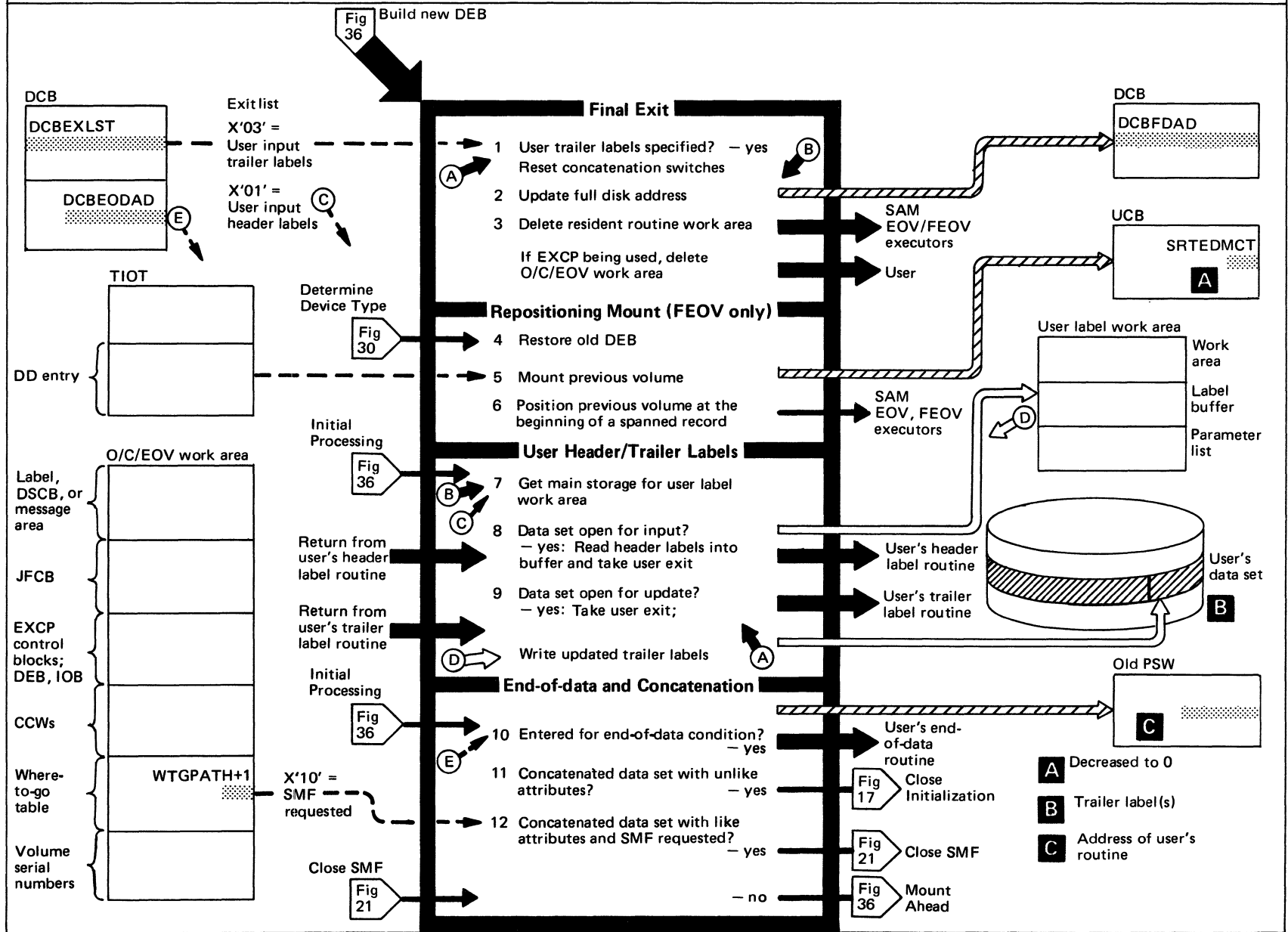


Figure 37. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>4 If an access method executor finds that an American National Standard COBOL record begins on one volume and continues on another (spanned record), it issues an FEOV macro instruction, which passes control to this function. If the record is not being updated, this function only frees the main storage obtained for the new DEB</p> <p>8 The user specifies further EOV processing with one 9 of the following return codes:</p> <p>4 = Stop user label processing 0 = Process additional user labels</p>	<p>IFG0554D (2321) IFG0554B</p> <p>IFG0554L</p>	<p>EDI15000 EDI12500</p> <p>ULEXCP</p>

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>10 Module IFG0552X performs this function for all I/O devices.</p> <p>12 The System Management Facility (SMF) provides information on how the system resources are being used.</p>	<p>IFG0552X</p>	<p>ECM60500</p>

Figure 38: EOV Direct-Access Output—Get More Space

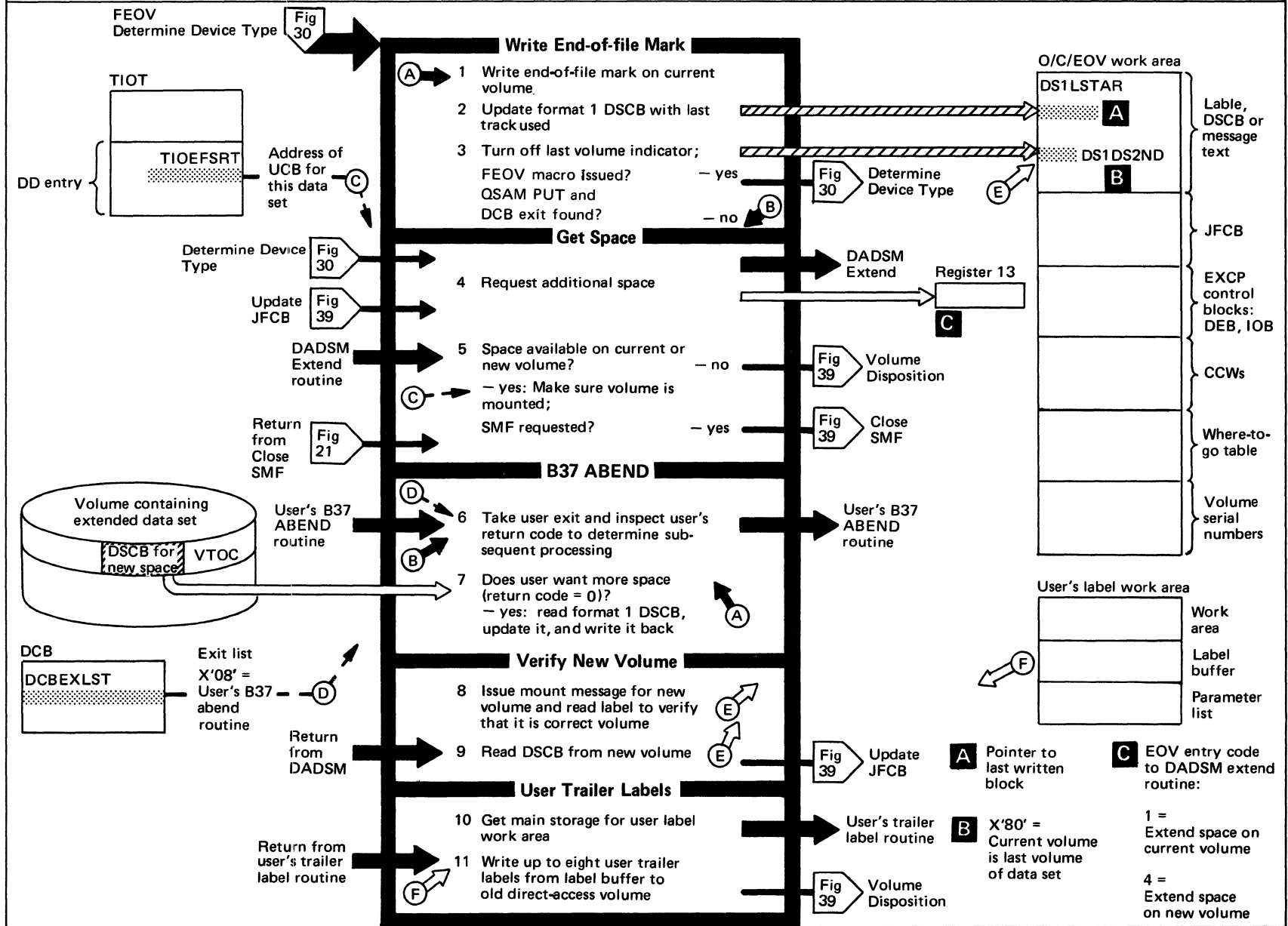


Figure 38. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 If the data set contains fixed-length records, the end-of-file mark is written on both the current track and the next track. If the data set contains variable-length records, the end-of-file mark is written on the next track only. In either case, an end-of-file mark is written only if it fits in the extent allocated to this data set.	IFG0554N	EDO00200
4 The request is processed by space management routines, which place one of the following return codes in register RD: +1 = space was allocated on the current volume +4 = space was allocated on a new volume -1 = no space available on the current volume -4 = no space available on a new volume The amount of space to be allocated is determined by the secondary allocation in the JFCB.	IFG0554P (not 2321) IFG0554R	EDO02700 EDO05600

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
6 The user may return one of the following codes. 0 = transfer control back to DADSM Extend routine to try to get space a second time 1 = return to QSAM processing 2 = issue B37 ABEND	IFG0554T	EDO08800

Figure 39: EOV Direct-Access Output—Processing with New Space

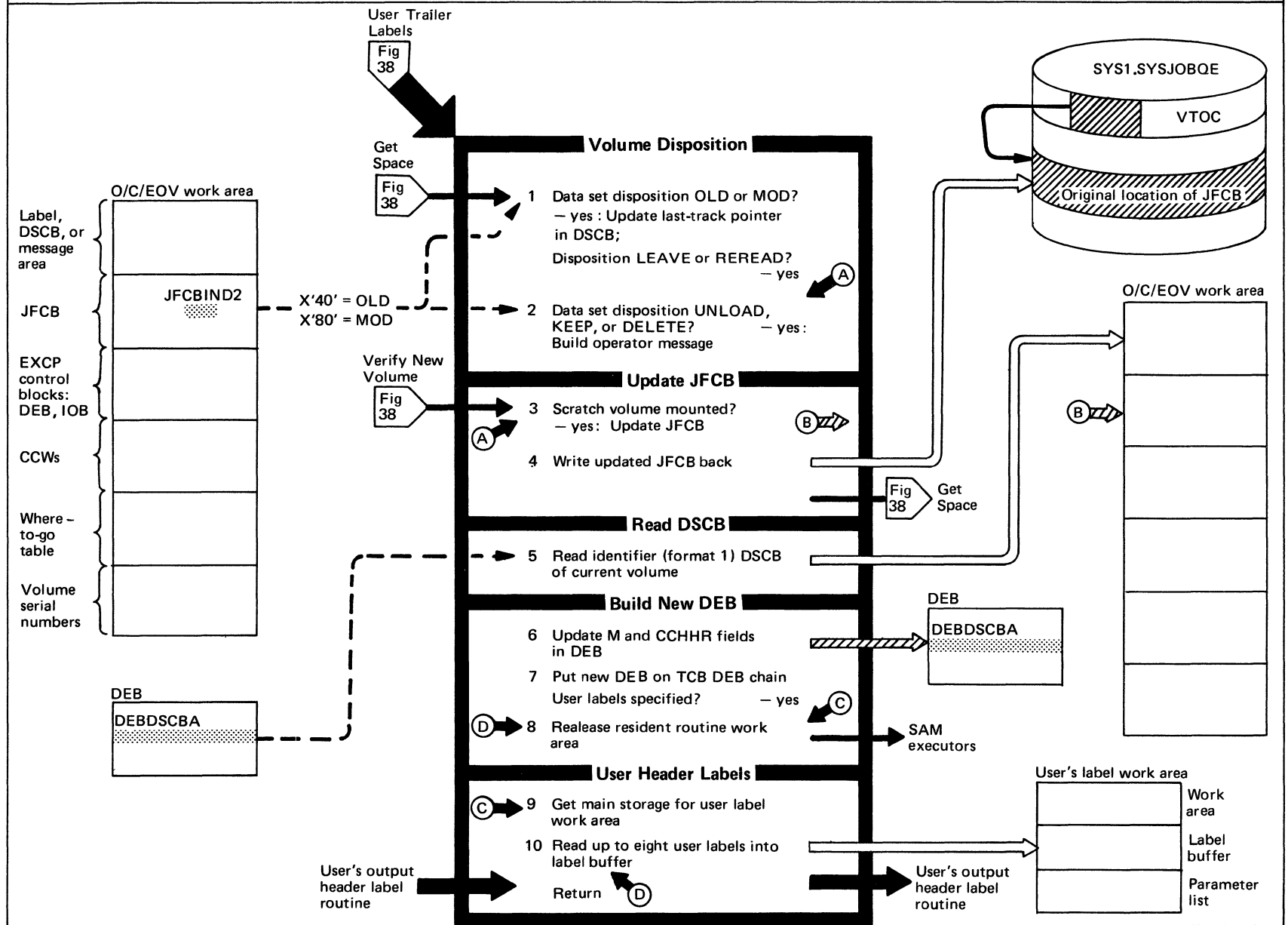


Figure 39. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
1 If the disposition specified is LEAVE or REREAD, the pointer is not changed.	IFG0554Z	EDO15400

Figure 40: Problem Determination Contents

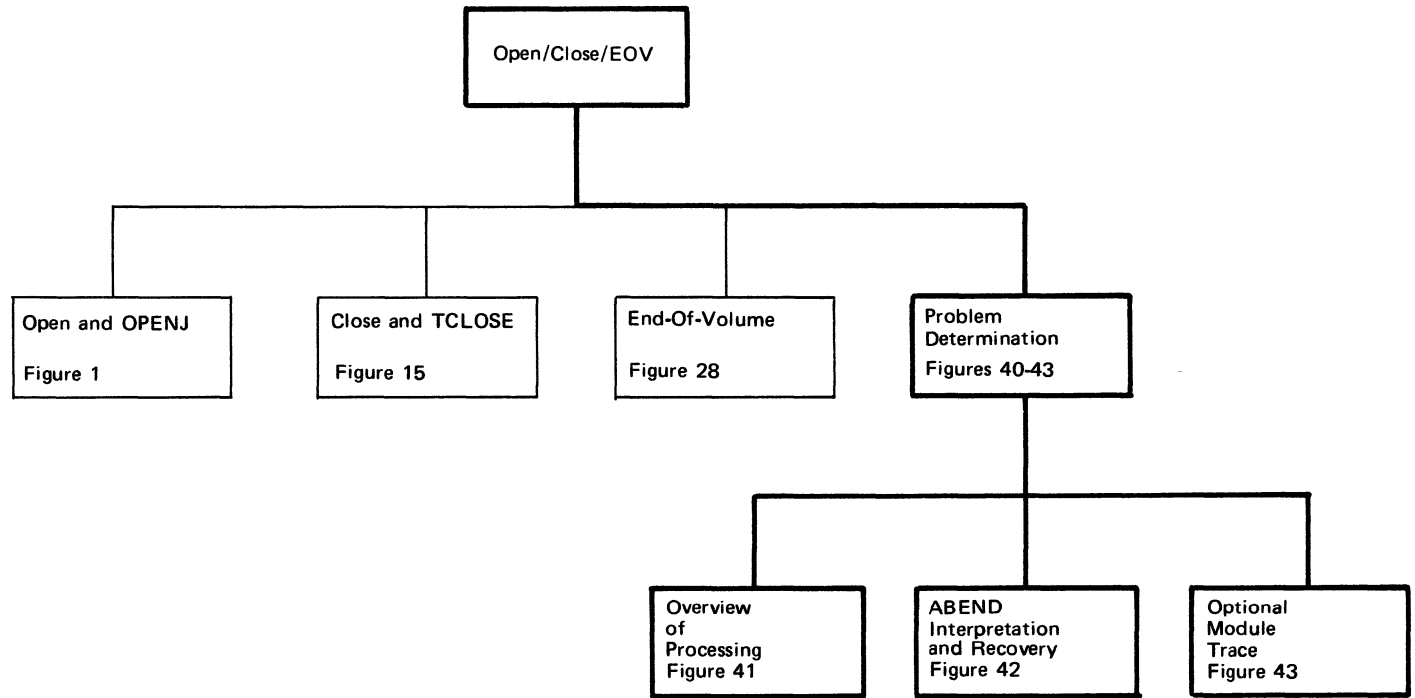
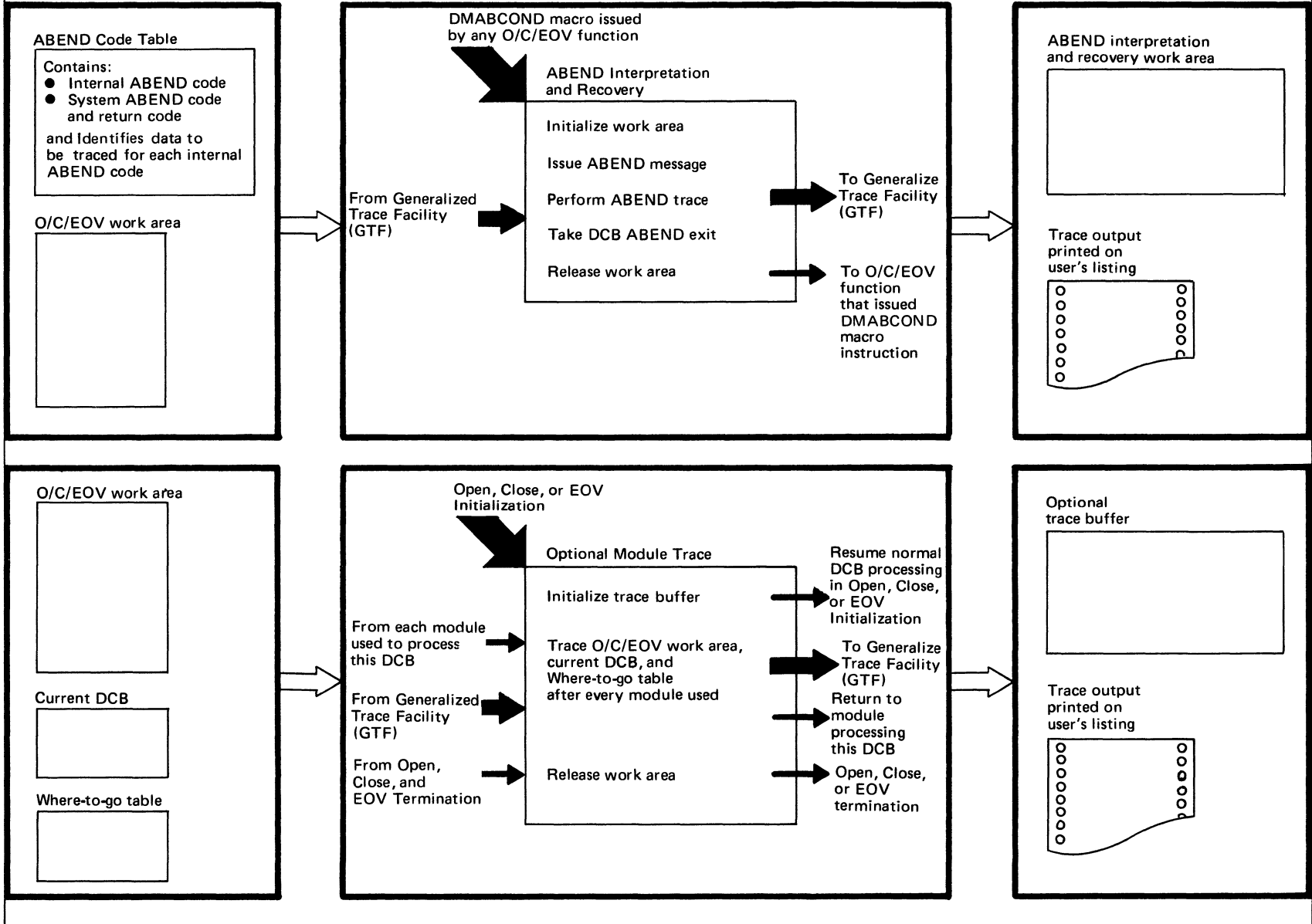


Figure 41: Problem Determination – Overview of Processing



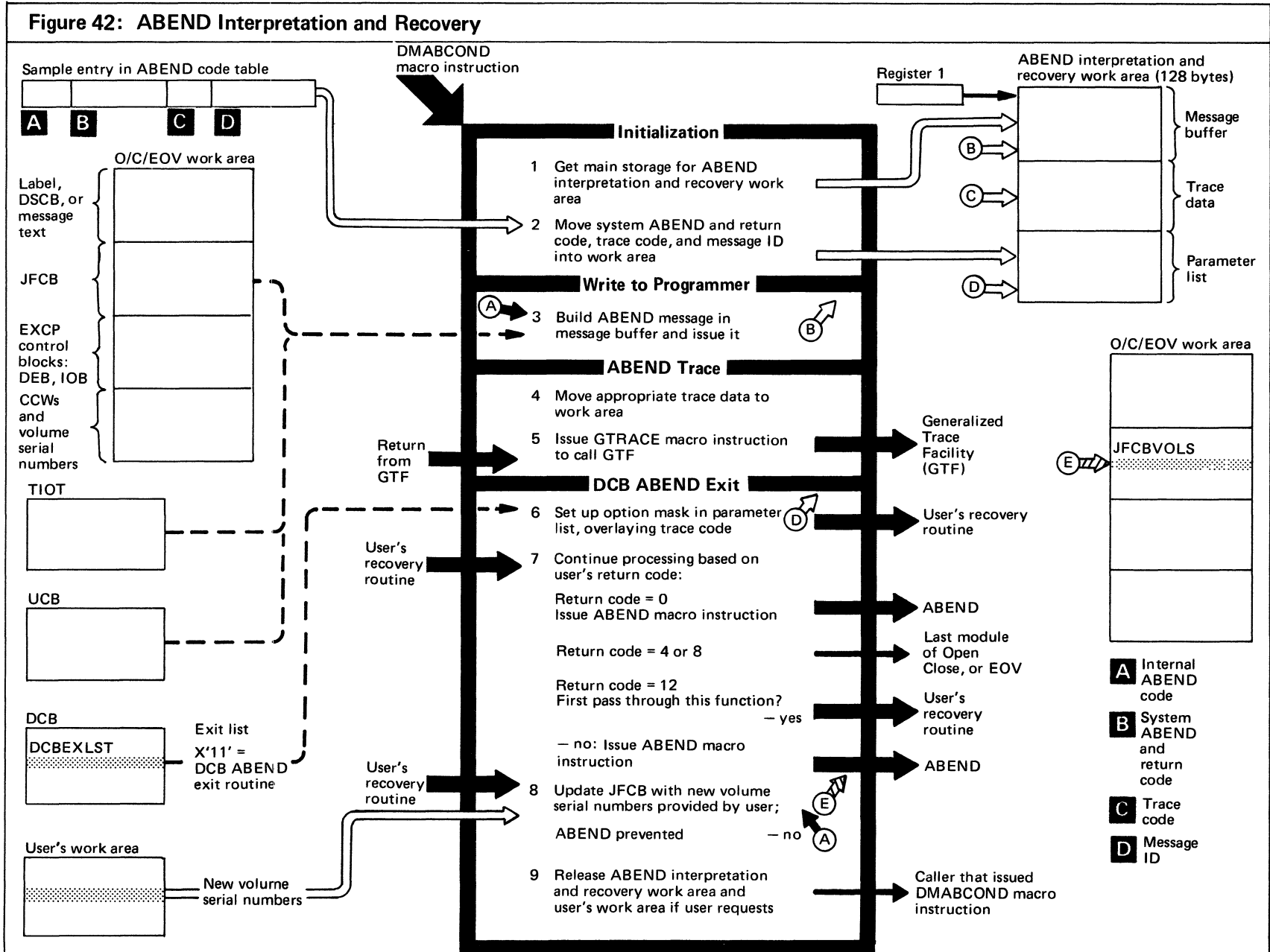


Figure 42. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>1 If an ABEND condition is detected during Open/Close/EOV processing, the module that detected the ABEND condition issues a DMABCOND macro instruction to call ABEND Interpretation and Recovery.</p> <p>The DMABCOND macro instruction passes the internal ABEND code and Open/Close/EOV registers 2 and 4-14 to the Initialization function of ABEND Interpretation and Recovery. Registers 2 and 4-8 are saved in the SVRB extended save area. The Initialization function uses the internal ABEND code in register 0 to refer to a table that contains the system ABEND and return code, a trace code that indicates the type of data to be traced for each ABEND condition, and the ID of the ABEND message issued to the programmer.</p>	<p>IFG0190P (from Open) IFG0200P (from Close) IFG0230P (from TCLOSE) IFG0550P (from EOVS) IFG0196M (from RDJFCB)</p>	
<p>2 If the correct entry is found, the Initialization function issues a conditional GETMAIN for 128 bytes from subpool 250 to be used as a work area. If the GETMAIN is successful, the work area is set to 0; the ABEND code issued to the user, the return code, the trace code, and the message ID are moved into it.</p>		
<p>3 If the user has provided a X '11' DCB ABEND exit routine and his recovery is successful, he still receives a message on his listing. This message refers to the ABEND that would have been issued if recovery had not been successful.</p>		
<p>4 The ABEND Trace function issues a GTRACE macro instruction to call the Generalized Trace Facility (GTF) , which traces data associated with a particular ABEND. The PDCODE field (set by the Initialization function) determines which data is to be traced. GTRACE is issued only if there is an entry in the PDCODE field.</p> <p>All data that has been traced for an Open/Close/EOV ABEND condition is given a format identification of X'EFFF'. The format identification indicates to the IMDPRDMP service aid which of its appendages will format the data. The IMDUSRFF appendage formats data traced by GTF for Open/Close/EOV.</p>	IFG0199D	PDF03800

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
<p>7 A return code of 4 means ignore the ABEND condition", a return code of 8 means " delay the issuing of the ABEND macro."</p> <p>If an invalid code is returned by the user, or if he requests an immediate ABEND, then Open/Close/EOV registers are restored and the ABEND macro instruction is issued.</p> <p>If the user chooses to ignore the error, the open and busy bits of the DCB are turned off (byte 48, bits 2 and 3), the ABEND interpretation and recovery work area is freed, registers are restored, and the ID of the last module of Open, Close, or End-of-Volume is moved into that DCB's WTG Table entry. Then ABEND interpretation and recovery returns to the resident routine.</p> <p>If the user requests a delayed ABEND, the ABEND code and return code are saved in the O/C/EOV work area at DXABCODE, JFCBMASK+4 bit 6 is set to 1, the busy bit is set to 0, and the ABEND interpretation and recovery work area is freed. The ID of the last load module of Open or Close is moved into the WTG Table entry and control is returned to the resident routine. (The combination of bit 6 in JFCBMASK+4 being on and the busy bit being off indicates that a delayed ABEND has been requested by the user.)</p> <p>Instructions for recovery are contained in <u>OS Data Management Services Guide</u>.</p>	IFG0199E	PDF04040
	IFG0199E	PDF04440
	IFG0199E	PDF04420
<p>8 The current volume serial number is the first to be changed. Succeeding volume serial numbers will be changed, depending on the number of new volume serial numbers supplied by the user.</p> <p>The JFCB extension blocks are updated and written back to the job queue. If an I/O error occurs during this process, a message is issued and the user's exit routine will again be given control, and message will no longer be a valid option. If the control blocks are updated successfully, the ABEND interpretation and recovery work area is freed, Open/Close/EOV registers are restored, JFCBMASK+4 bit 6 is set to 1, and control is returned to the module that issued the DMABCOND macro instruction</p>	IFG0199E	PDF04100
	IFG0199E	PDF04160

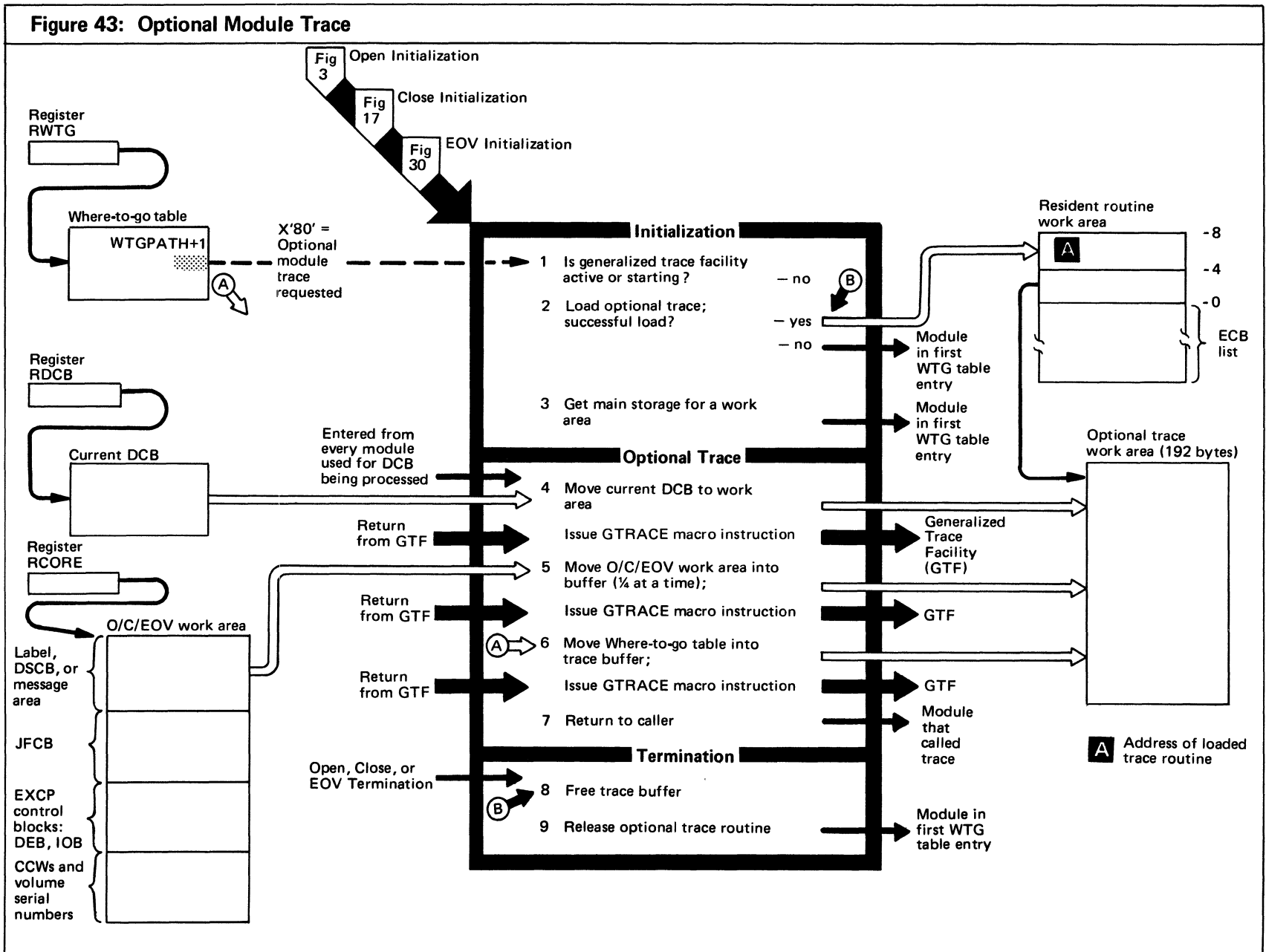


Figure 43. (Continued)

EXTENDED DESCRIPTION	MODULE NAME	CODE LABEL
2 Before a LOAD macro instruction is issued to bring the Optional Module Trace into main storage, a STAE macro instruction is issued to intercept any ABEND conditions in the load routine. If an ABEND condition occurs while the trace is being loaded, the Optional Module Trace will not be initialized.	IFG0199R	FIVE

Legend for Method of Operation Figures

	Shows where to start reading a figure.
	Flow of control within Open/Close/End-of-Volume
	Exit to and entrance from OS or user-written routine not documented in this manual.
	Data flow, usually from a device to main storage, or vice versa.
	Points to bits or fields set or altered in main storage (always output).
	Indicates address chaining.
	Key to a brief description of data field.
	Comes from a data field used as input to a processing step that tests it to determine subsequent processing.
	On-page connectors — always used with one of the above arrow types.
	Off-page connectors from one figure to another.

SECTION 3: PROGRAM ORGANIZATION

Introduction to This Section

In this chapter, Open/Close/EOV is divided into modules, that is, into 1K units of code. The emphasis in this chapter is on the flow of control between modules.

The modules are described in module-flow diagrams, which combine module-description information with flow-of-control information. The functions or parts of functions that are in the modules are indicated at the top of each module description. The “Method of Operation” chapter discusses the functional organization of Open/Close/EOV.

How to use these diagrams and how the diagrams in this chapter relate to the rest of the book are discussed under “How to Use this Book” in the Preface.

This chapter is intended for people who need to know about one or more modules in Open/Close/EOV.

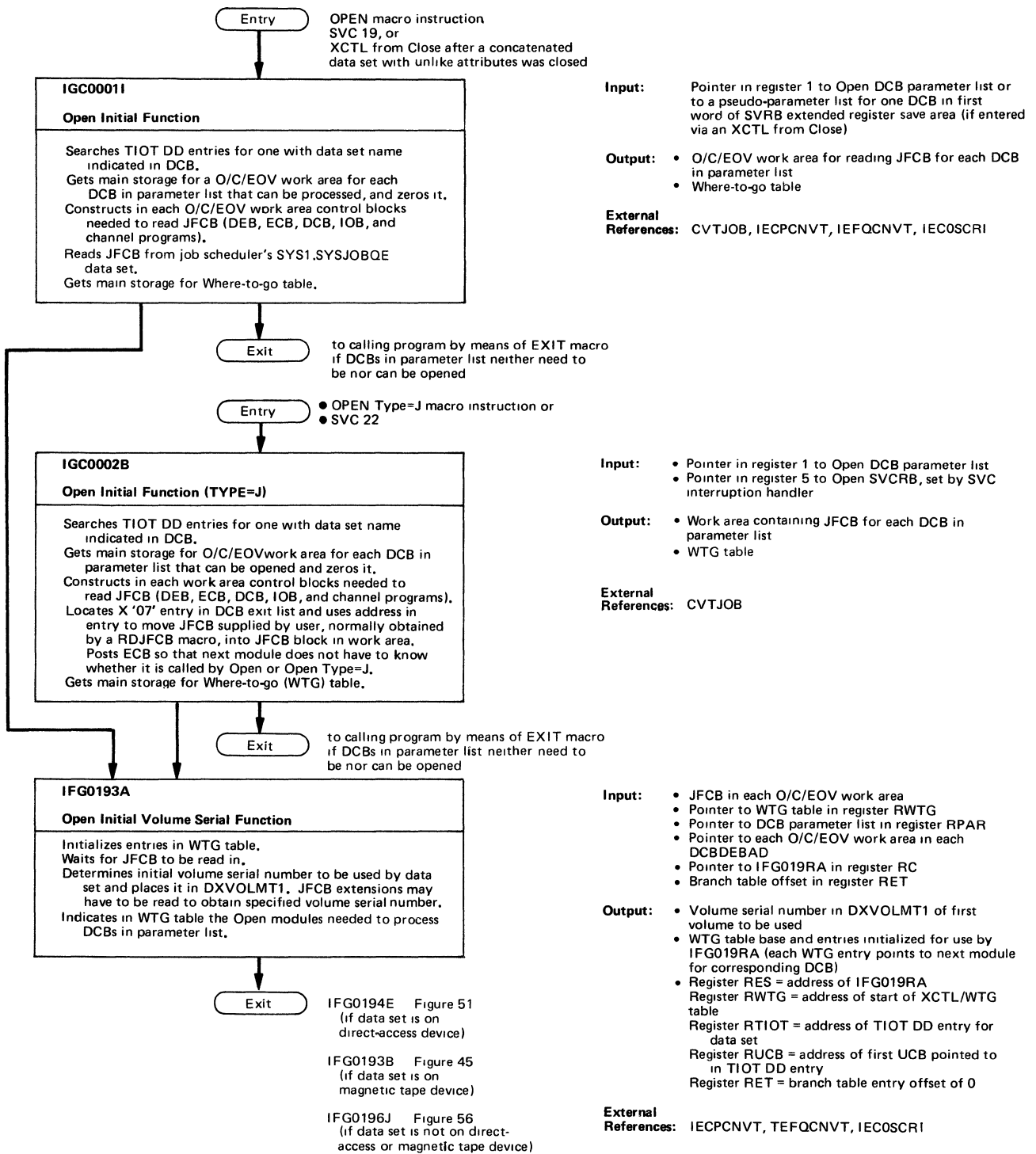


Figure 44 (Chart 1 of 1). Open and OPENJ Initialization Processing

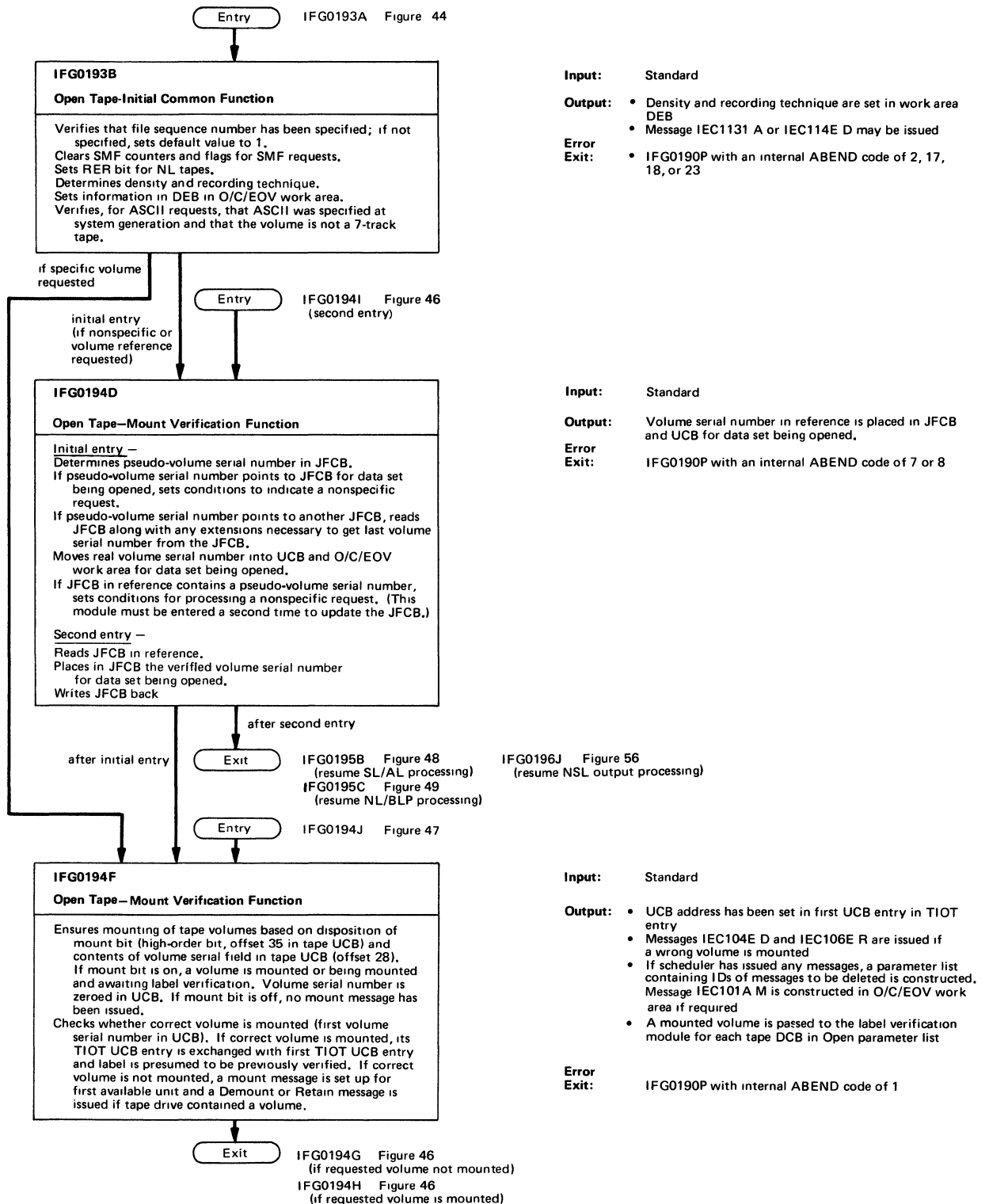


Figure 45 (Chart 1 of 6). Open Tape Processing

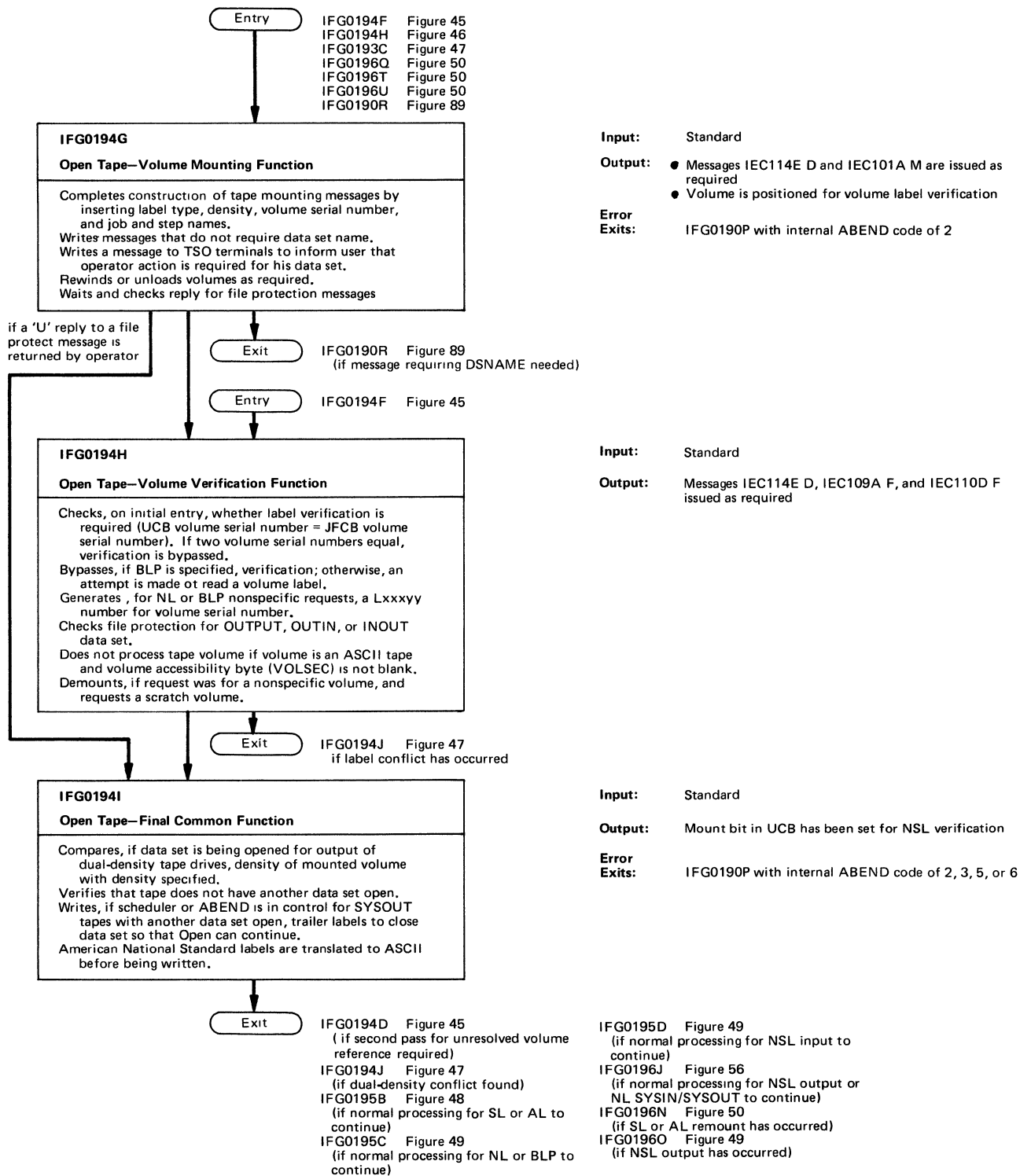


Figure 46 (Chart 2 of 6). Open Tape Processing

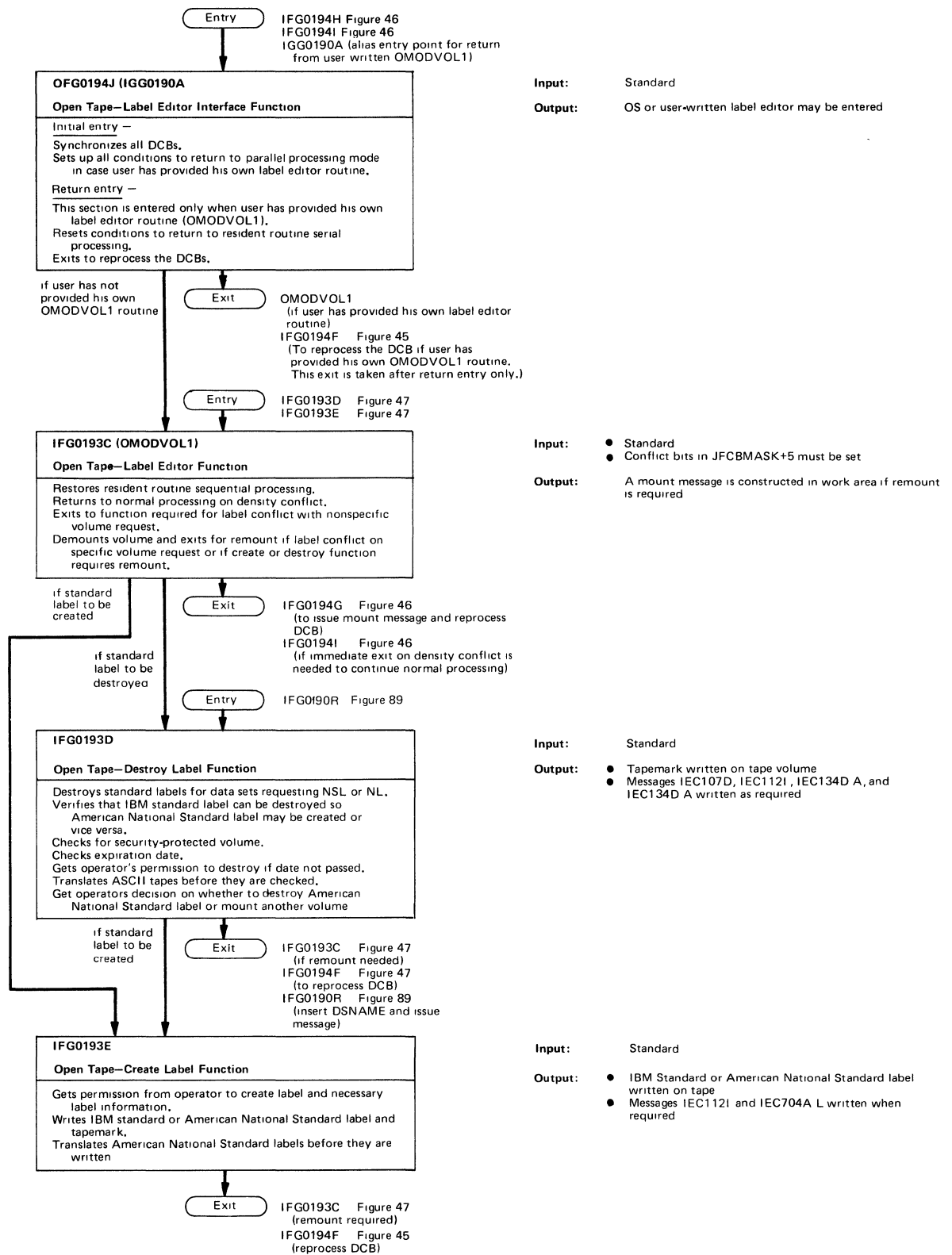


Figure 47 (Chart 3 of 6). Open Tape Processing

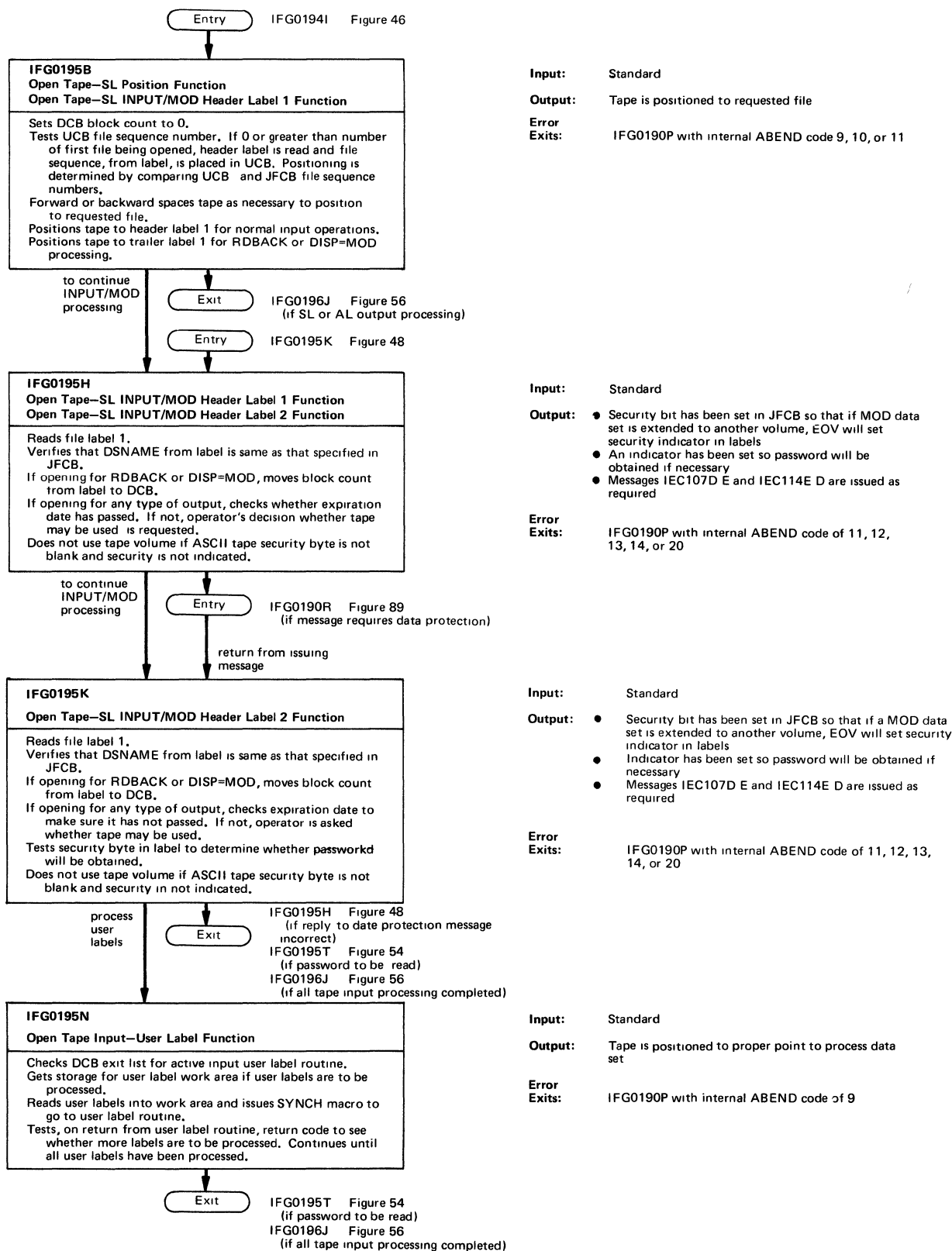


Figure 48 (Chart 4 of 6). Open Tape Processing

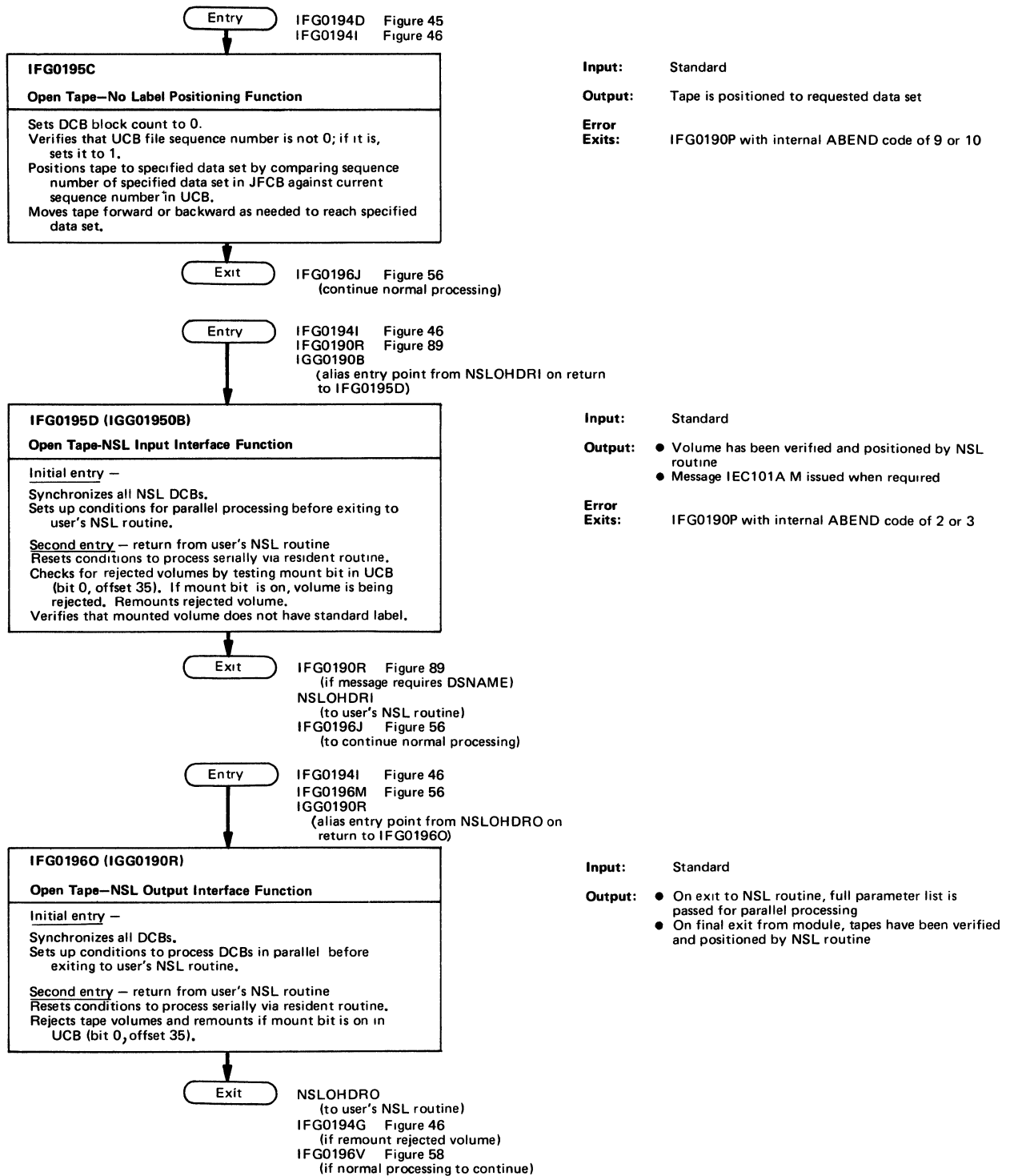


Figure 49 (Chart 5 of 6). Open Tape Processing

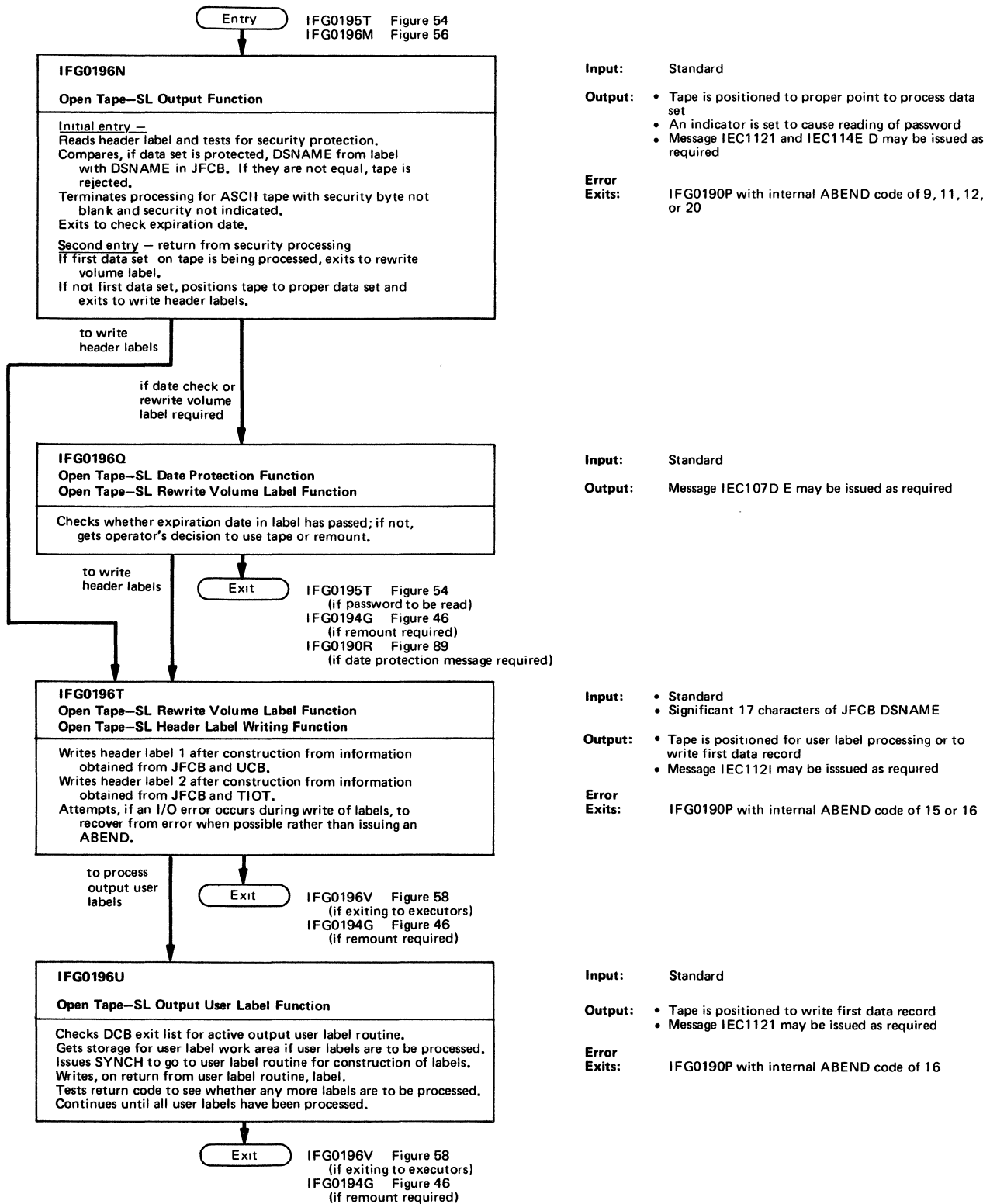


Figure 50 (Chart 6 of 6). Open Tape Processing

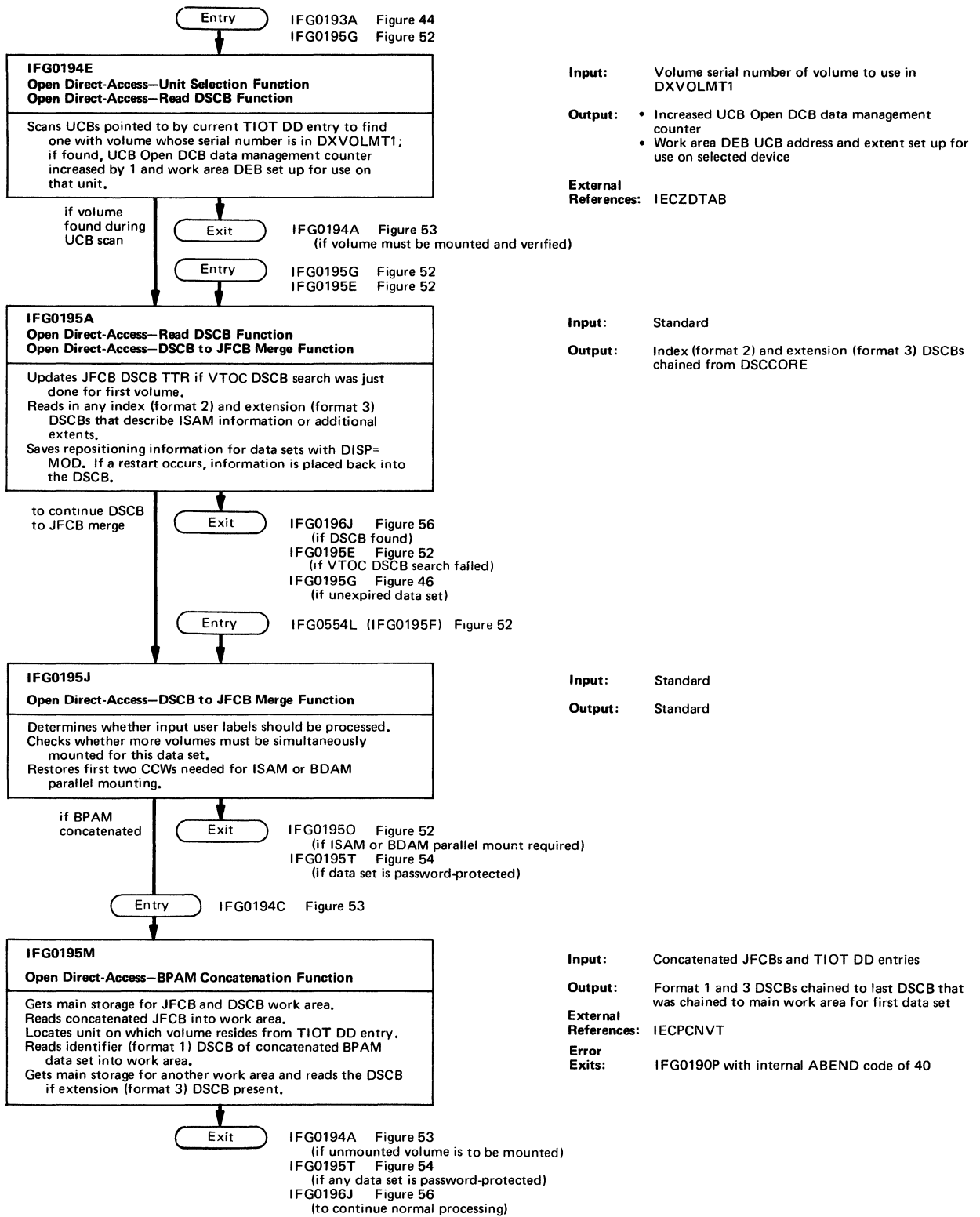
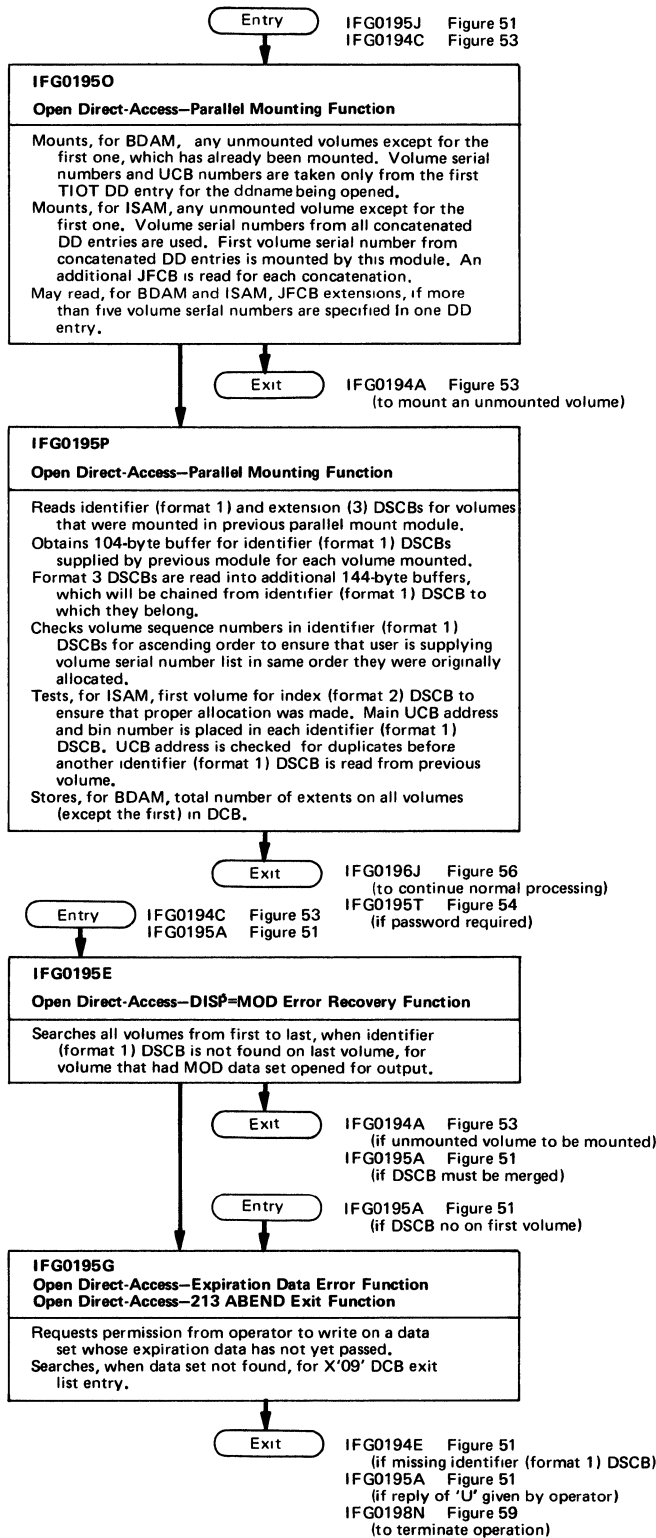


Figure 51 (Chart 1 of 3). Open Direct-Access Processing



Input:

- Register RTIOT - pointer to TIOT DD entry (Must never point to a concatenated entry because RTIOT will be restored later to first TIOT DD entry for data set)
- Work area - DSCORE - pointer to any chained index (format 2) and/or extension (format 3) DSCB already read by first volume basic mounting function

Output:

- All required volumes mounted, verified, and their UCB Open DCB data management counters increased by 1 to lock them on their units
- 104-byte additional work areas, chained to last DSCB, to be used to read identifier (format 1) DSCBs of additional volumes

External References: IECPCNVT, IECZDTAB

Input: Chained identifier (format 1) DSCB work areas

Output:

- Identifier (format 1) DSCBs read into identifier (format 1) DSCB work areas
- Any extension (format 3) DSCBs chained from identifier (format 1) DSCBs to which they belong

External References: IECPCNVT

Input:

- List of volume serial numbers for data set, either already in main storage in JFCB or to be read in from JFCB extensions
- DXCCW1 and DXCCW2 must contain search ID EQ, TIC +8
- RUCB contains address of current UCB on which full VTOC search failed

Output:

- TIOT UCB pointer rotated to correct volume
- Work DEB set up with address of correct volume
- DSCB from correct volume in main work area

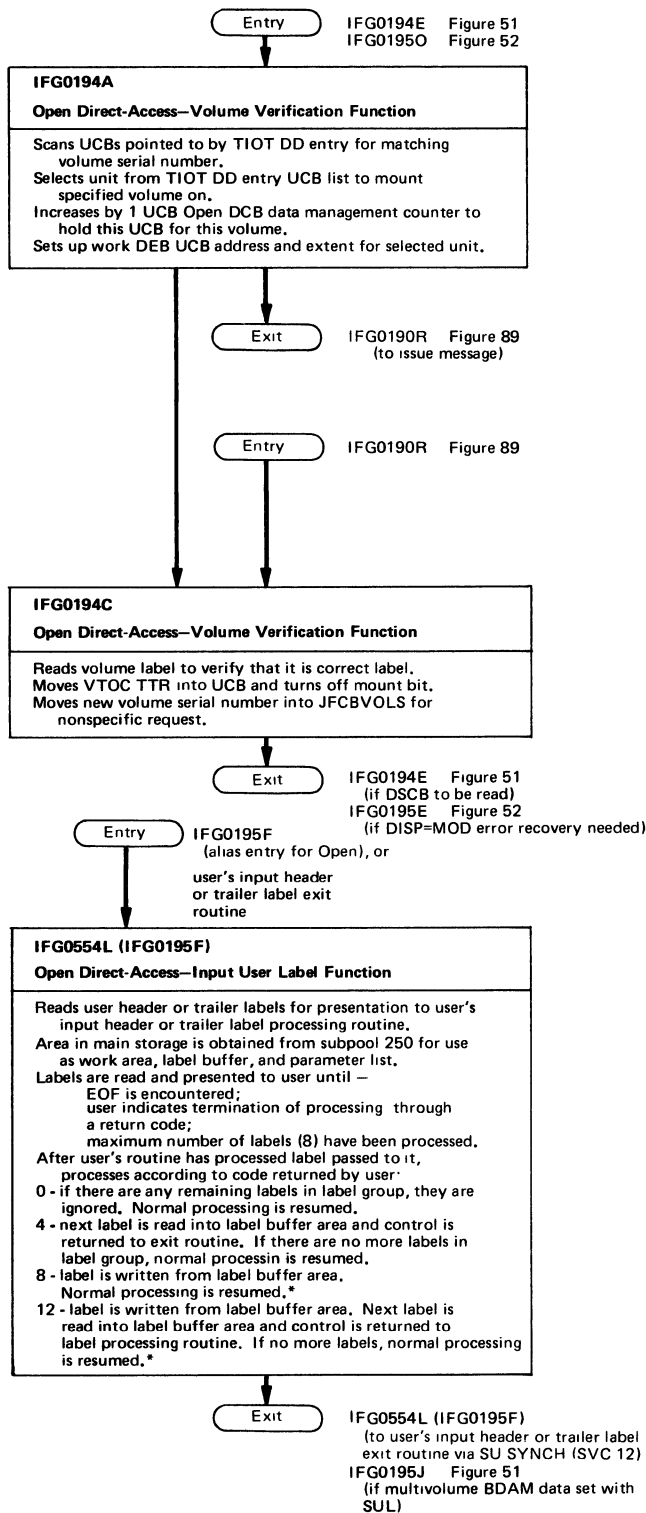
Input:

- Register RET=0,
- Reply of 'U' to use data set, or reply of 'M' not to use data set

Output: Message IEC107D E is issued

Error Exits: IFG0190P with internal ABEND code of 28

Figure 52 (Chart 2 of 3). Open Direct-Access Processing



Input:

- Registers —
 - RTIOT — current TIOT DD entry (May be each of several entries for concatenated BPAM or ISAM data sets)
 - RET — entry indicator (If 4 or 8, work DEB must already be set up for unit selected for volume)
- Work areas:
 - DXLBL (96 bytes) — must be available for writing messages and reading volume label
 - DXJBF (176) bytes — used for DSNAM and temporary data set indicator in mount message
 - DXCCW2+4 — address in TIOT DD entry UCB list of first UCB pointers that can be used
 - DXCCW8 - DXCCW11 — register save area
 - DXVOLMT1 — contains volume serial number of specified volume

Output:

- Registers
 - RUCB - current or main UCB address
 - RET — set to 4 as return entry indicator
- Work areas:
 - DXLBL — volume label
 - DXCCW2+4 — address in TIOT DD entry UCB list of first UCB pointer that can be used
 - DXCCW1 - DXCCW3 - contain search ID EQ, TIC * -8 channel program
 - DXVOLMT1 — contains volume serial number of specified volume

Messages IEO106E R and IEC101A M may be issued

Input: Same as IFG0194A

Output:

- Same as IFG0194A
- May issue message IEC111E D

IFG0195M Figure 51
 (if BPAM concatenation)
 IFG0195O Figure 52
 (if parallel mounting required)

Input:

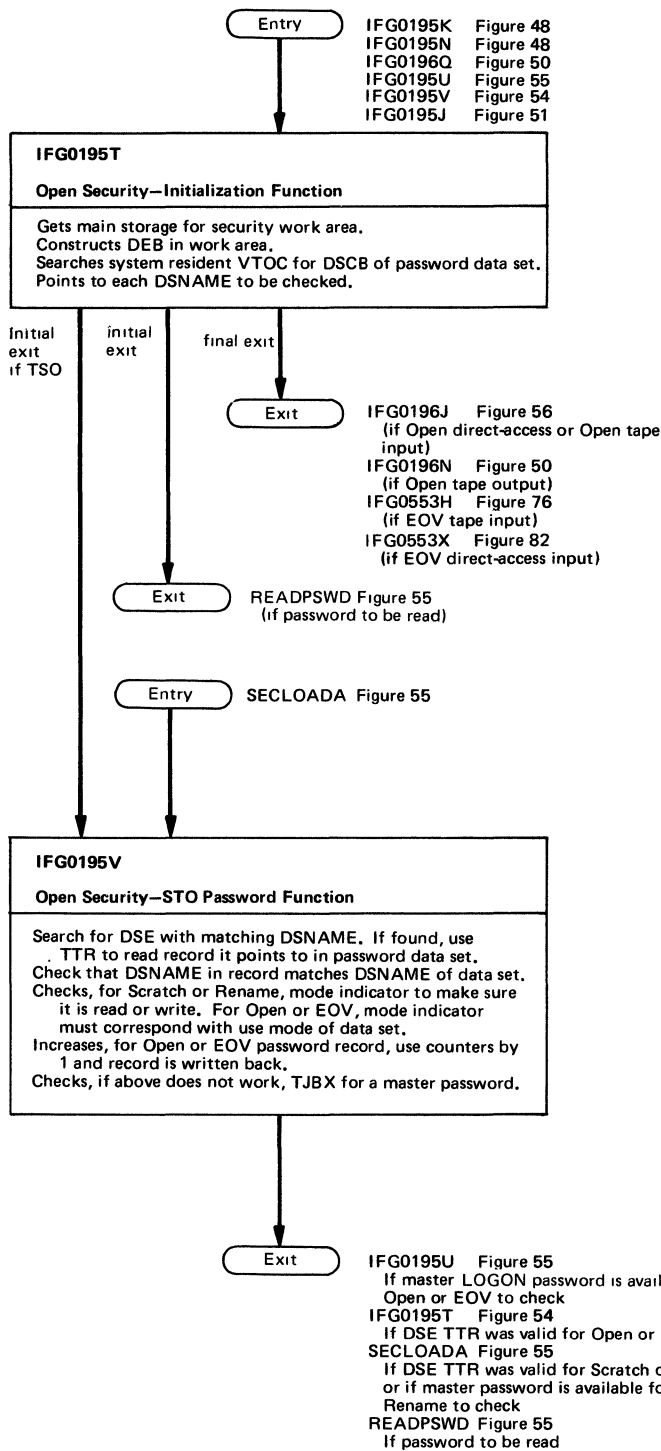
- Register RC - contents of fullword or user's DCB exit list entry being processed
- Work areas:
 - DSCEXTYP — extent type and size of first extent in format 1 DSCB
 - DXDAADDR — MBB must already be set up for extent 0 and bin number for this volume
 - DXDEBMOD - DXDEBNTR — must be set up to cover whole volume
 - DXVOLMT — used as save area for CCHHR of format 1 DSCB from DXCCW7
- User labels — existing user header or trailer labels are read from one track user label extent for this data set

Output:

- User labels — if data set opened for UPDAT user labels may be modified and written back at option of user's exit routine
- Registers and work areas same as for input

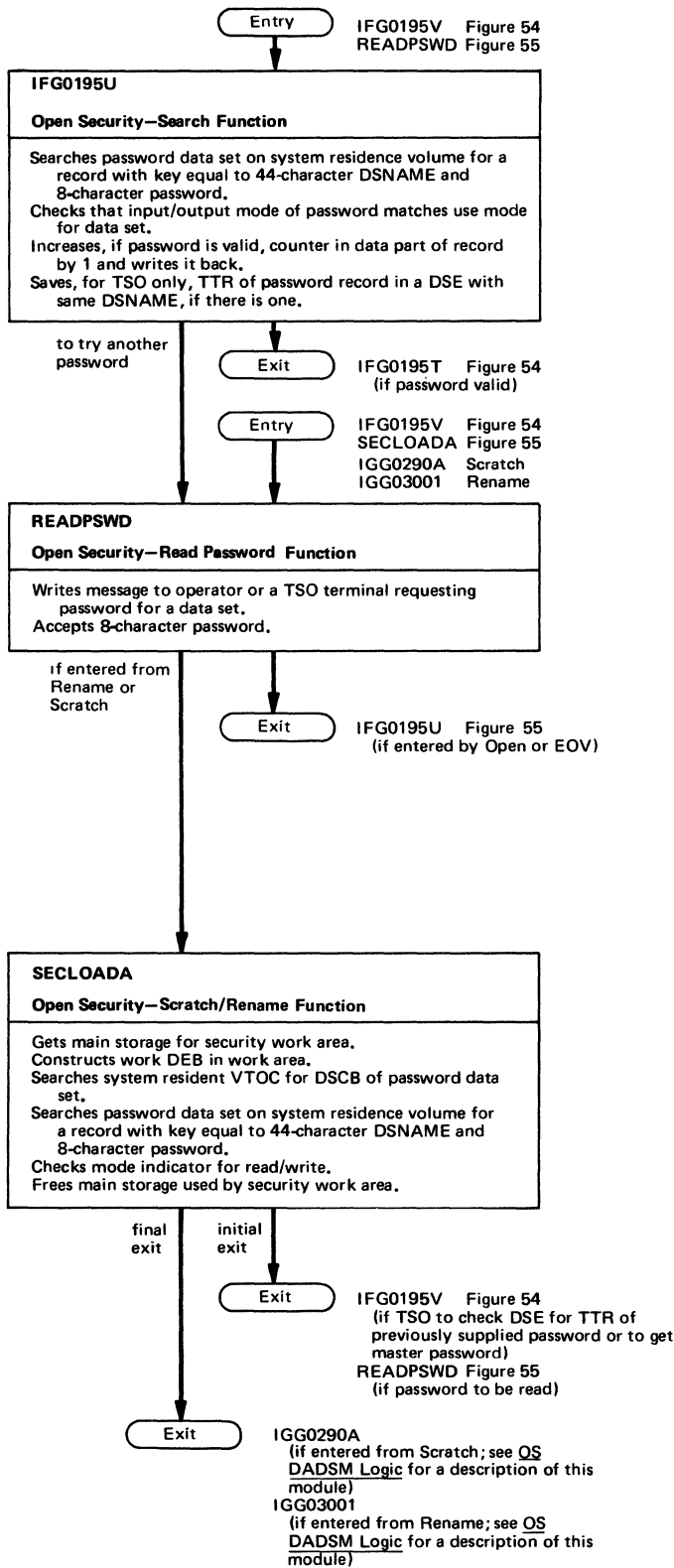
*Only for physical sequential data set opened for UPDAT (UHL, UTL), physical sequential data set opened for OUTPUT or OUTIN with DISP=MOD (UTL), or direct data set opened for UPDAT or OUTPUT (UHL, UTL)

Figure 53 (Chart 3 of 3). Open Direct-Access Processing



- Input:**
- Registers:
 - RDCB — DCB address
 - RCORE — main work area address
 - RES — resident routine address
 - RWTG — address of WTG table (inside main work area if entered by EOVS)
 - RPARC — address of current entry in DCB parameter list.
 - RWTGC — address of current entry in WTG table
 - RTIOT — address of TIOT DD entry
 - RET — 0 = branch table initial entry code
4 = parallel DCB entry
8 = successful return from IFG0195U or IFG0195V
 - Main work area:
 - RXCCW8 - 11 - register save area (contents of registers 9-14, 0-1 saved by resident routine); BPAM identifier (format 1) DSCBs, if any, chained to DSCCORE
 - WTG table:
 - WTGMODNM — must be IFG0195T every time this module is entered
 - WTGIDTTR — ID/TTR (TTR may be 0)
- Output:**
- Registers:
 - RBASE — base register of IFG0195T
 - RUCB — first UCB address from TIOT DD entry
 - RET — branch table initial entry code for next module:
0 for IFG0196J
0 for IFG0196N
0 for IFG0553X
16 for IFG0553H
 - WTG table:
 - WTGMODNM — IFG0195T for Open or IFG0555T for EOVS
 - WTGIDTTR — ID/TTR of next module (TTR is 0 if entered by EOVS)
- Input:**
- Registers:
 - RCORE — pointer to DXCORE work area if Open or EOVS
— pointer to DXCCW1 work area if Scratch or Rename
 - RES — pointer to resident routine if Open or EOVS
 - R5 — pointer to work IOB if Scratch or Rename
 - RWTG — WTG table address if Open or EOVS
 - RPARC — pointer to DCB parameter entry if Open or EOVS
 - R7 — pointer to work ECB if Scratch or Rename
 - RB — pointer to 44 character DSNAME
 - RD — pointer to security work area
 - Security work area DEB already chained to work DCB
 - Address of DXCCW1 already in IOB start address
- Output:**
- Exit code in MYCODE2
 - Master password in MYREPLY if exit code = 2
- Error Exit:** IFG0195U for Open or EOVS if I/O error during updating of valid password record

Figure 54 (Chart 1 of 2). Open Security Processing



Input:

- 8-byte password in MYREPLY
- 44-character DSNNAME pointed to by Register RB

Output: Messages IEC1171 and IEC1181 may be issued for TSO tasks only

Input:

- MYCODE1 indicates initial entry:
 - 00 from Open
 - 01 from EOVS
 - 02 from Rename
 - 03 from Scratch
- MYCODE2 indicates retry count actions:
 - 00 initial entry
 - 01 TSO DSE TTR not present or not valid and master password not available; get password from terminal
 - 02 TSO with master password failed; get password from terminal
 - 03 first password supplied was wrong, one more try
 - 04 second password supplied was wrong
 - 05 TSO DSE TTR was valid
 - 08 terminal supplied password longer than 8 characters
 - 09 I/O error while updating record read by TSO DSE TTR

Output: Message IEC301A may be written to operator or messages IEC113A, IEC115I, IEC116A, or IEC1171 may be written to TSO terminal.

Input:

- Register RUCB:
 - X'02000000' for initial entry from Rename
 - X'03000000' for initial entry from Scratch
 - 0 on all reentries
- MYCODE2:
 - 00 initial entry
 - 01 TSO to READPSWD for first try
 - 02 TSO master password in MYREPLY to be checked
 - 03 ordinary or TSO password in MYREPLY to be checked
 - 04 ordinary or TSO password in MYREPLY to be checked
 - 05 TSO DSE TTR password was valid
 - 08 TSO terminal supplied password longer than 8 characters
 - 09 TSO wrong mode error

Output:

- Message IEC1171 may be issued as required
- Register RUCB:
 - X'00000000' for successful password
 - X'01000000' for unsuccessful password
 - X'02000000' for no password data set on system residence volume

Figure 55 (Chart 2 of 2). Open Security Processing

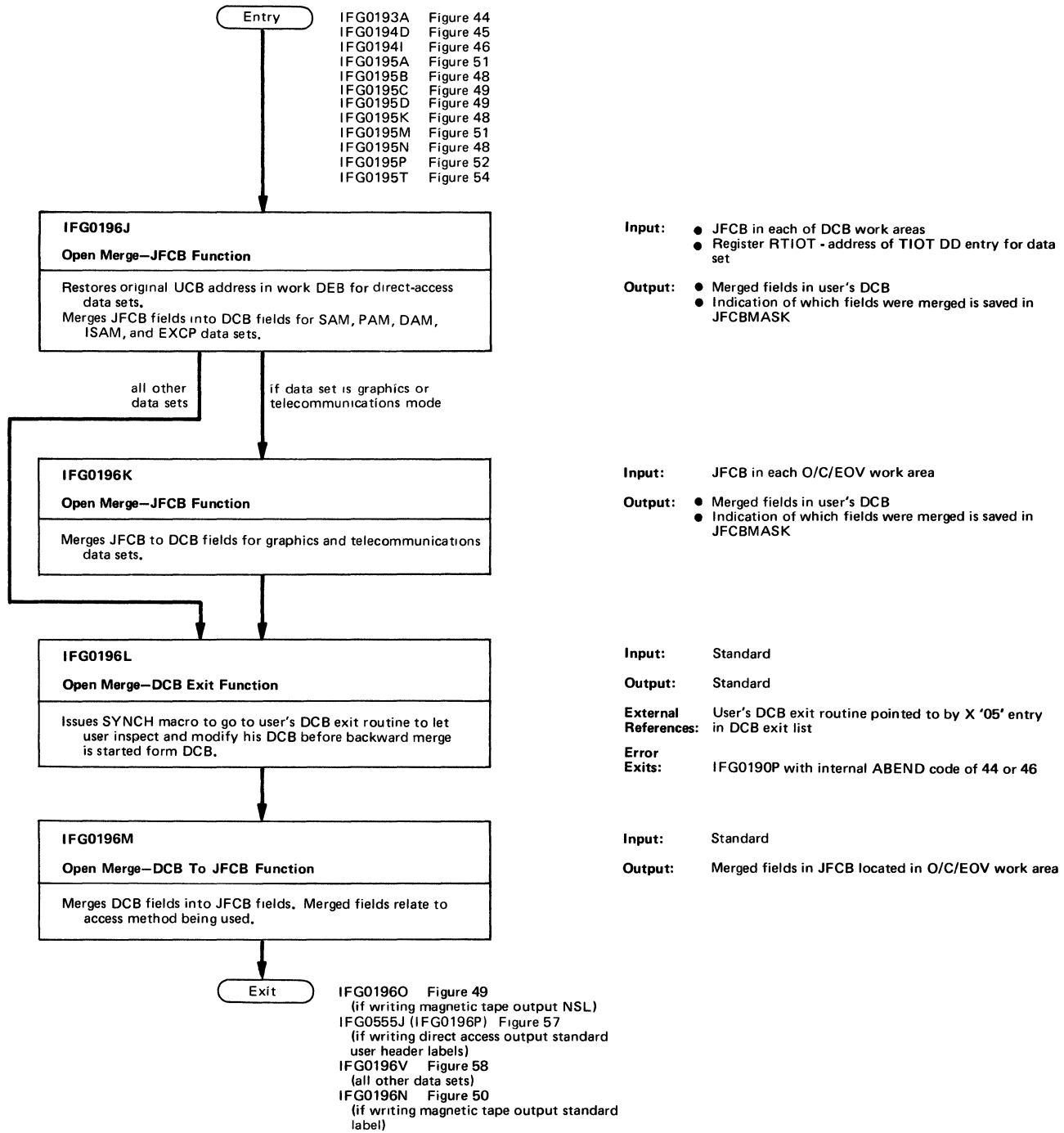


Figure 56 (Chart 1 of 2). Open Merge Processing

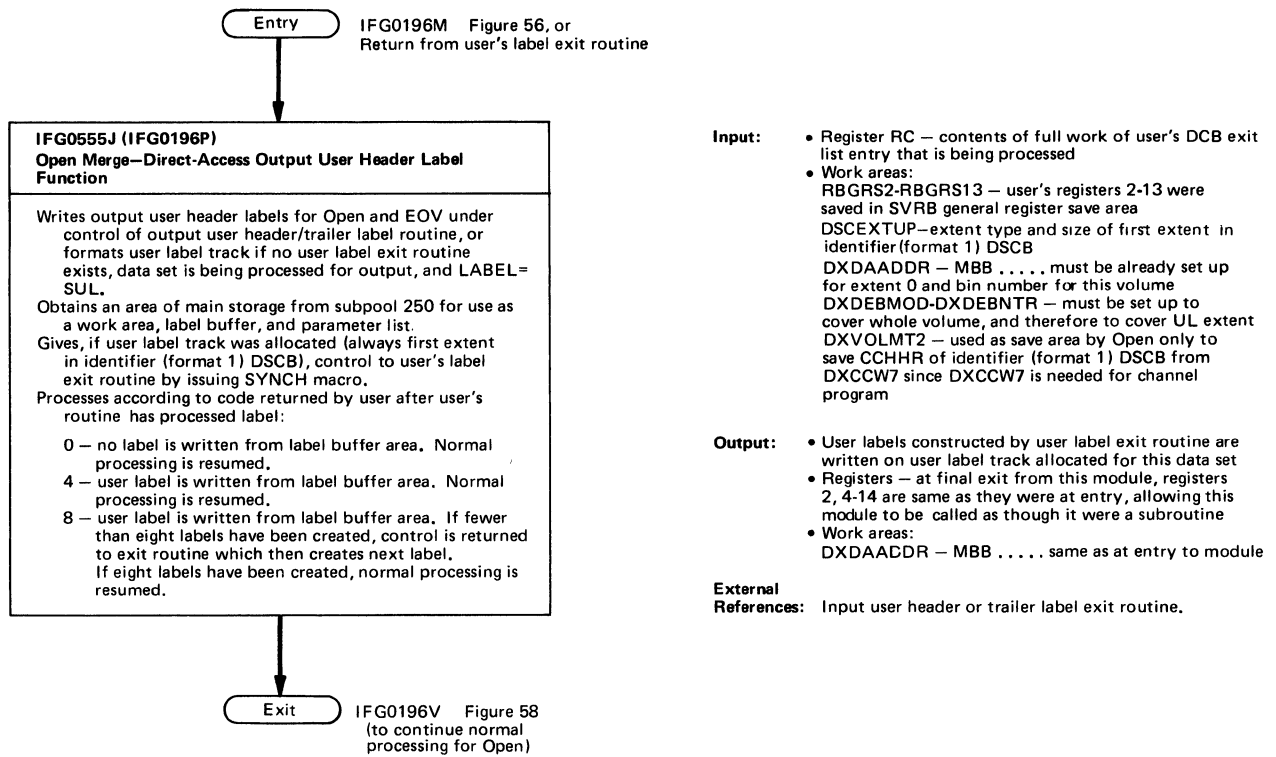


Figure 57 (Chart 2 of 2). Open Merge Processing

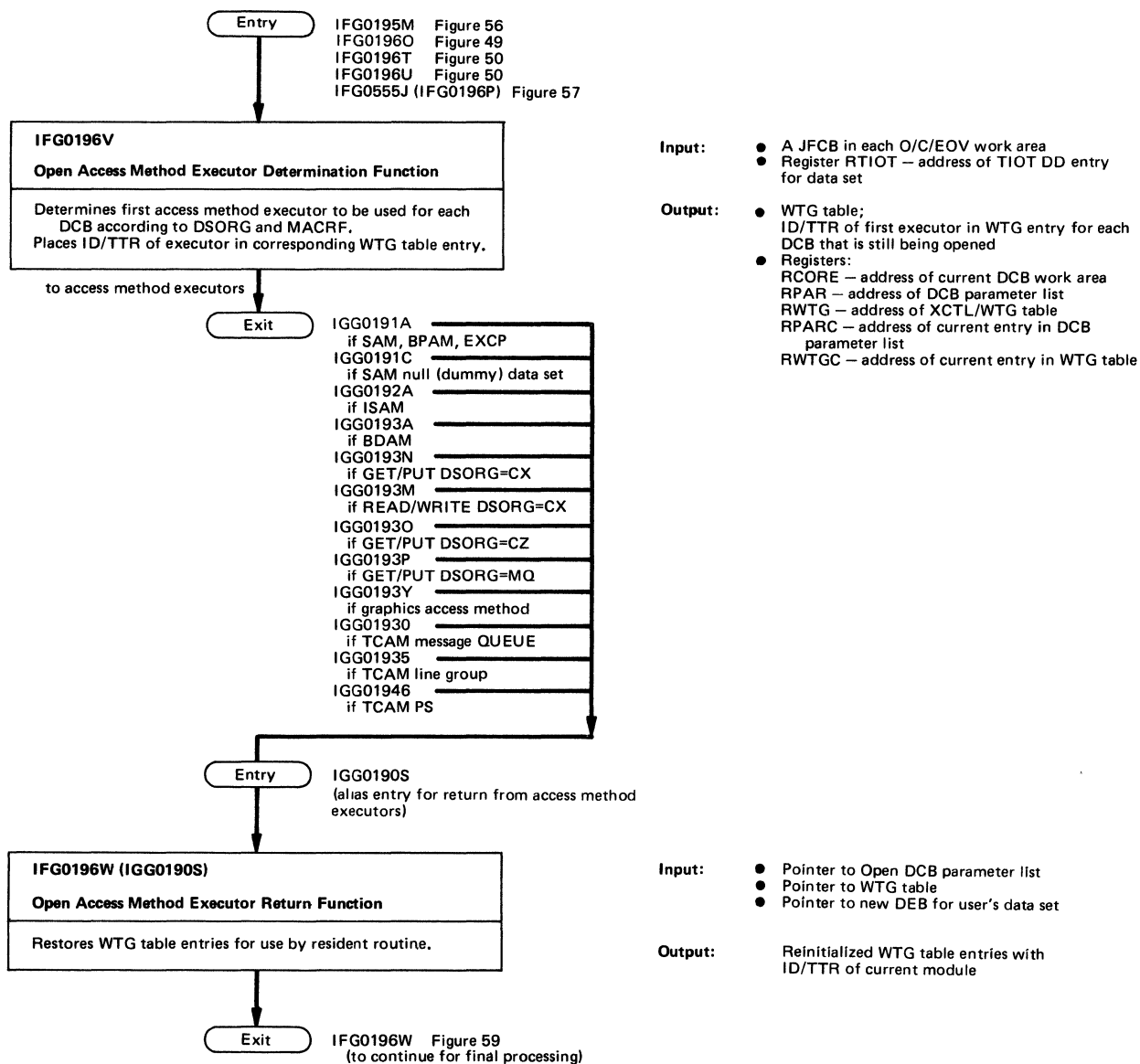


Figure 58. Open Access Method Executor Processing

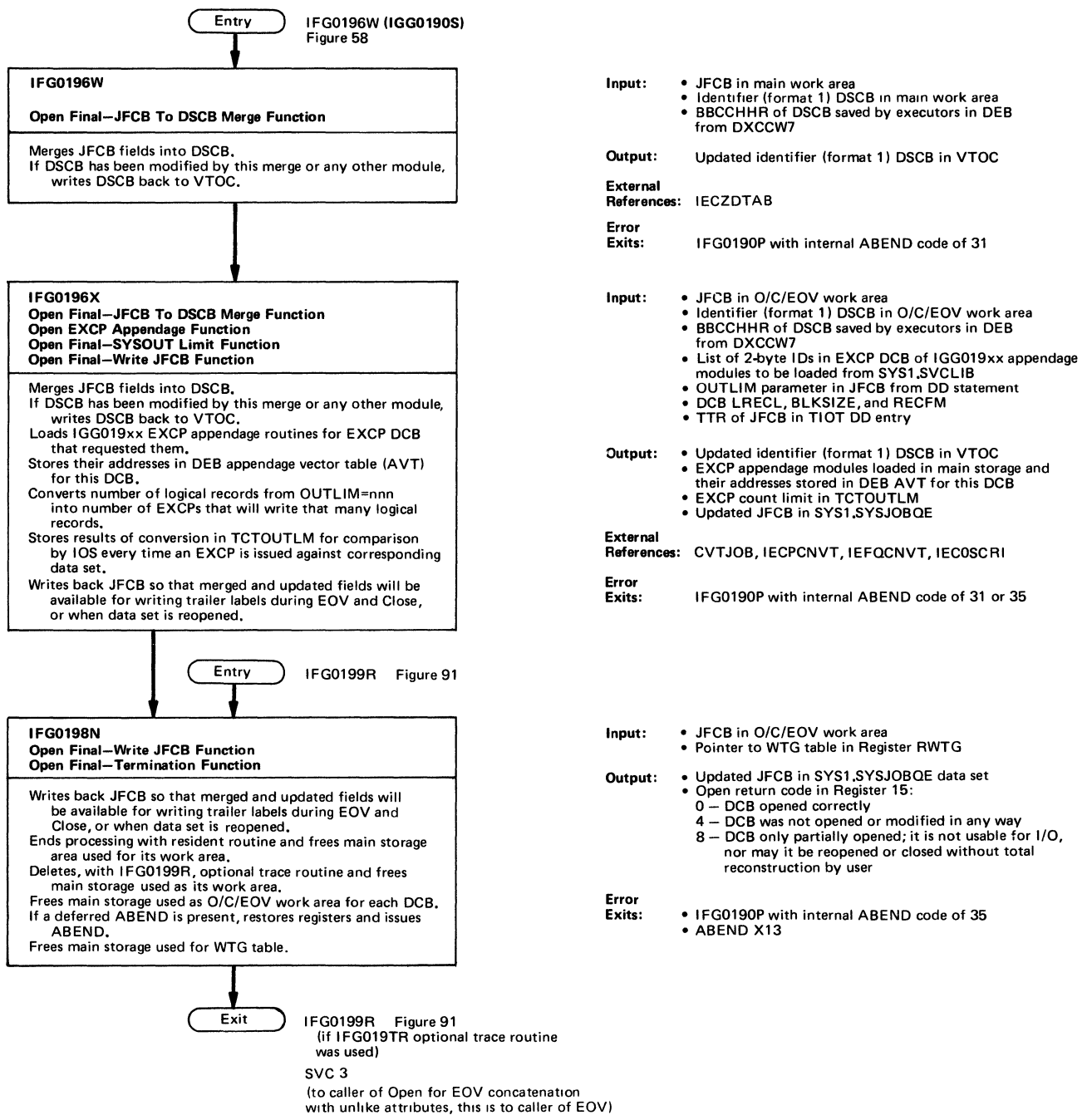


Figure 59. Open Final Processing

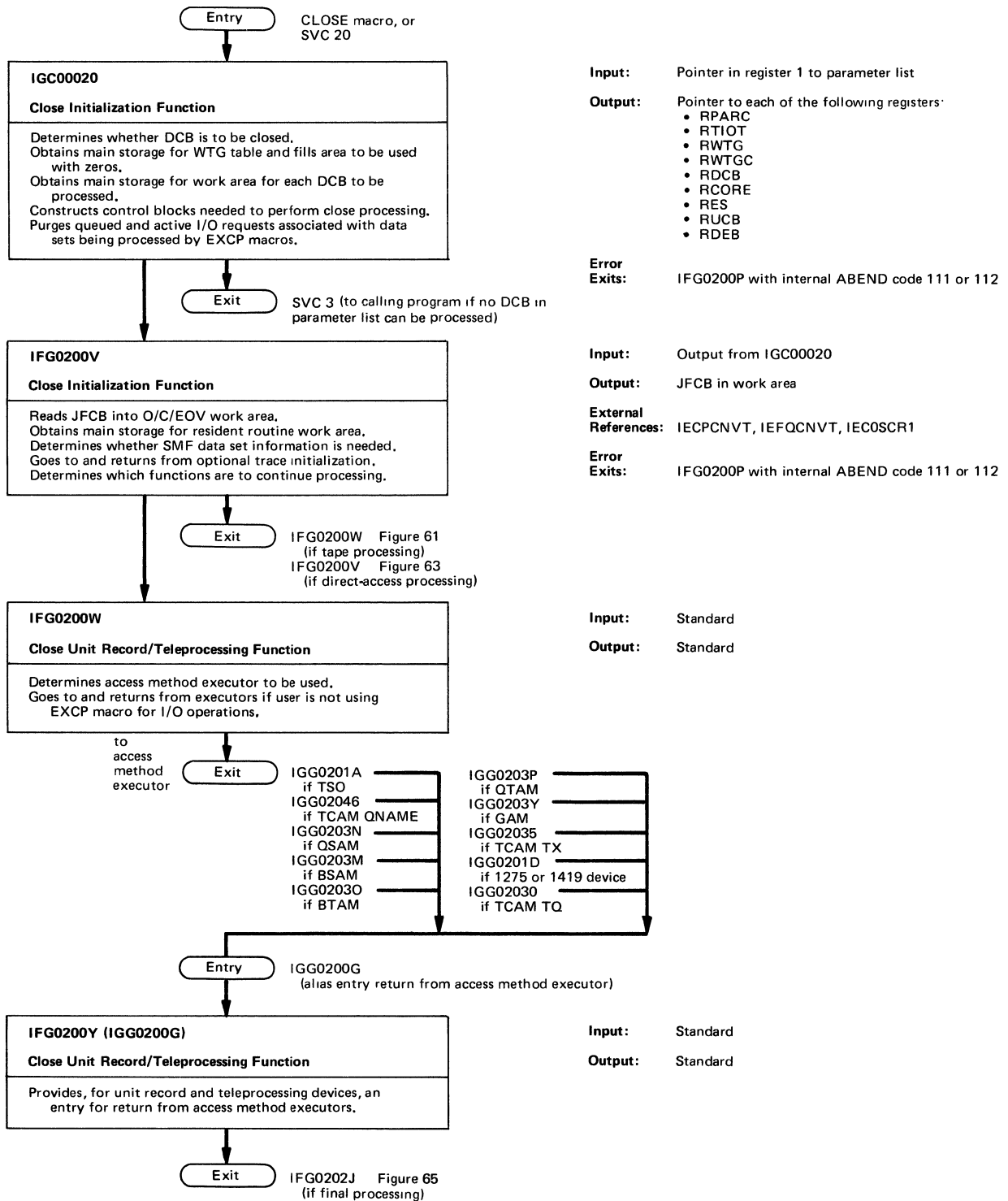


Figure 60. Close Initialization Processing

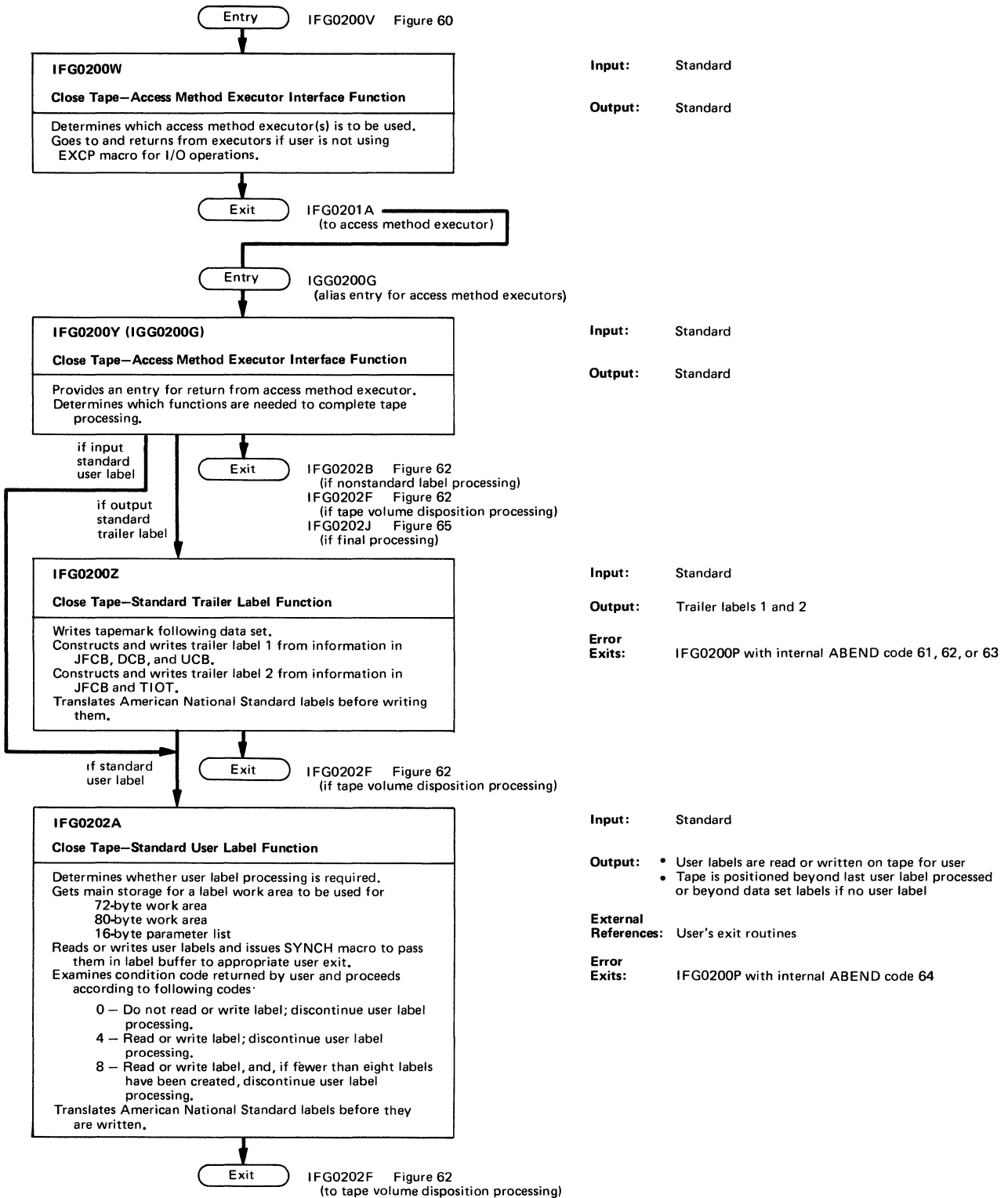


Figure 61. Close Tape Processing Chart

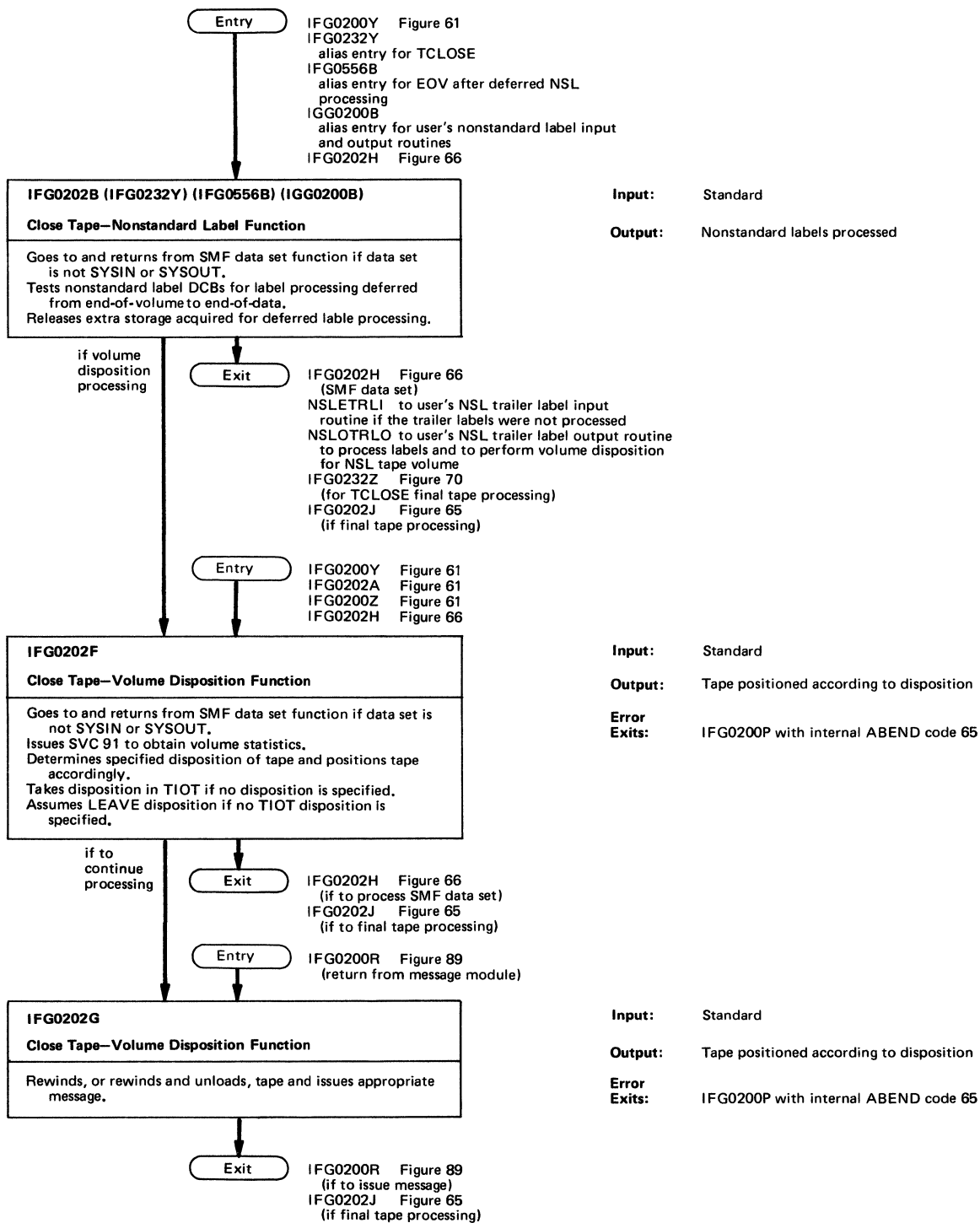


Figure 62 (Chart 2 of 2). Close Tape Processing

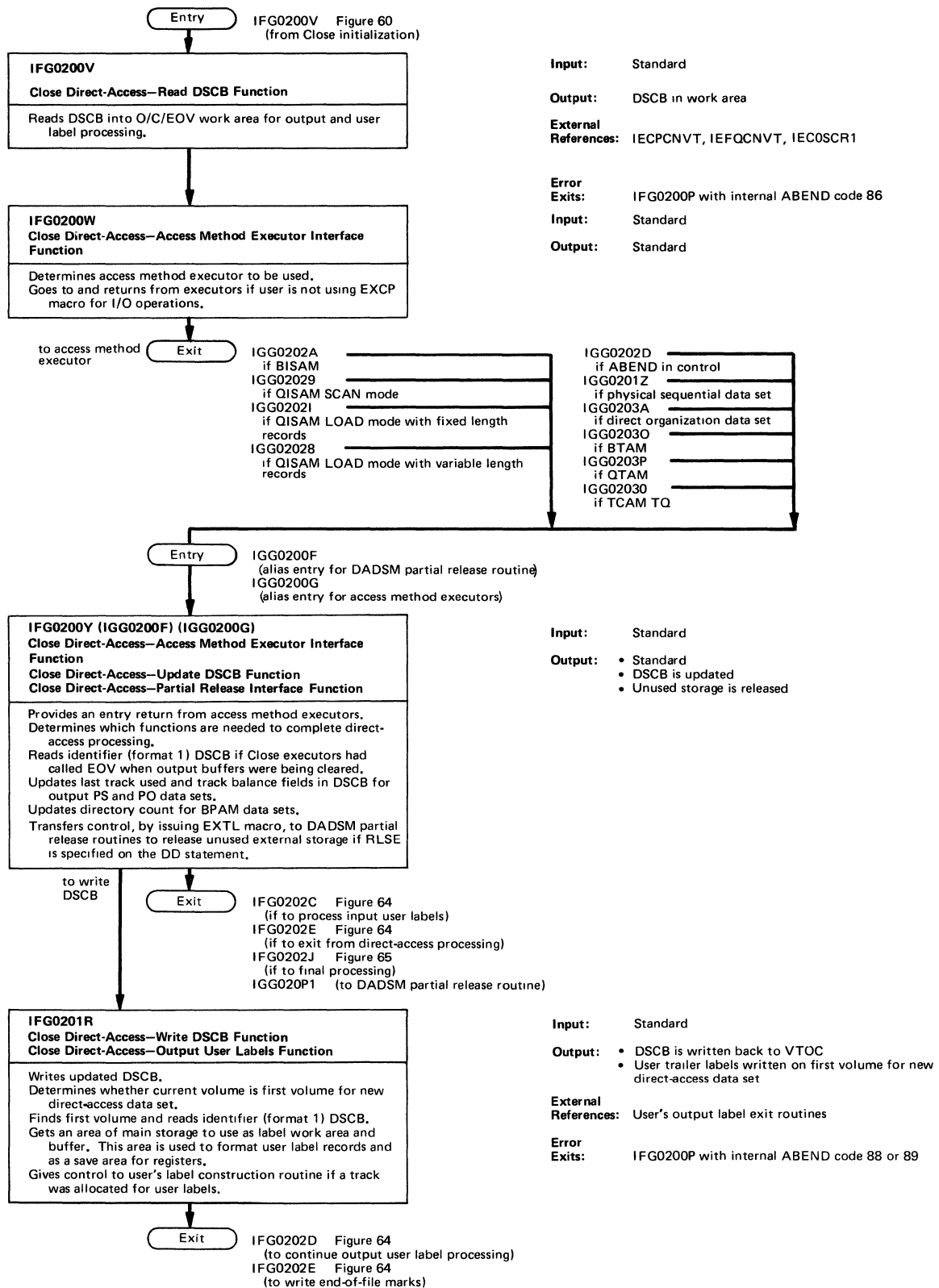
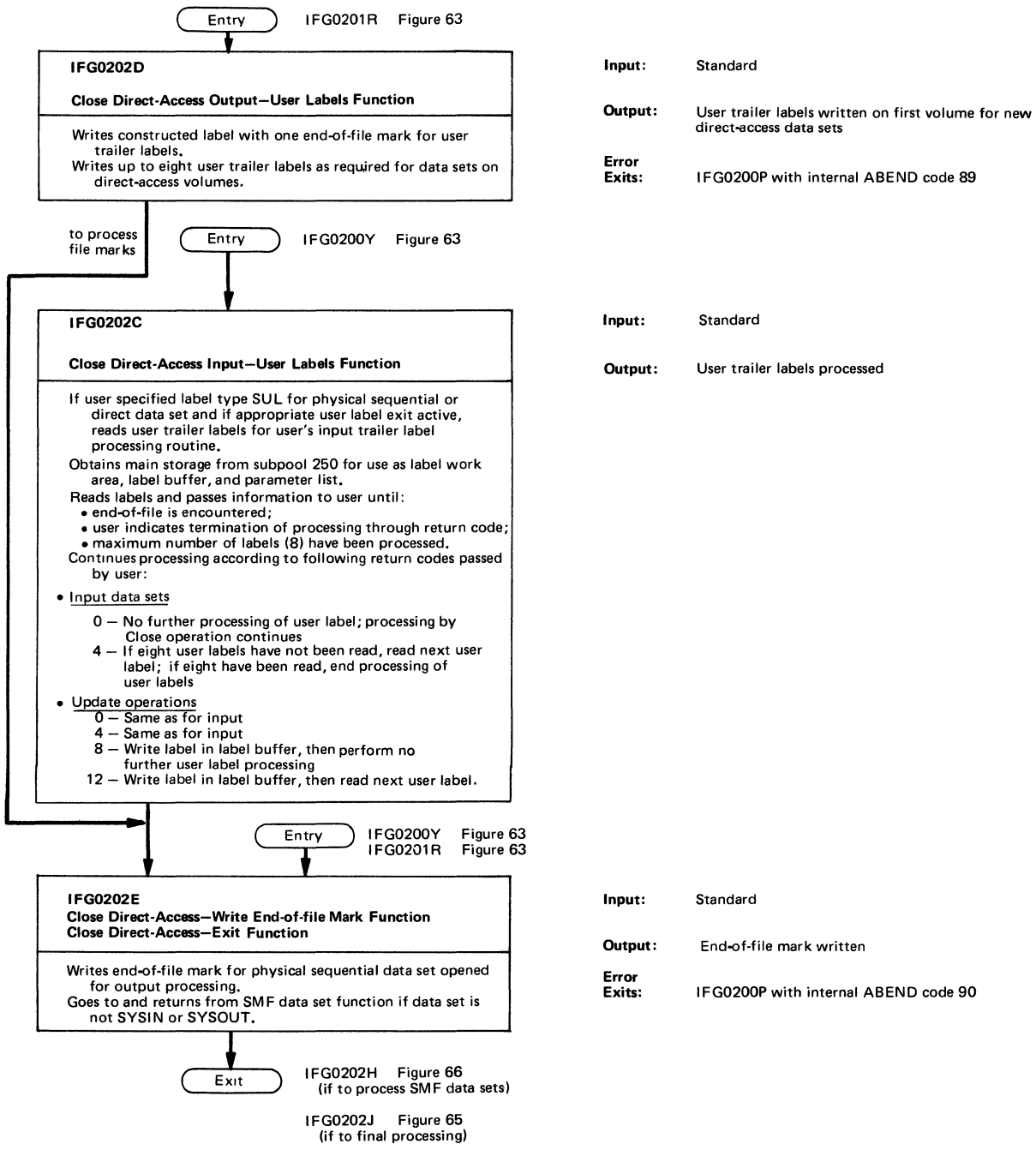


Figure 63 (Chart 1 of 2). Close Direct-Access Processing



Input: Standard
Output: User trailer labels written on first volume for new direct-access data sets
Error Exits: IFG0200P with internal ABEND code 89

Input: Standard
Output: User trailer labels processed

Input: Standard
Output: End-of-file mark written
Error Exits: IFG0200P with internal ABEND code 90

Figure 64 (Chart 2 of 2). Close Direct-Access Processing

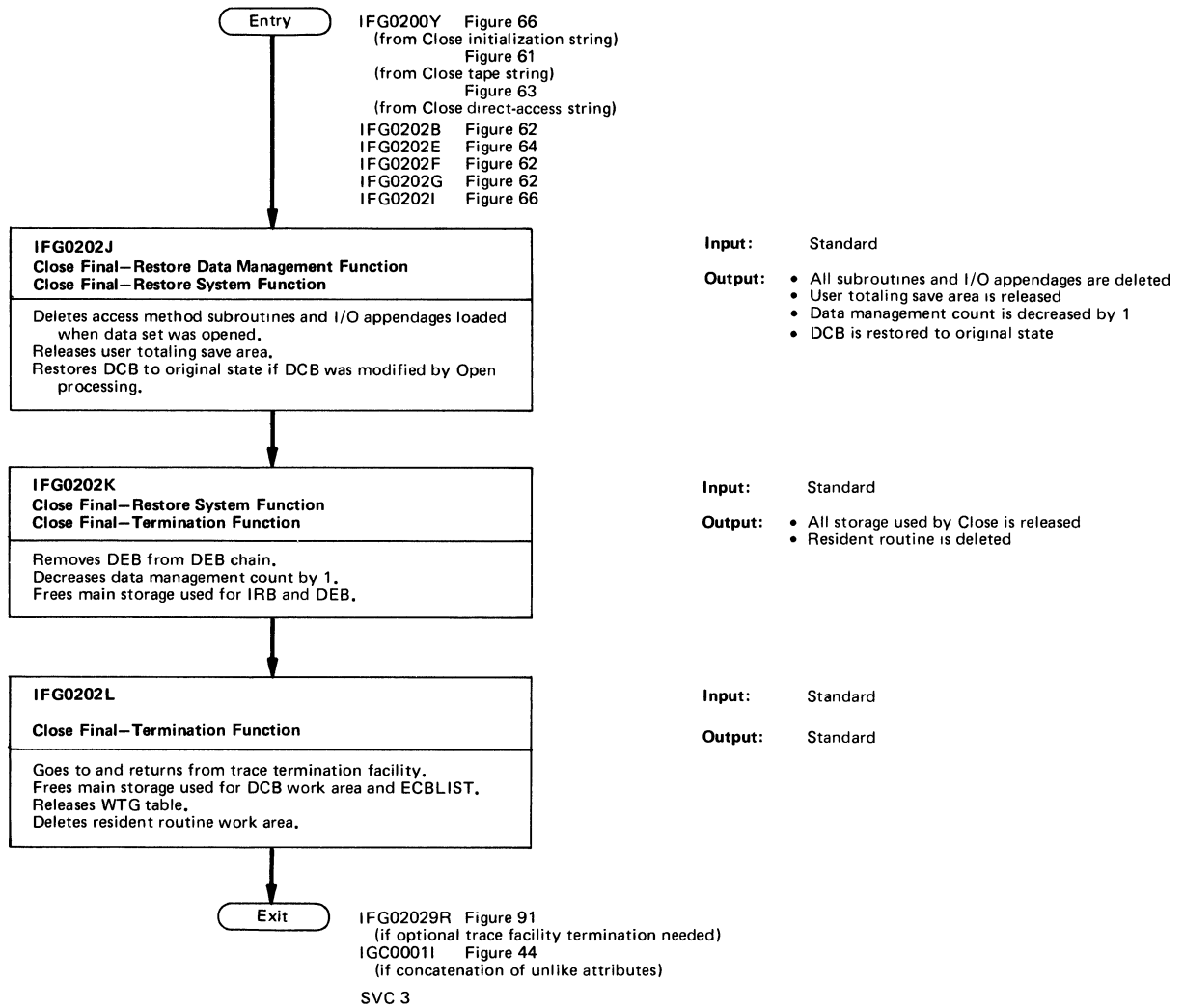


Figure 65. Close Final Processing

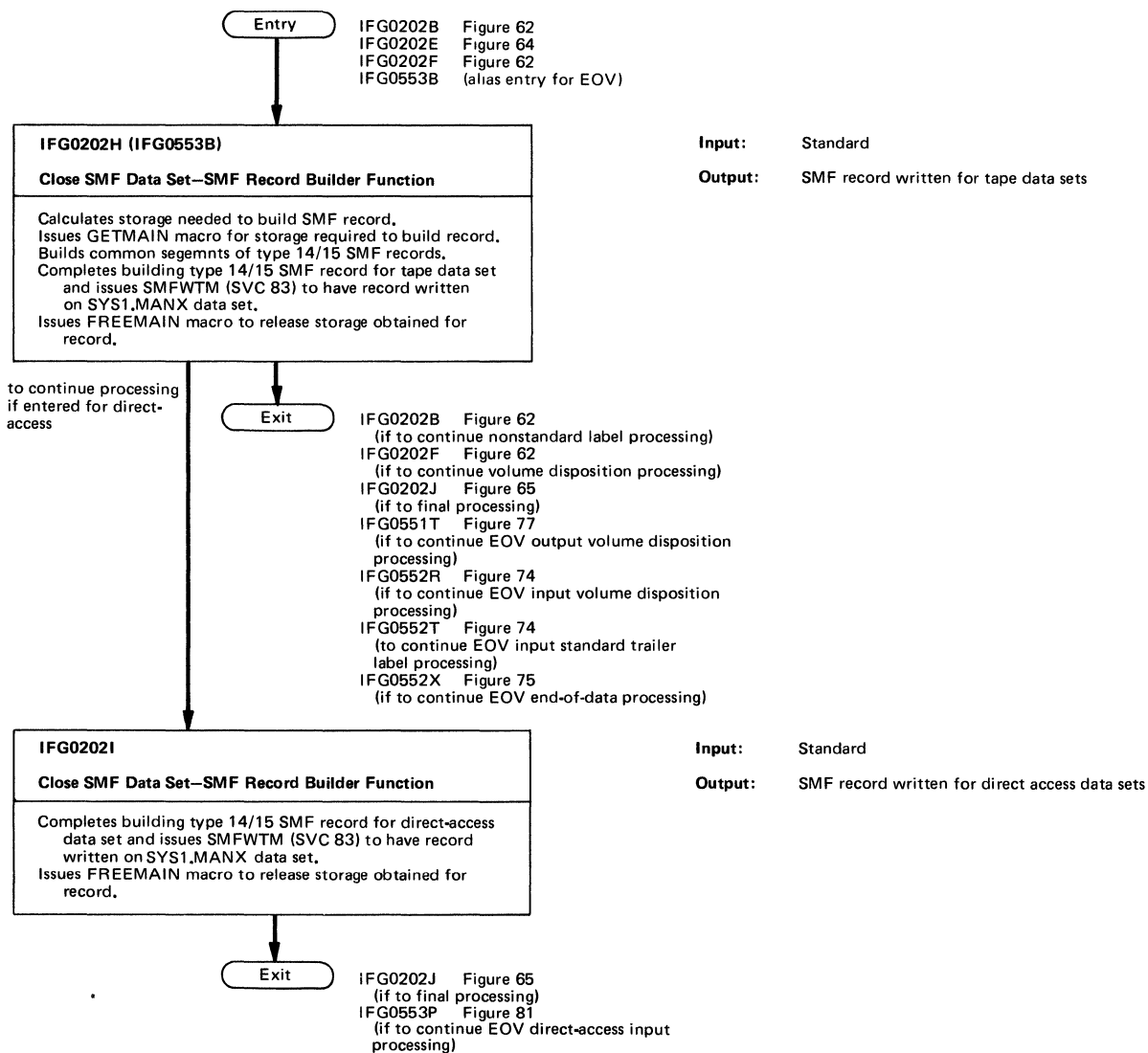


Figure 66. SMF Data Set Processing

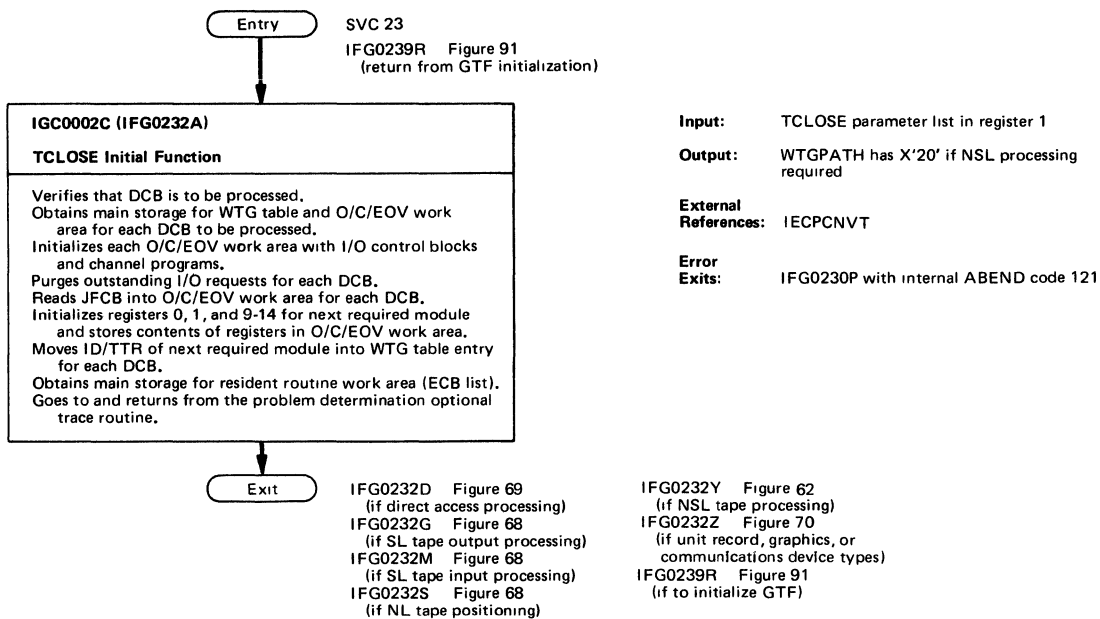


Figure 67. TCLOSE Initialization Processing

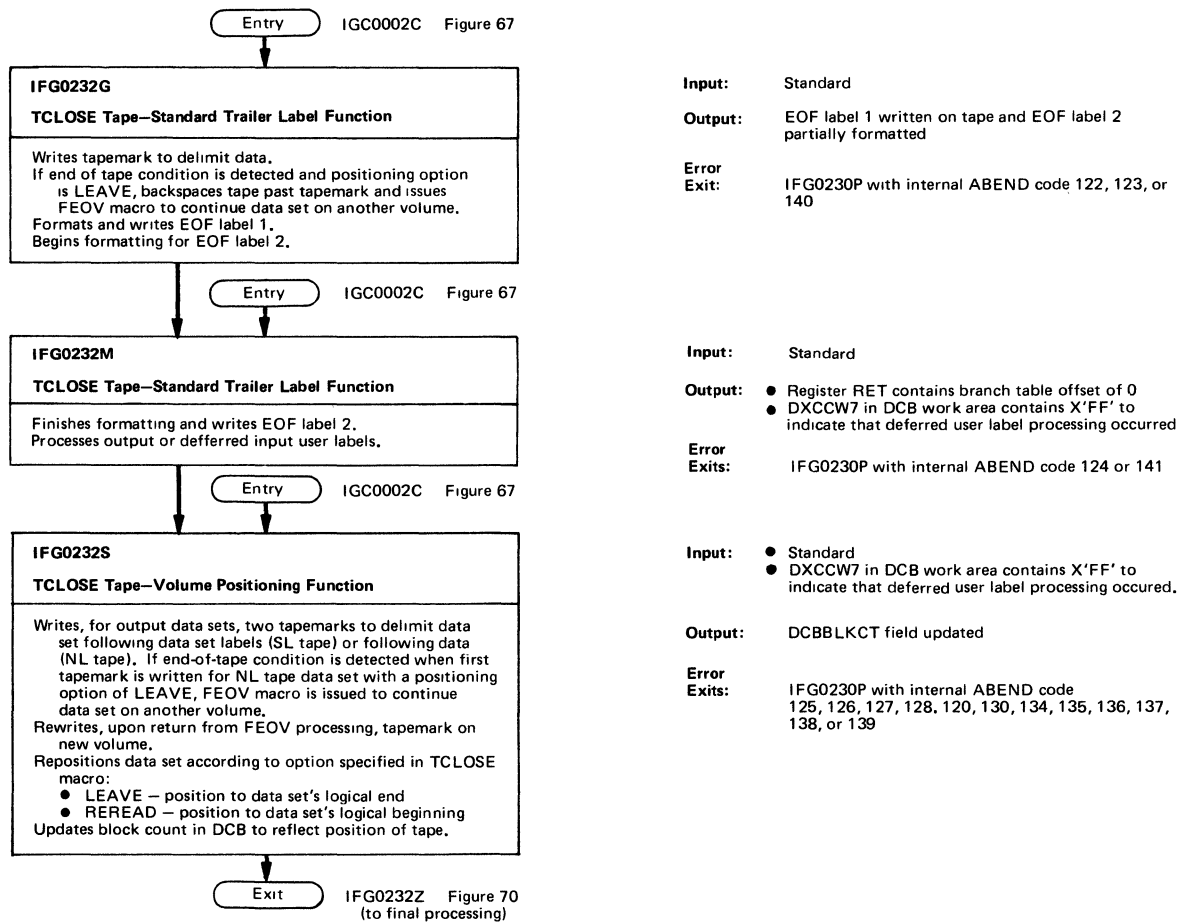
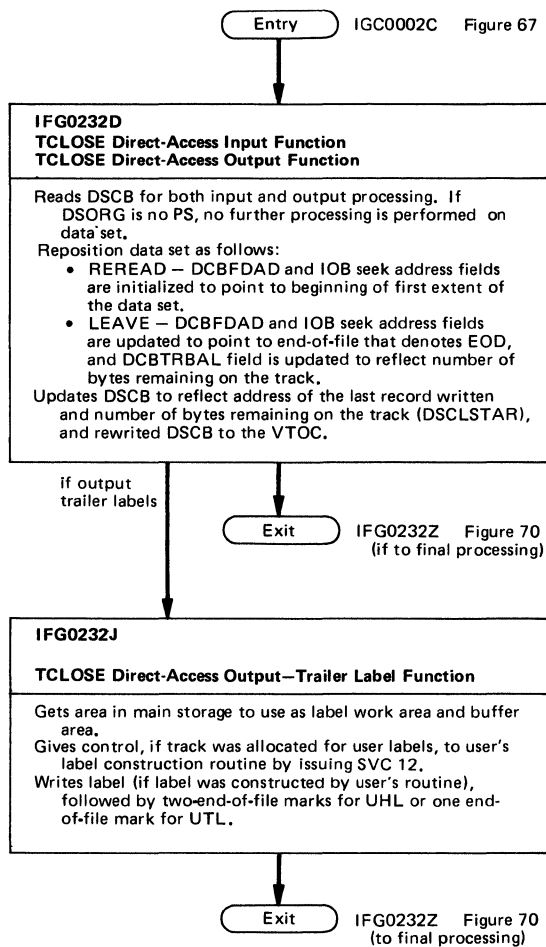


Figure 68 (Chart 1 of 1). TCLOSE Tape Processing



Input: Pointer to parameter list, WTG table, and TIOT.

Output:

- 0 in RET to finish output
- 4 in RET to finish input
- DXCCW in DCB work area contains X'FF' to indicate that data set was not processed.

Error Exits: IFG0230P with internal ABEND code 121, 132, or 133.

Input:

- Register 12 contains exit list entry for exit to be taken: bits 3-7 are exit type code; bits 8-31 are exit address (0s if there is no active exit)
- JFCBMASK+5 (DCB work area) – bits 5-6 indicate function requiring user label processing.

Output: User labels written as requested

External References: User's label construction routine

Figure 69 (Chart 1 of 1). TCLOSE Direct-Access Processing

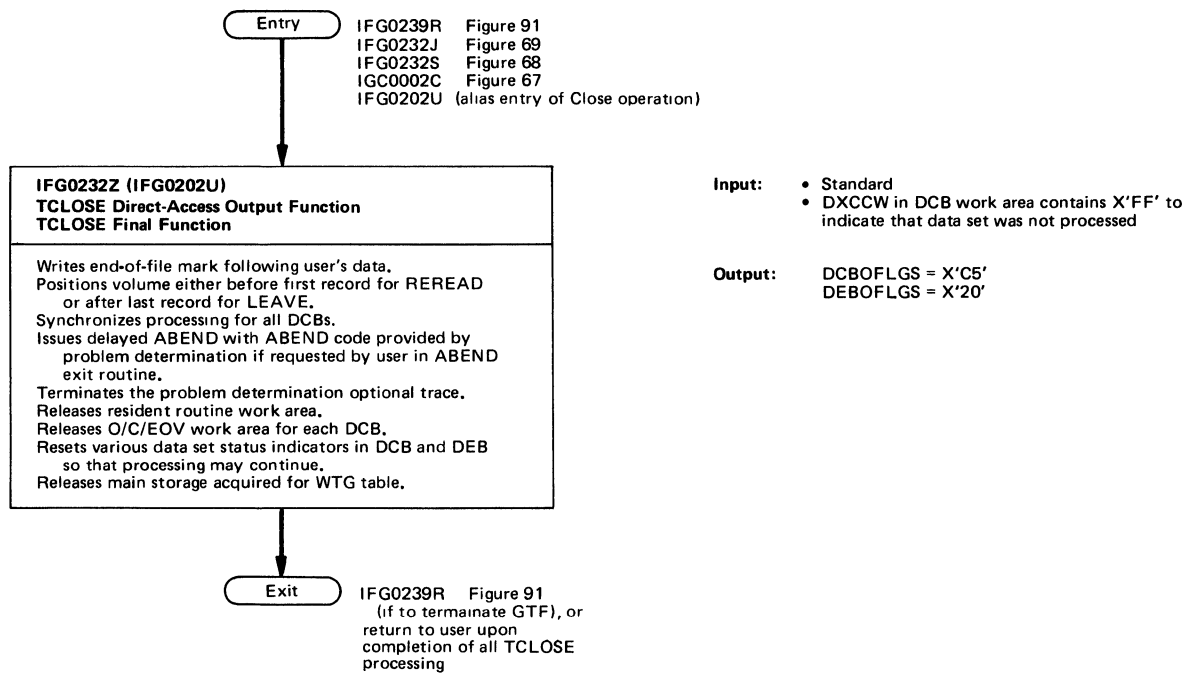
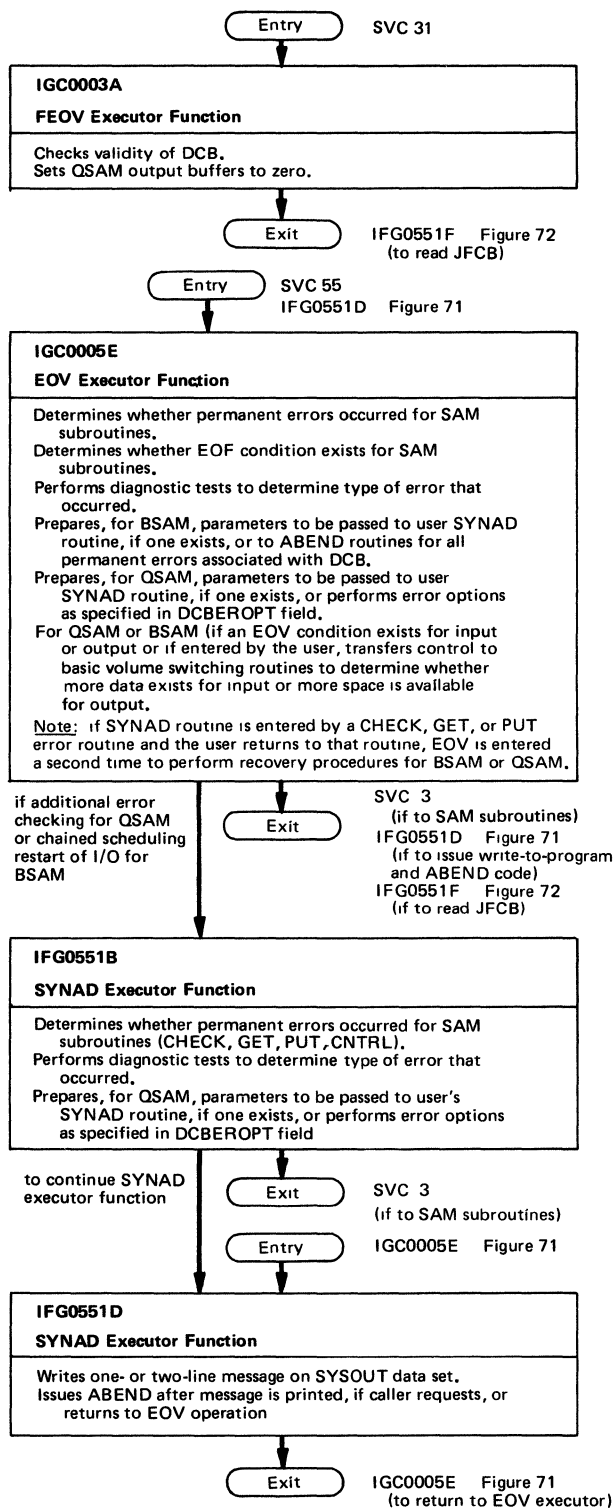


Figure 70 (Chart 1 of 1). TCLOSE Final Processing



Input: Standard
Output: Address of user's DCB in registers
External References: PUT routine, PUT error routine

Input:

- Register 0 contains IOB address from SAM subroutines
- Register 1 contains DCB address

Output: Parameters passed back to access method modules
Error Exits: IFG0550P with internal ABEND code 174

Input: Standard
Output: Parameters passed back to access method modules for CHECK, GET, and PUT error routines

Input:

- Register 1 points to parameter area constructed as follows:
 - Bytes 0 and 1 contain ABEND code
 - Byte 2 contains return code to be printed in message
 - If either ABEND code or return code is 0, message is not printed
 - Byte 3 is not used
 - Bytes 4-7 contain pointer to calling routine's 8-byte ID, if return is requested, and otherwise contain zeros
 - Bytes 8-11 contain pointer to calling routine's formatted message, if second line requested; otherwise, second line contains zeros
- Register 2 contains address of DCB on which error occurred
- Register 4 contains either pointer to DCB work area or zeros

Output: One- or two-line message containing ABEND code, return code, jobname, stepname, ddname, DSNNAME, and message passed by calling program

Figure 71 (Chart 1 of 3). EOV Initialization Processing

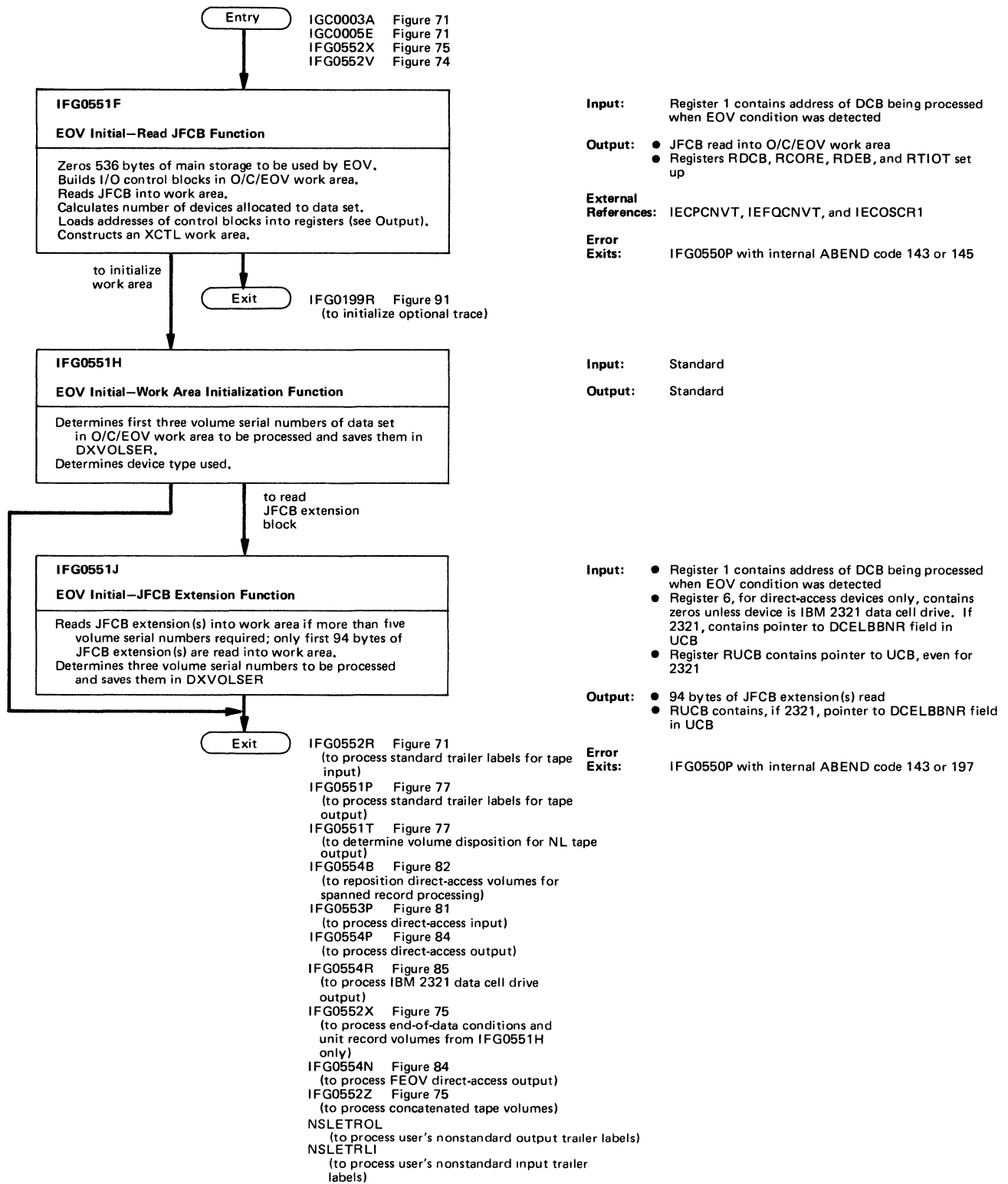


Figure 72 (Chart 2 of 3). EOVS Initialization Processing

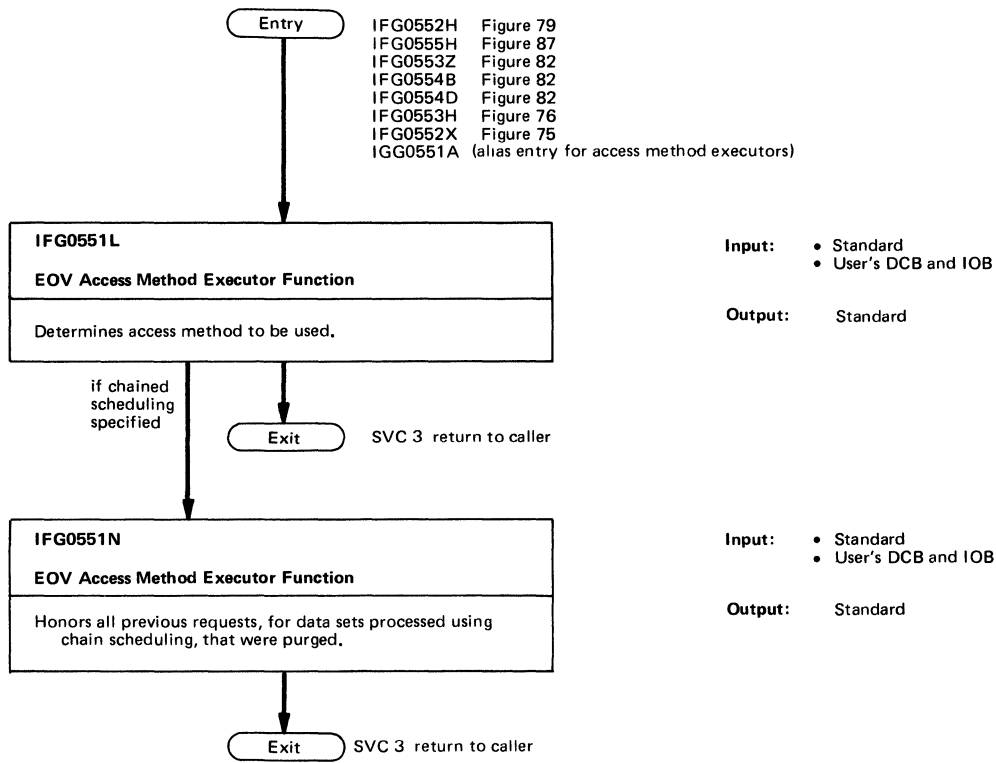


Figure 73 (Chart 3 of 3). EOVS Initialization Processing

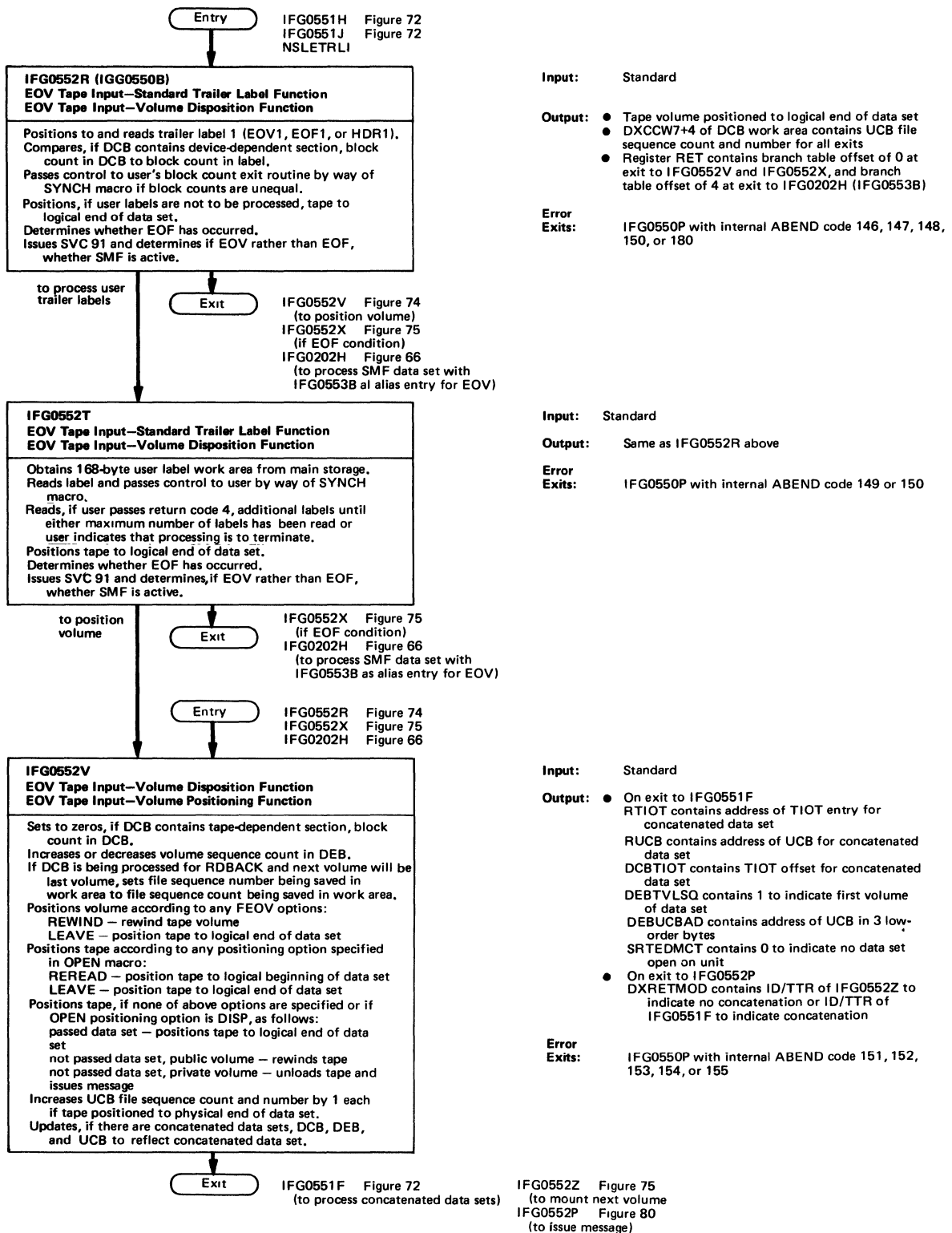


Figure 74 (Chart 1 of 3). EOVS Tape Input Processing

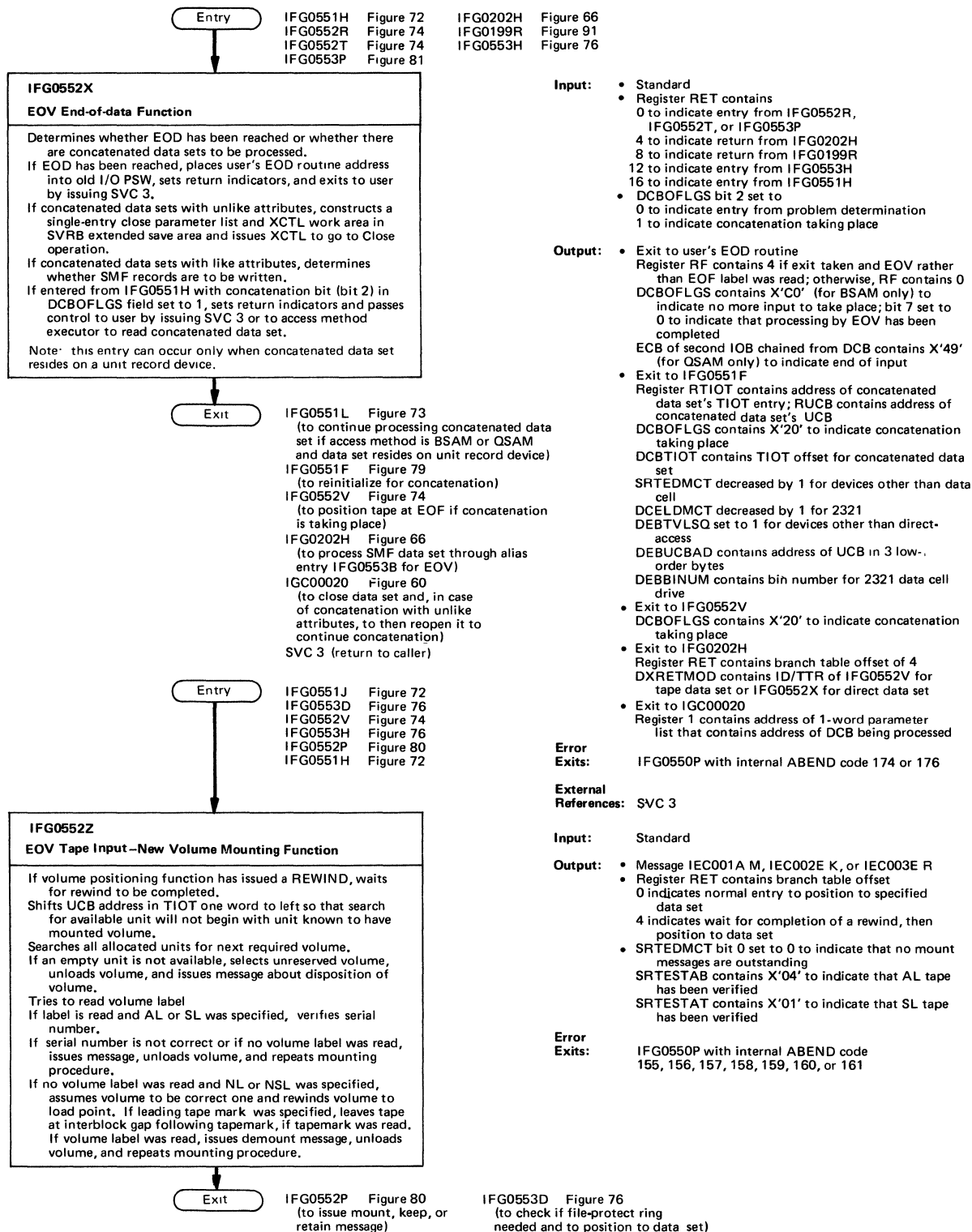
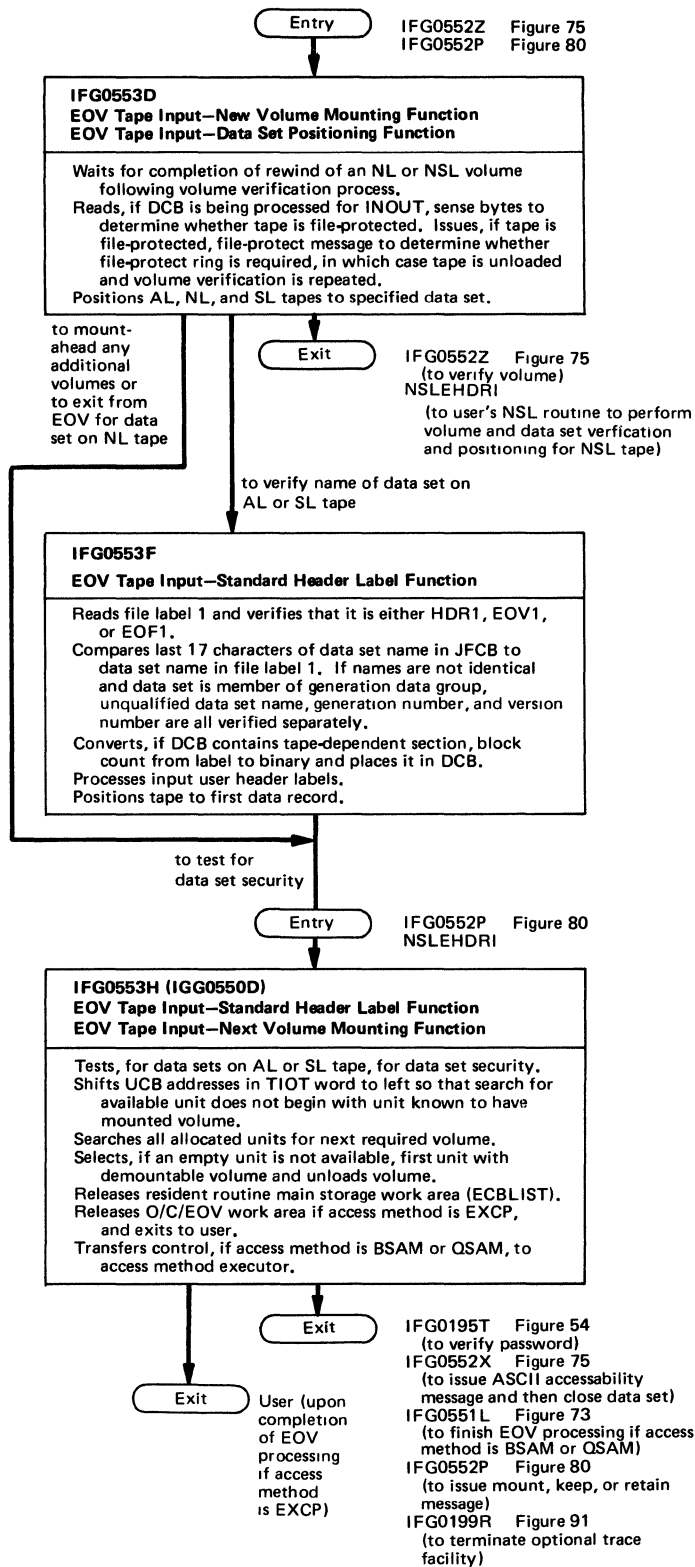


Figure 75 (Chart 2 of 3). EOF Tape Input Processing



Input: Standard *

Output:

- Exit to IFG0552Z: Register RET contains branch table offset of 12 SRTEVOLI contains 0 indicating that volume must be verified again UCBFL2 contains X'40' to indicate unit is not ready
- Exit to IFG0553H: Register RET contains branch table offset 24
- Exit to user's NSLEHDR1: Register RET contains branch table offset 20 for return to IFG0553H SRTEFSCT and SRTEFSEQ have been updated to reflect logical and physical position of data set SRTEDMCT has been set to 1

Error Exits: IFG0550P with internal ABEND code 162, 163, 193

Input: Standard

Output:

- DCBBLKCT contains block count found in file label 1
- Tape is positioned to first data record

Error Exits: IFG0550P with internal ABEND code 164, 165, 191, 195, or 196

Input: Standard

Output:

- Exit to IFG0195T: Register RET contains branch table offset of 0 DCBOFLGS bit 2 set to 0 to indicate concatenation has been completed DEBFLGS1 bit 0 set to 0 to indicate that password has not previously been verified
- Exit to IFG0552X: Register RET contains branch table offset of 12
- Exit to IFG0551L: Register RES no longer points to resident routine DCBOFLGS bit 2 set to 0 to indicate concatenation has been completed.
- Exit to IFG0199R: Register RET contains branch table offset of 12 Register RWTGC contains 1 to indicate, upon return from trace termination, that trace facility was active
- Exit to user: DCBOFLGS bit 2 set to 0 to indicate that concatenation has been completed; bit 7 set to 0 to indicate that processing by EOVS has been completed

Error Exits: IFG0550P with internal ABEND code 193

Figure 76 (Chart 3 of 3). EOVS Tape Input Processing

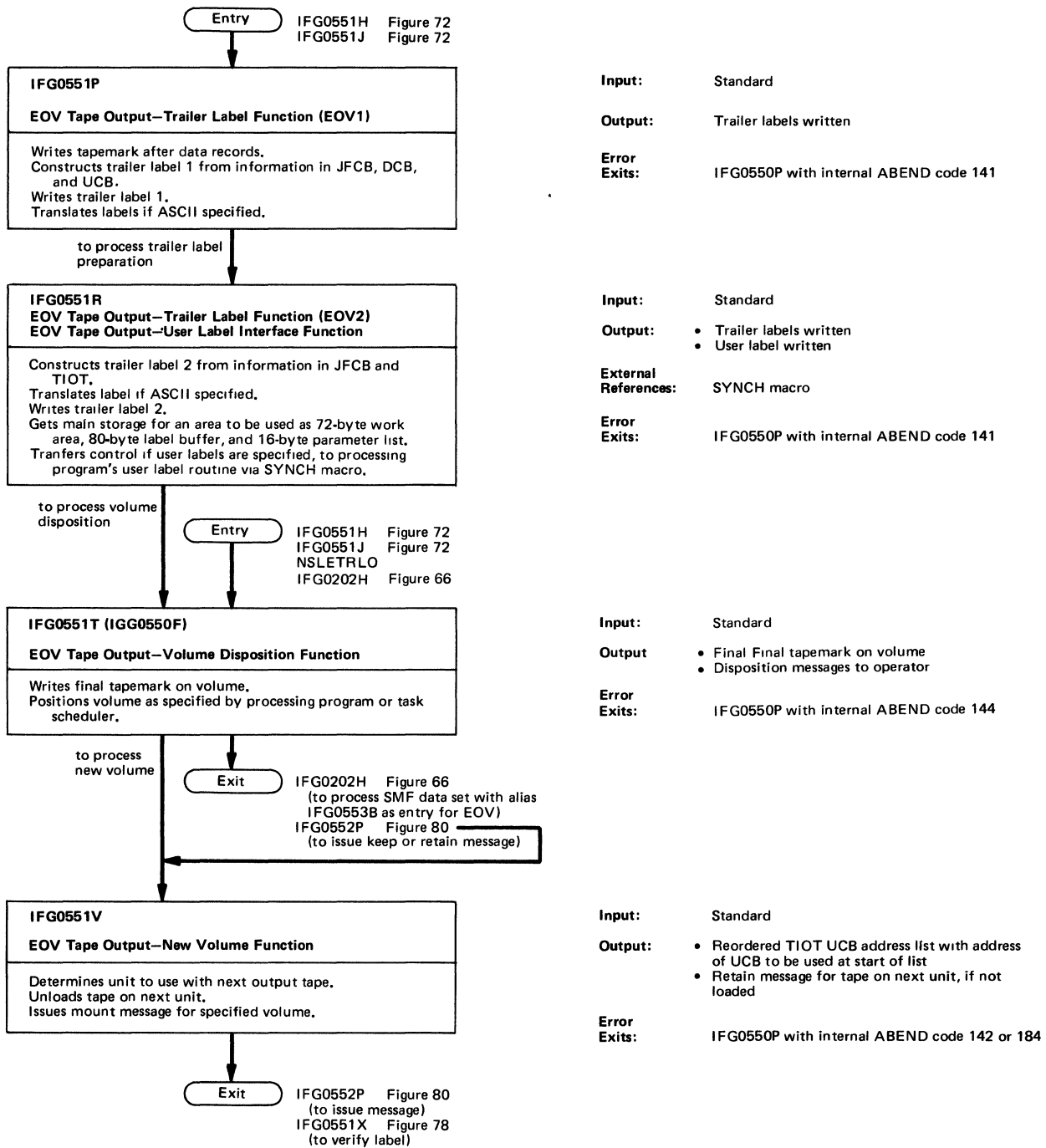


Figure 77 (Chart 1 of 4). EOVS Tape Output Processing

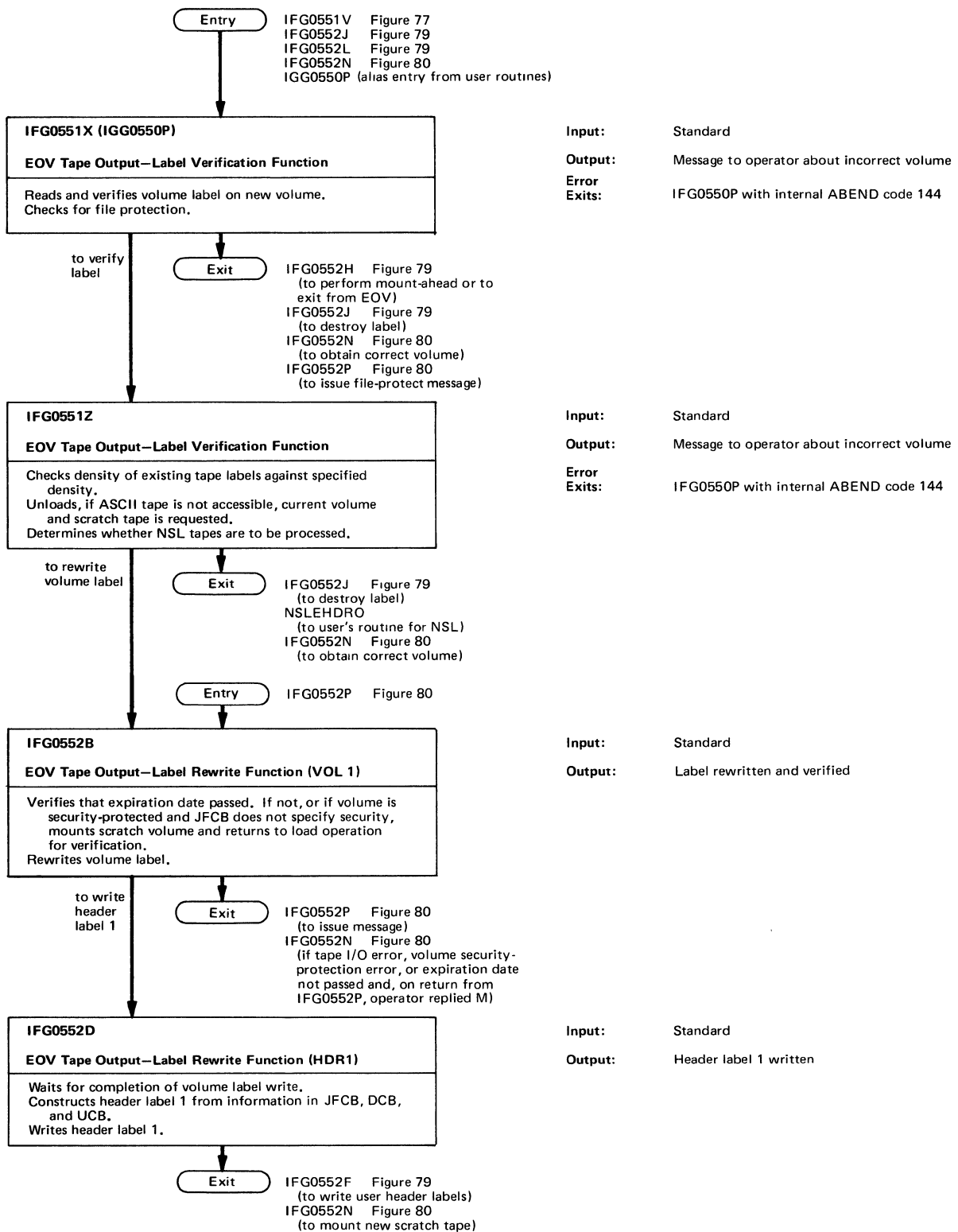


Figure 78 (Chart 2 of 4). EOVS Tape Output Processing

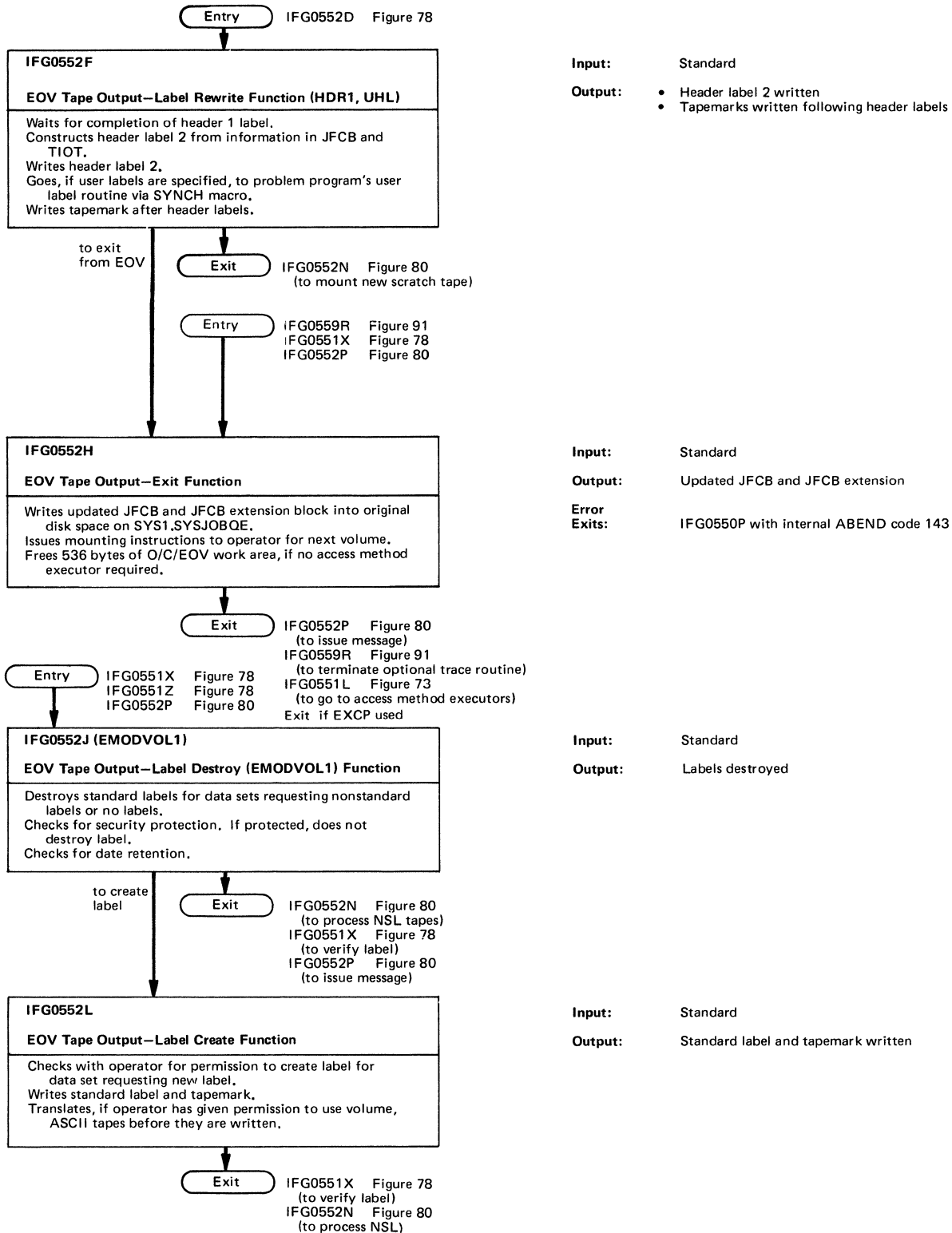


Figure 79 (Chart 3 of 4). EOVS Tape Output Processing

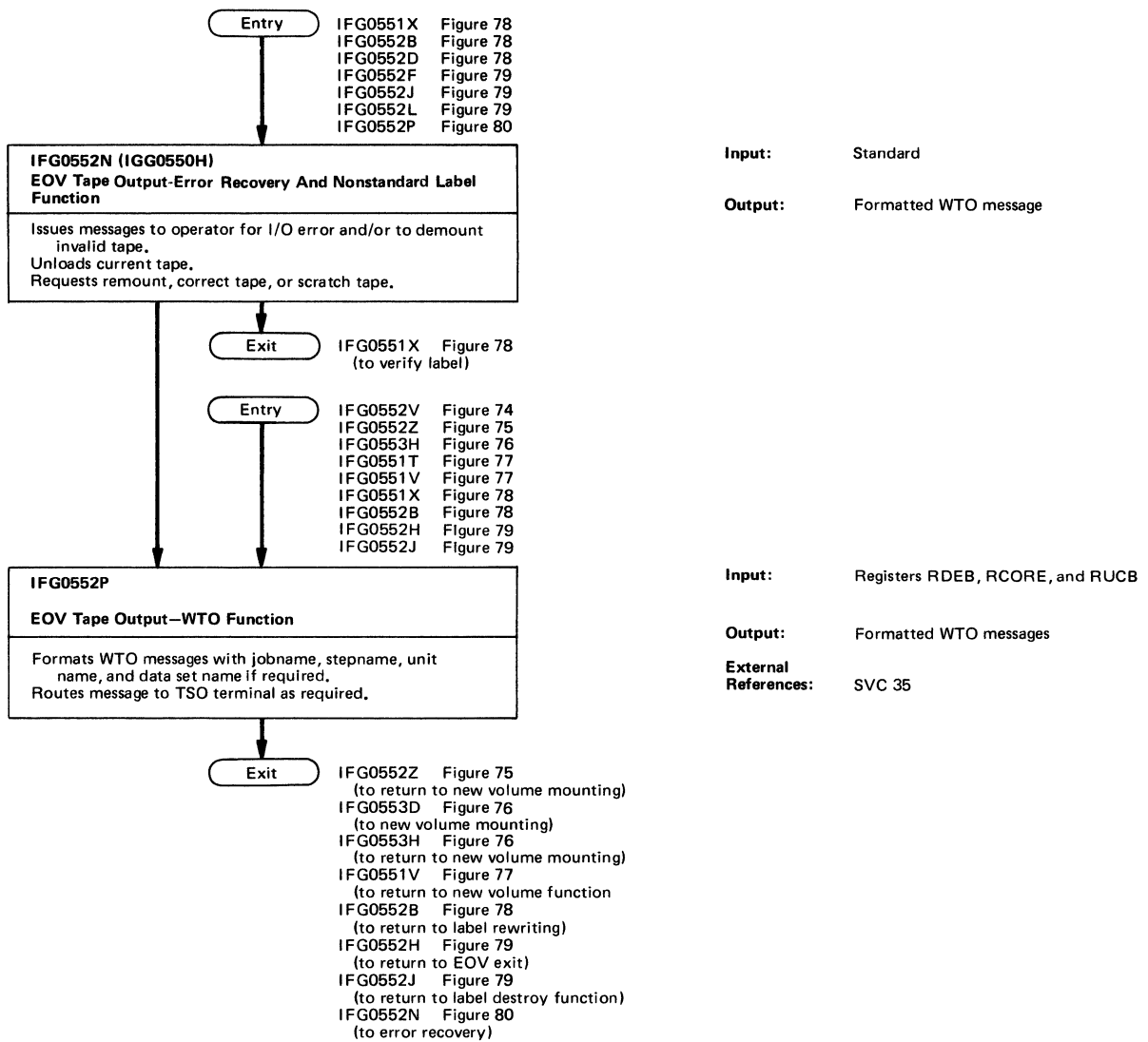


Figure 80 (Chart 4 of 4). EOVS Tape Output Processing

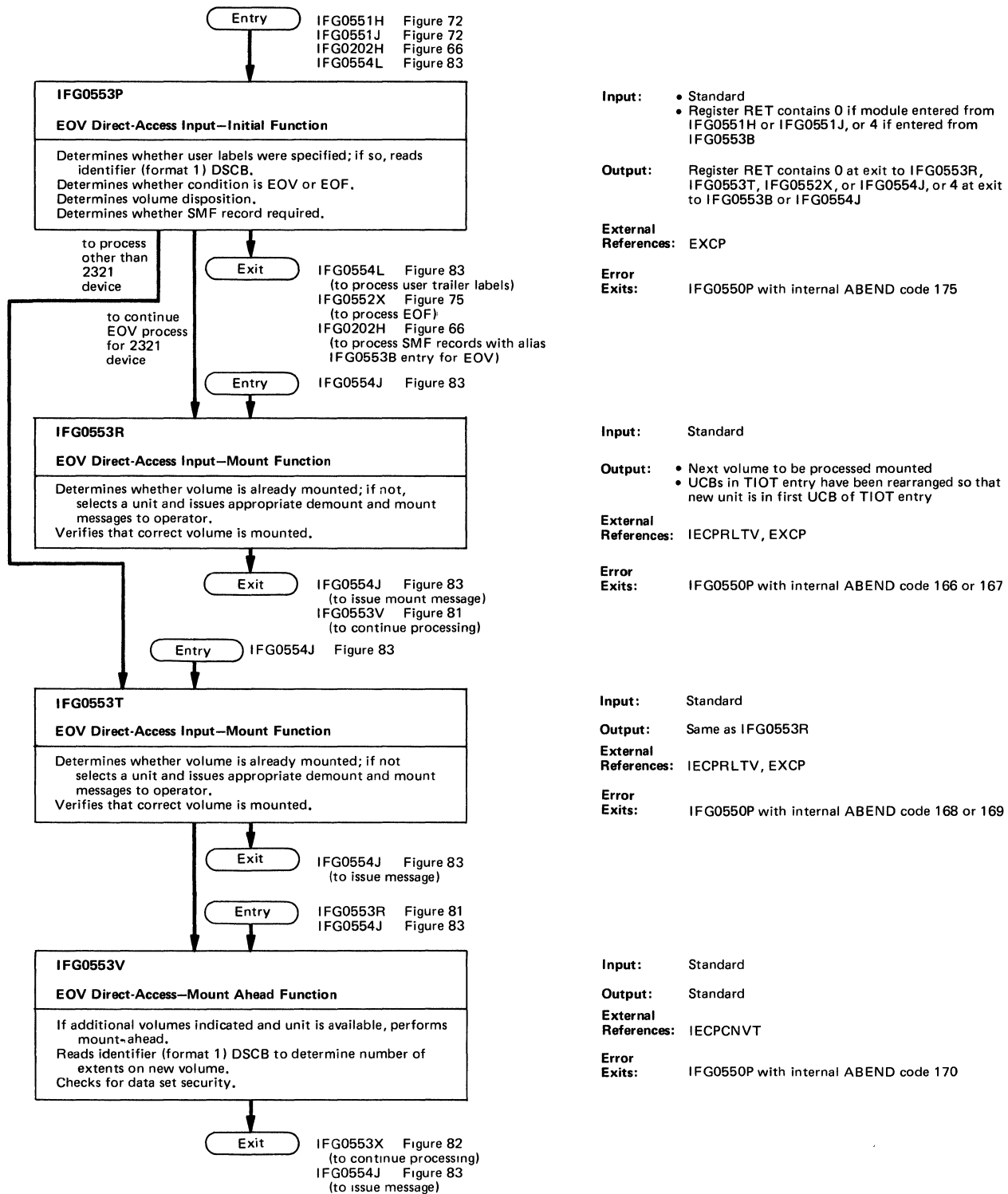


Figure 81 (Chart 1 of 3). EOV Direct-Access Input Processing

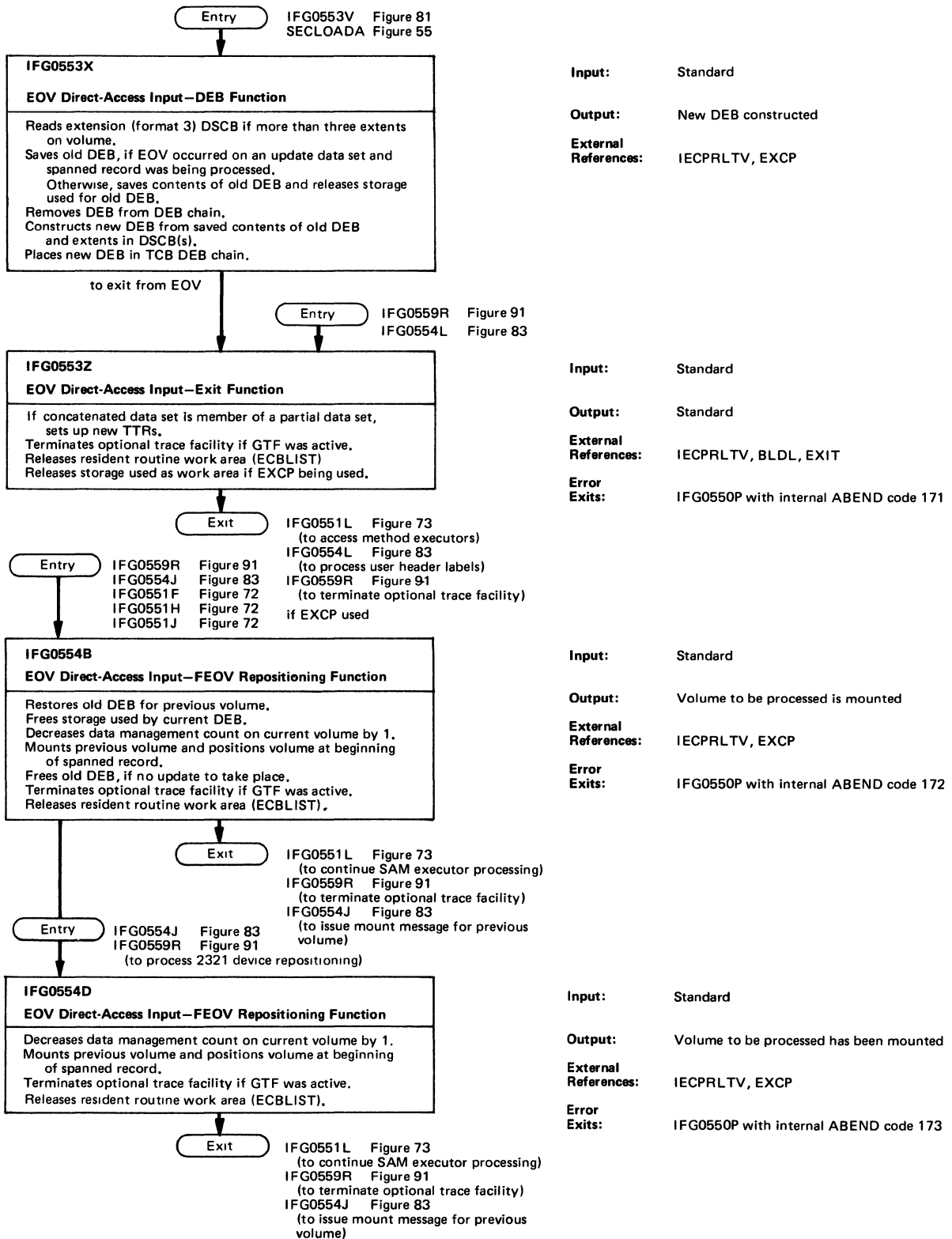


Figure 82 (Chart 2 of 3). EOVS Direct-Access Input Processing

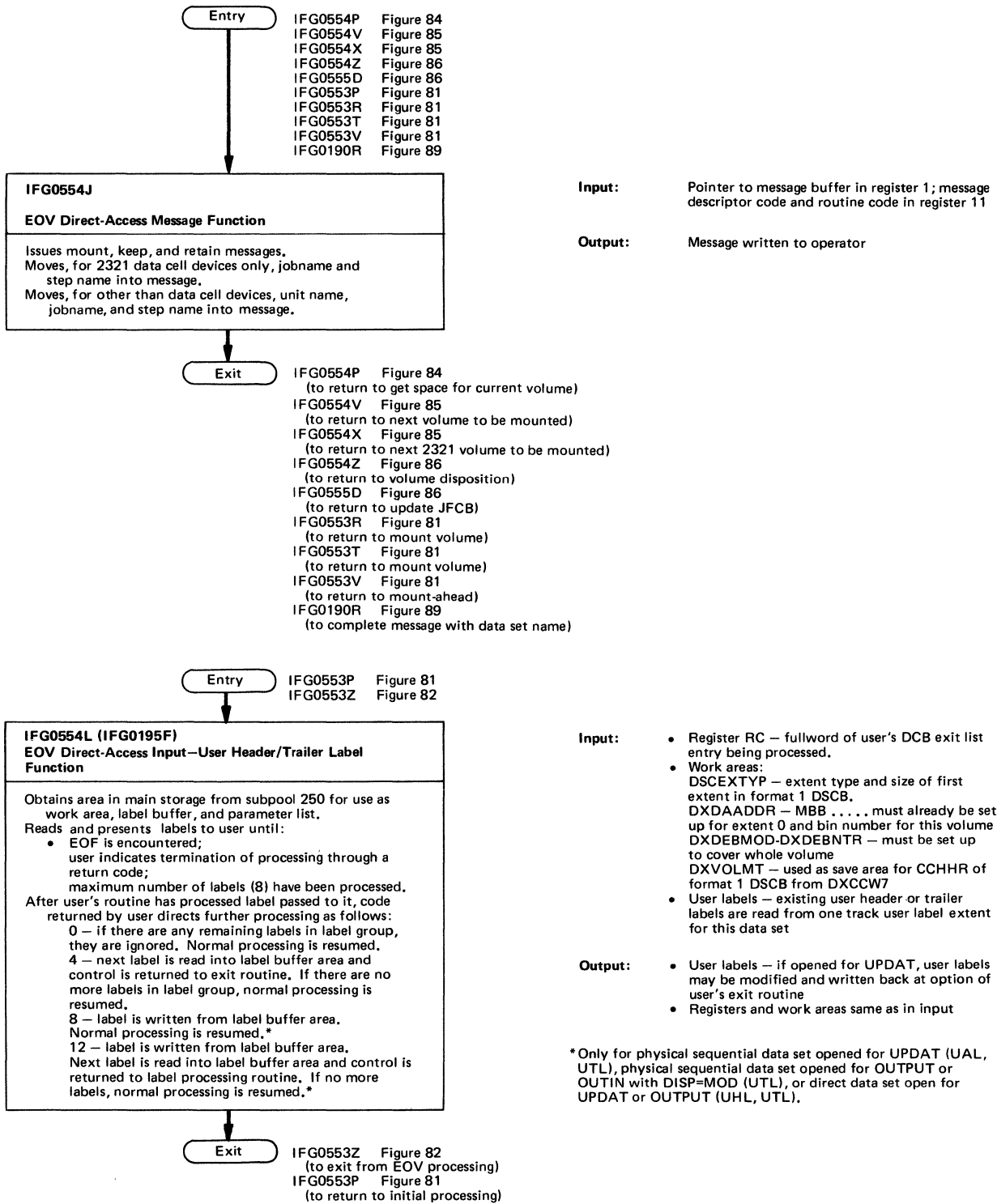


Figure 83 (Chart 3 of 3). EOVS Direct-Access Input Processing

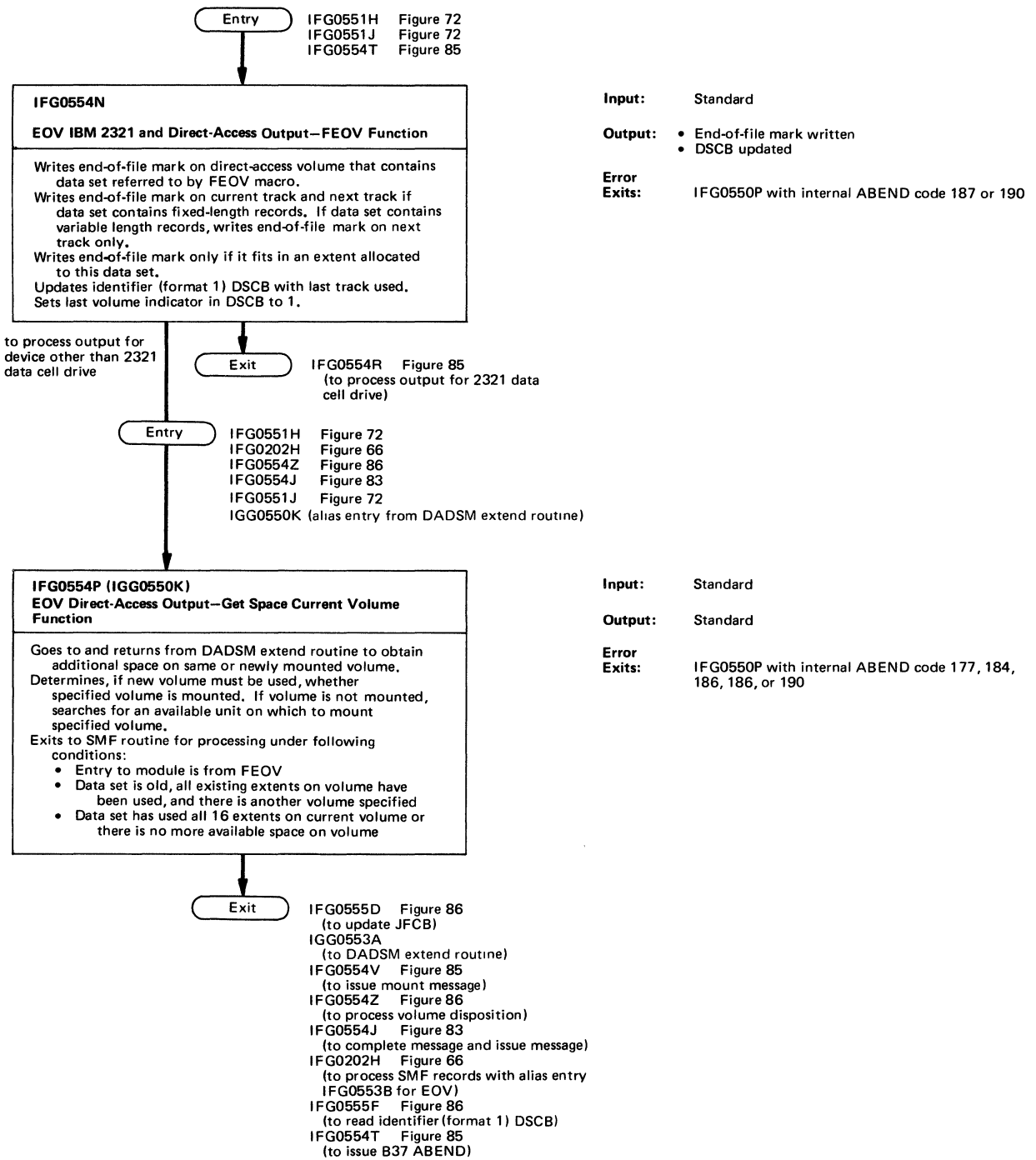


Figure 84 (Chart 1 of 4). EOVS Direct-Access Output Processing

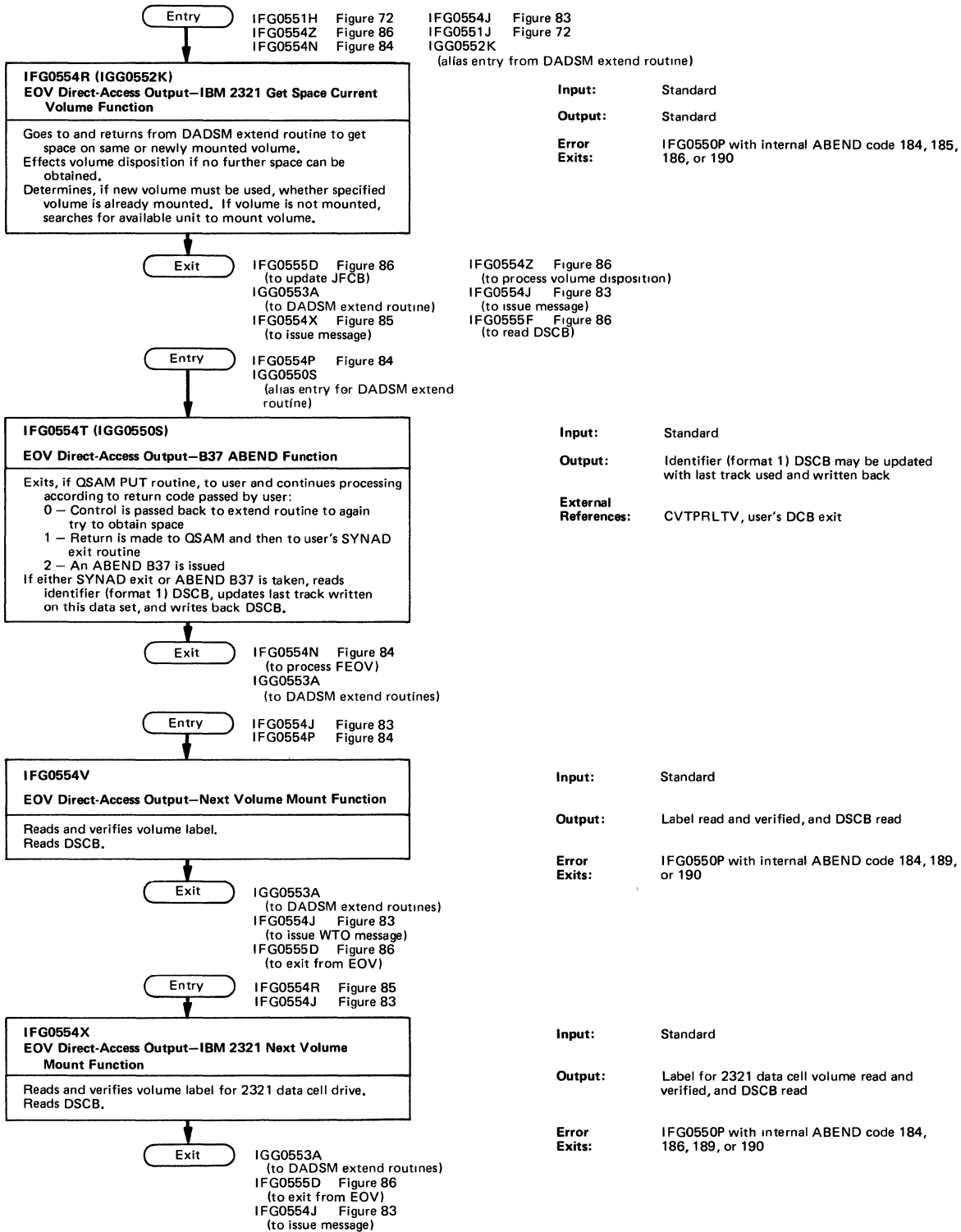


Figure 85 (Chart 2 of 4). EOVS Direct-Access Output Processing

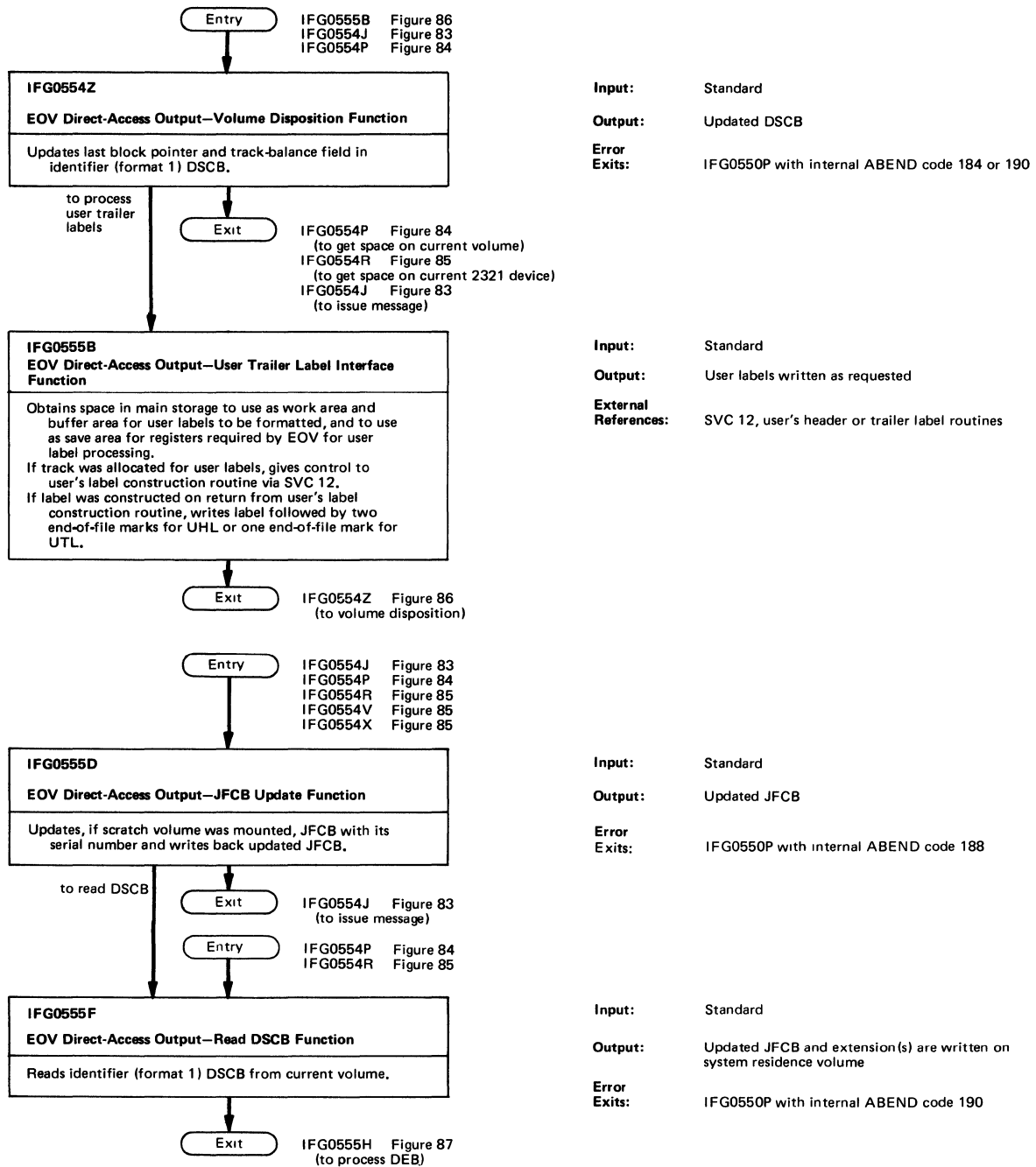
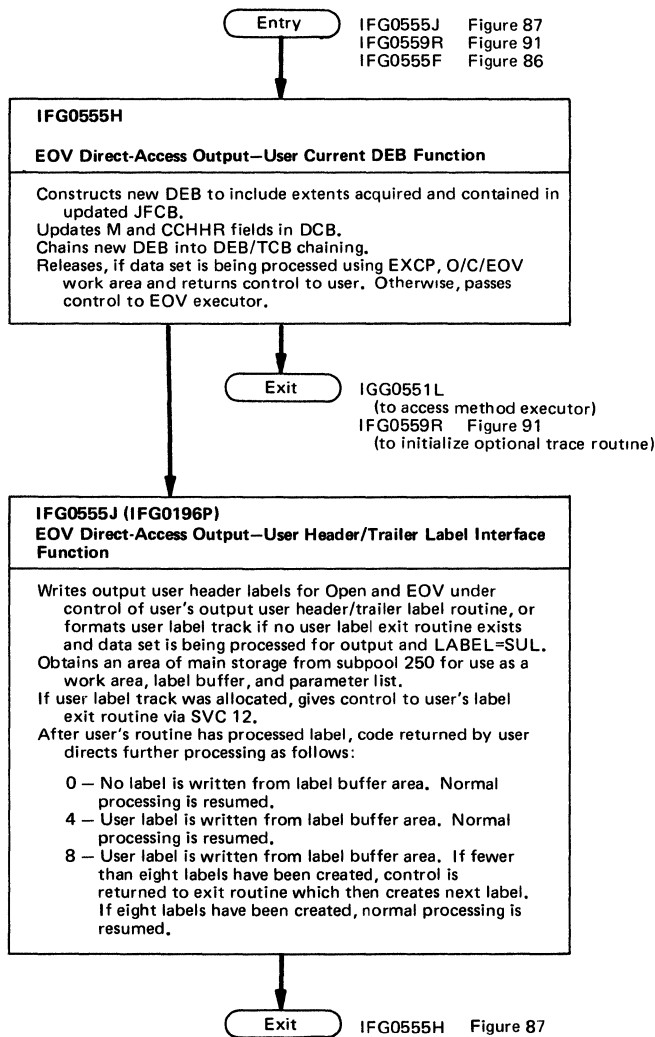


Figure 86 (Chart 3 of 4). EOVS Direct-Access Output Processing



Input: Standard

Output: DEB constructed

Error Exits: IFG0550P with internal ABEND code 190

Input:

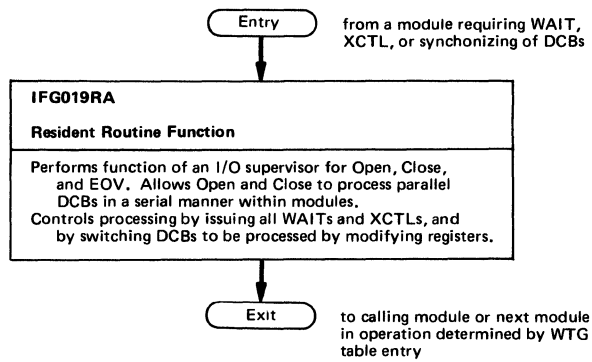
- Register RC — contains fullword of user's DCB exit list entry that is being processed
- Work areas:
 - RBGRS2-RBGRS13 — contains user's registers 2-13, saved in SVRB register save area
 - DSCEXTYP — extent type and size of first extent in identifier (format 1) DSCB
 - DXDAADDR — MB set up for extent 0 and bin number for this volume
 - DXDEBMOD-DXDEBNTR — set up to cover whole volume, and therefore to cover UL extent
 - DXVOLMT2 — used as save area by Open only to save CCHHR of identifier (format 1) DSCB from DXCCW7 since DXCCW7 is needed for channel program

Output:

- User labels constructed by user label exit routine are written on user label track allocated for this data set
- Registers 2 and 4-14 have the same contents as at entry, allowing this module to be called as though it were a subroutine
- Work areas:
 - DXDAADDR — MBB same as at entry to module

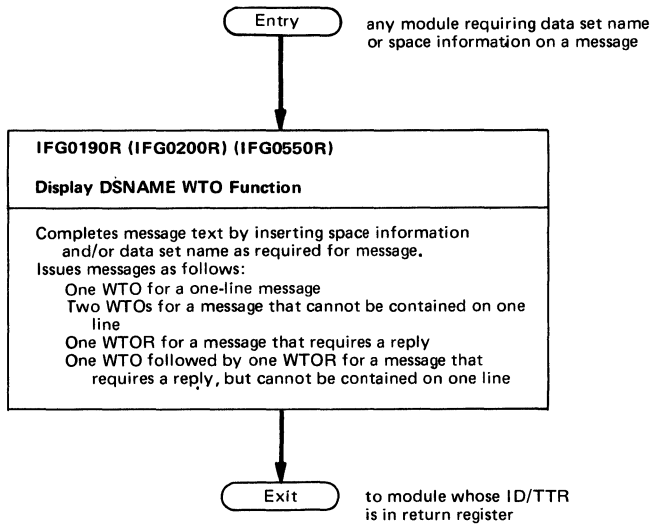
External References: User's input user header or trailer label exit routine

Figure 87 (Chart 4 of 4). EOJ Direct-Access Output Processing



- Input:**
- Registers:
 - 0 – Saved by resident routine in O/C/EOV work area
 - 1 – Saved by resident routine in O/C/EOV work area
 - 3 – Base register of calling module
 - 4 – Address of DCBs O/C/EOV work area
 - 5 – Address of resident routine
 - 6 – Where-to-go table address
 - 9 – Saved by resident routine in O/C/EOV work area
 - 10 – Saved by resident routine in O/C/EOV work area
 - 11 – Saved by resident routine in O/C/EOV work area
 - 12 – Saved by resident routine in O/C/EOV work area
 - 13 – Saved by resident routine in O/C/EOV work area
 - 14 – Return address to calling module for WAIT and DCB synchronization. May be a branch table offset for the next module when XCTL issued; saved by resident routine in O/C/EOV work area
 - Where-to-go table:
 - + 0 – 8-byte current module name
 - + 6 – 2-byte current module ID
 - +16 – Fullword with resident routine work area address in low order three bytes
 - +31 – (WTGPATHS+1) trace routine indicator
 - Where-to-go table DCB entry
 - + 0 – 2-byte module ID
 - + 2 – 3-byte module TTR
 - + 5 – DCB work area address
 - O/C/EOV work area:
 - JFCBMASK+6 – Trace routine DCB indicator
 - DXREG9 – Register save area (register 9)
 - DXREGA – Register save area (register 10)
 - DXREGB – Register save area (register 11)
 - DXREGC – Register save area (register 12)
 - DXREGD – Register save area (register 13)
 - DXREG E – Register save area (register 14)
 - DXREG0 – Register save area (register 0)
 - DXREG1 – Register save area (register 1)
 - DXRESIND –
 - X'80' – WAIT pending
 - X'40' – Pseudo-WAIT
 - DXECB – ECB
 - Resident routine work area
 - 8 – Prefix for optional trace, address of optional trace routine
 - + 0 – Fullword with parameter list address
 - + 4 – Multiple WAIT ECBLIST
- Output:**
- Registers:
 - 0 – Contents of DXREG0
 - 1 – Contents of DXREG1
 - 2 – DCB address (from Open/Close parameter list)
 - 4 – DCB work area address (from WTG table DCB entry)
 - 7 – Address of current parameter list entry
 - 8 – Address of current WTG DCB entry
 - 9 – (Address of optional trace function prefix to the Resident routine work area when calling the optional trace routine)
 - 9 – Contents of DXREG9
 - 10 – Contents of DXREGA
 - 11 – Contents of DXREGB
 - 12 – Contents of DXREGC
 - 13 – Contents of DXREGD
 - 14 – Contents of DXREG E
 - Where-to-go table:
 - + 0 – 8-byte module name
 - +14 – 3-byte TTR
 - Work area:
 - DXREG9 – Register save area (register 9)
 - DXREGA – Register save area (register 10)
 - DXREGB – Register save area (register 11)
 - DXREGC – Register save area (register 12)
 - DXREGD – Register save area (register 13)
 - DXREG E – Register save area (register 14)
 - DXREG0 – Register save area (register 0)
 - DXREG1 – Register save area (register 1)
 - DXRESIND –
 - X'80' – WAIT pending
 - X'40' – Pseudo-WAIT

Figure 88. Open, Close, and EOVS Resident Routine



- Input:**
- Registers:
 - R1 contains pointer to last character of message; High-order bit indicates reason for entry:
 - 0 – DSNAME required in message
 - 1 – Space information required in message
 - RUCB contains pointer to beginning of UCB or DCELBBNR field of UCB
 - RB contains MCS routing and descriptor codes
 - Work area:
 - ID/TTR of return module
 - Reply, indicating whether message is WTO or WTOR and is DSNAME is to be taken from JFCB or work area
- Output:**
- Message written with required space information or data set name
 - Register RB indicates whether one- or two-line WTO was issued:
 - 0 – one line
 - not blank – two lines

Figure 89. Write-to-Operator Module

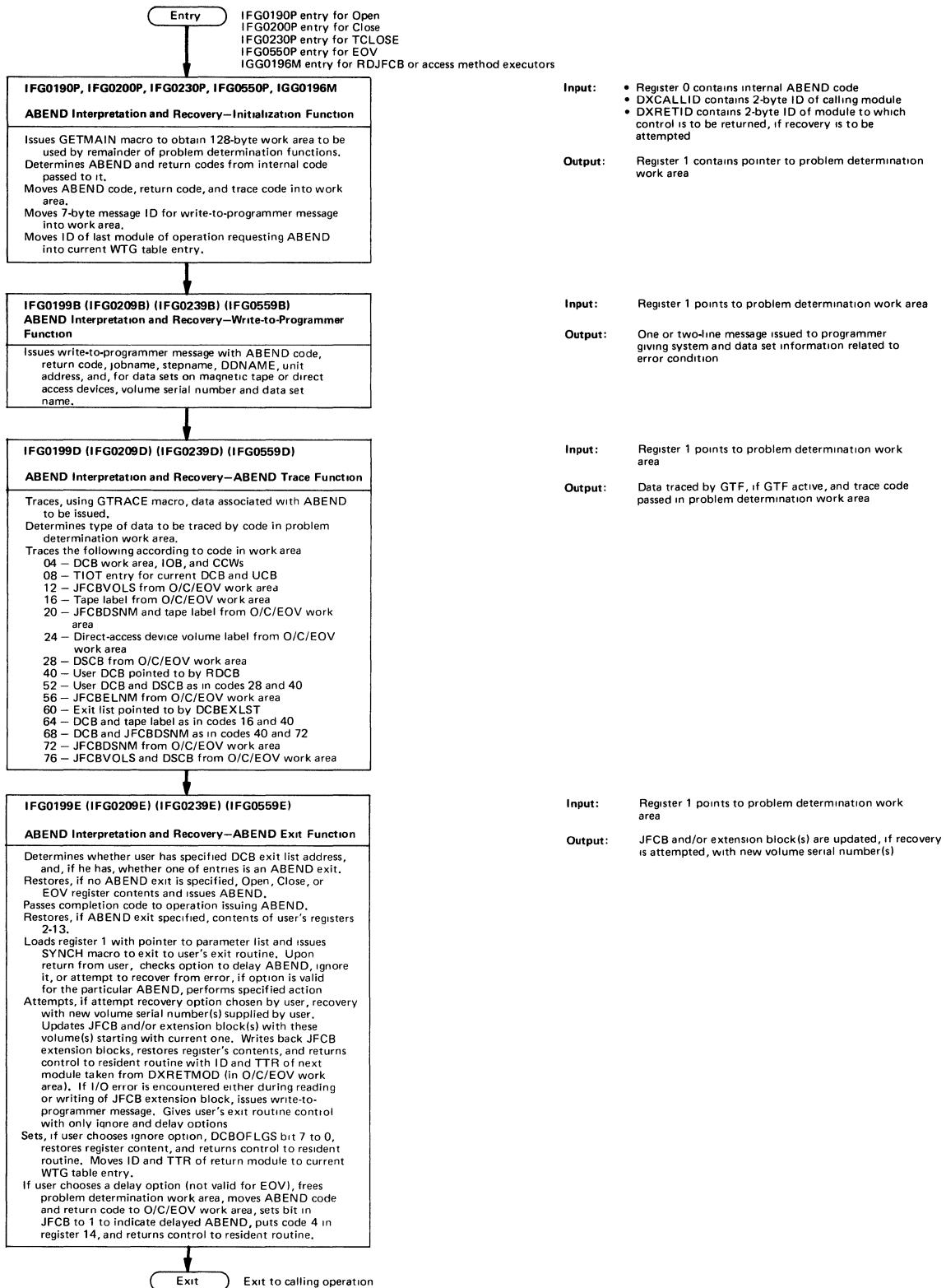


Figure 90 (Chart 1 of 2). Problem Determination

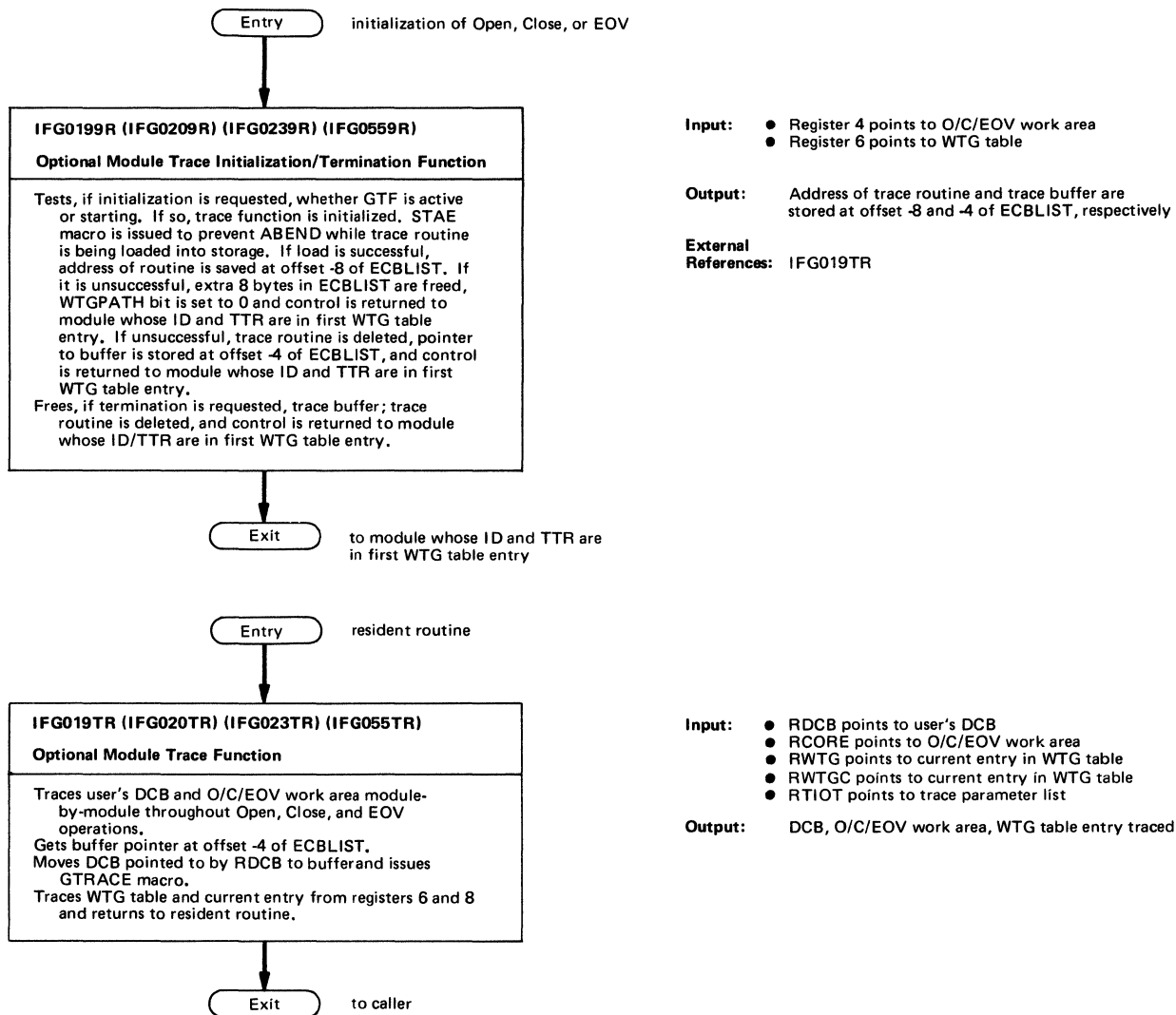


Figure 91 (Chart 2 of 2). Problem Determination

SECTION 4: DIRECTORY

Function-to-Module Directory

This directory lists the functions that are performed in the Open, Close, and EOJ operations and the functions that are common to these three operations. Knowing the name of a function, you can use this directory to learn what modules make up the function and where to find more information about the function.

The directory alphabetically lists the functions performed under each of the following categories:

- Open (Including OPEN TYPE=J) Operation
- Close (Including CLOSE TYPE=T) Operation
- EOJ Operation
- Common Functions (Including Problem Determination)

The column headings and their meanings are:

Function Name: This name refers to a function in the category it is listed under.

Method of Operation Figure: This is the number of the figure in the “Method of Operation” section where the function is described.

Module Name: This is the symbolic name of the object module. The microfiche cards containing program listings are ordered by object module name.

Note: The modules listed below have different names in the microfiche and in the SVCLIB. They are listed in the directory under both names.

<u>SVCLIB Name</u>	<u>Microfiche Name</u>
IFG0552J	EMODVOL1
IFG0193C	OMODVOL1

Program Organization Figure: This is the number of the figure in the “Program Organization” section where the module is described.

Alias Name For: If the Module name listed under “Module Name” is an alias name for another module name, the name under which the module is described in the “Program Organization” section is given here.

Open (Including Open TYPE=J) Operation

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Open Access Method Executor Determination Function	13	IFG0196V	58	
Open Access Method Executor Return Function	13	IFG0196W	59	
		IGG0190S	59	IFG0196W

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Open Direct-Access BPAM Concatenation Function	5	IFG0195M	51	
Open Direct-Access DISP=MOD Error Recovery Function	4	IFG0195E	52	
Open Direct-Access DSCB To JFCB Merge Function	4	IFG0195A	51	
		IFG0195J	51	
Open Direct-Access Expiration Date Error Function	4	IFG0195G	52	
Open Direct-Access Input User Label Function	4	IFG0195F	53	IFG0554L
Open Direct-Access Parallel Mounting Function	5	1FG0195O	52	
		IFG0195P	52	
Open Direct-Access Read DSCB Function	4	IFG0194E	51	
		IFG0195A	51	
Open Direct-Access Unit Selection Function	4	IFG0194E	51	
Open Direct-Access Volume Verification Function	5	IFG0194A	53	
		IFG0194C	53	
Open Direct-Access 213 ABEND Exit Function	4	IFG0195G	52	
Open Final EXCP Appendage Function	14	IFG0196X	59	
Open Final JFCB To DSCB Merge Function	14	IFG0196W	59	
		IFG0196X	59	
Open Final SYSOUT Limit Function	14	IFG0196X	59	
Open Final Termination Function	14	IFG0198N	59	
Open Final Write JFCB Function	14	IFG0196X	59	
		IFG0198N	59	
Open Initial Function (OPEN and OPEN TYPE=J)	3	IGC0001I	44	
		IGC0002B	44	
Open Initial Volume Serial Function	3	IFG0193A	44	
Open Merge DCB Exit Function	12	IFG0196L	56	
Open Merge DCB to JFCB Function	12	IFG0196M	56	
Open Merge Direct-Access Output User Header Label Function	12	IFG0196P	57	IFG0555J
		IFG0555J	87	

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Open Merge JFCB To DCB Function	12	IFG0196J	56	
		IFG0196K	56	
Open Security Initialization Function	11	IFG0195T	54	
Open Security Read Password (READPSWD) Function	11	READPSWD	55	
Open Security Scratch-Rename Interface Function	11	SECLOADA	55	
Open Security Search Function	11	IFG0195U	55	
Open Security TSO Password Function	11	IFG0195V	54	
Open Tape Create Label Function	7	IFG0193E	47	
Open Tape Destroy Label Function	7	IFG0193D	47	
Open Tape Final Common Function	6	IFG0194I	46	
Open Tape Initial Common Function	6	IFG0193B	45	
Open Tape Label Editor Function	7	IFG0193C	47	
		OMODVOL1	47	IFG0193C
Open Tape Label Editor Interface Function	7	IFG0194J	47	
		IGG0190A	47	IFG0194J
Open Tape Mount Verification Function	6	IFG0194F	45	
Open Tape No Label Positioning Function	9	IFG0195C	49	
Open Tape Nonstandard Label Input Interface Function	9	IFG0195D	49	
		IGG0190B	49	IFG0195D
Open Tape Nonstandard Label Output Interface Function	9	IFG0196O	49	
		IGG0190R	52	IFG0196O
Open Tape Standard Label Date Protection Function	10	IFG0196Q	50	
Open Tape Standard Label Header Label Writing Function	10	IFG0196T	50	
Open Tape Standard Label INPUT/MOD Header Label 1 Function	8	IFG0195B	48	
		IFG0195H	48	
Open Tape Standard Label INPUT/MOD Header Label 2 Function	8	IFG0195H	48	
		IFG0195K	48	

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Open Tape Standard Label Input User Label Function	8	IFG0195N	48	
Open Tape Standard Label Output Security Function	10	IFG0196N	50	
Open Tape Standard Label Output User Label Function	10	IFG0196U	50	
Open Tape Standard Label Positioning Function	8	IFG0195B	48	
Open Tape Standard Label Rewrite Volume Label Function	10	IFG0196Q	50	
		IFG0196T	50	
Open Tape Volume Mounting Function	6	IFG0194G	46	
Open Tape Volume Reference Function	7	IFG0194D	45	
Open Tape Volume Verification Function	6	IFG0194H	46	

Close (Including Close TYPE=T) Operation

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Close Direct-Access Exit Function	20	IFG0202E	64	
Close Direct-Access Input User Labels Function	20	IFG0202C	64	
Close Direct-Access Method Executor Interface Function	17	IFG0200W	63	
		IFG0200Y	63	
		IGG0200F	63	IFG0200Y
		IGG0200G	63	IFG0200Y
Close Direct-Access Output User Labels Function	20	IFG0201R	63	
		IFG0202D	64	
Close Direct-Access Partial Release Interface Function	20	IFG0200Y	63	
		IGG0200F	63	IFG0200Y
		IGG0200G	63	IFG0200Y
Close Direct-Access Read DSCB Function	None	IFG0200V	63	
Close Direct-Access Update DSCB Function	20	IFG0200Y	63	
		IGG0200F	63	IFG0200Y
		IGG0200G	63	IFG0200Y

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
Close Direct-Access Write DSCB Function	20	IFG0201R	63	
Close Direct-Access Write End-of-File Mark Function	20	IFG0202E	64	
Close Final Restore Data Management Function	22	IFG0202J	65	
Close Final Restore System Function	22	IFG0202J	65	
		IFG0202K	65	
Close Final Termination Function	22	IFG0202K	65	
		IFG0202L	65	
Close Initialization Function	17	IFG0200V	60	
		IGC00020 ¹	60	
Close SMF Data Set SMF Record Builder Function	21	IFG0202H	66	
		IFG0202I	66	
		IFG0553B	66	IFG0202H
Close Tape Access Method Executor Interface Function	17	IFG0200W	61	
		IFG0200Y	61	
		IGG0200F	61	IFG0200Y
		IGG0200G	61	IFG0200Y
Close Tape Nonstandard Label Function	18	IFG0202B	62	
		IFG0232Y	62	IFG0202B
		IFG0556B	62	IFG0202B
		IGG0200B	62	IFG0202B
Close Tape Standard Trailer Label Function	18	IFG0200Z	61	
Close Tape Standard User Label Function	18	IFG0202A	61	
Close Tape Volume Disposition Function	19	IFG0202F	62	
		IFG0202G	62	
Close Unit Record/Teleprocessing Function	None	IFG0200W	60	
		IFG0200Y	60	
		IGG0200F	60	IFG0200Y
		IGG0200G	60	IFG0200Y
TCLOSE Direct-Access Input Function	26	IFG0232D	69	
TCLOSE Direct-Access Output Function	26	IFG0232D	69	
TCLOSE Direct-Access Output Trailer Label Function	26	IFG0232J	69	

¹The final 0 of this module name is actually an EBCDIC 12-punch over a 0-punch.

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
TCLOSE Final Function	27	IFG0202U	70	IFG0232Z
		IFG0232Z	70	
TCLOSE Initial Function	24	IGC0002C	67	
TCLOSE Tape Standard Trailer Label Function	25	IFG0232G	68	
		IFG0232M	68	
TCLOSE Tape Volume Positioning Function	25	IFG0232S	68	

EOV Operation

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
EOV Access Method Executor Function	37	IFG0551L	73	
		IFG0551N	73	
EOV Direct-Access Input Concatenation/End-of-Data Function	37	IFG0552X	75	
EOV Direct-Access Input DEB Function	36	IFG0553X	82	
EOV Direct-Access Input Exit Function	37	IFG0553Z	82	
EOV Direct-Access Input FEOV Repositioning Function	37	IFG0554B	82	
		IFG0554D	82	
EOV Direct-Access Input Initial Function	36	IFG0553P	81	
EOV Direct-Access Input Mount Ahead Function	36	IFG0553V	81	
EOV Direct-Access Input Mount Function	36	IFG0553R	81	
		IFG0553T	81	
EOV Direct-Access Input User Header/Trailer Label Function	37	IFG0554L	83	
EOV Direct-Access Output B37 ABEND Function	38	IFG0554T	85	
EOV Direct-Access Output Current DEB Function	None	IFG0555H	87	
EOV Direct-Access Output Get Space Current Volume Function	38	IFG0554P	84	
EOV Direct-Access Output And IBM 2321 FEOV Function	None	IFG0554N	84	

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
EOV Direct-Access Output IBM 2321 Get Space Current Volume Function	36	IFG0554R	85	
		IGG0552K	85	IFG0554R
EOV Direct-Access Output IBM 2321 Next Volume Mount Function	36	IFG0554X	85	
EOV Direct-Access Output JFCB Update Function	39	IFG0555D	86	
EOV Direct-Access Output Next Volume Mount Function	None	IFG0554V	85	
EOV Direct-Access Output Read DSCB Function	39	IFG0555F	86	
EOV Direct-Access Output User Header Label Interface Function	39	IFG0555J	87	
EOV Direct-Access Output User Trailer Label Interface Function	38	IFG0555B	86	
EOV Direct-Access Output Volume Disposition Function	39	IFG0554Z	86	
EOV End-of-Data Function	31	IFG0552X	75	
EOV Executor Function		IGC0005E	71	
EOV Initial JFCB Extension Function	None	IFG0551J	72	
EOV Initial Read JFCB Function	30	IFG0551F	72	
EOV Initial Device Determination Function	30	IFG0551H	72	
		IFG0551J	72	
EOV Initial Work Area Initialization Function	30	IFG0551H	72	
EOV Tape Input Data Set Positioning Function	32	IFG0553D	76	
EOV Tape Input New Volume Mounting Function	32	IFG0552Z	75	
		IFG0553D	76	
EOV Tape Input Next Volume Mounting Function	32	IFG0553H	76	
		IGG0550D	76	IFG0553H
EOV Tape Input Standard Header Label Function	32	IFG0553F	76	
EOV Tape Input Standard Trailer Label Function	31	IFG0552R	74	
		IFG0552T	74	
EOV Tape Input Volume Disposition Function	31	IFG0552R	74	
		IGG0550B	74	IFG0552R

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
EOV Tape Input Volume Positioning Function	31	IFG0552V	74	
EOV Tape Output Error Recovery and Nonstandard Label Function	35	IFG0552N	80	
		IGG0550H	80	IFG0552N
EOV Tape Output Exit Function	35	IFG0552H	79	
EOV Tape Output Label Create Function	35	IFG0552L	79	
EOV Tape Output Label Destroy (EMODVOL1) Function	35	EMODVOL1	79	IFG0552J
		IFG0552J	79	
EOV Tape Output Label Rewrite Function (HDR1)	34	IFG0552D	78	
EOV Tape Output Label Rewrite Function (HDR2, UHL)	34	IFG0552F	79	
EOV Tape Output Label Rewrite Function (VCL1)	34	IFG0552B	78	
EOV Tape Output Label Verification Function	34	IFG0551X	78	
		IFG0551Z	78	
		IGG0550P	78	IFG0551X
EOV Tape Output New Volume Function	33	IFG0551V	77	
EOV Tape Output Trailer Label Function (EOV1)	33	IFG0551P	77	
EOV Tape Output Trailer Label Function (EOV2)	33	IFG0551R	77	
EOV Tape Output User Label Interface Function	33	IFG0551R	77	
EOV Tape Output Volume Disposition Function	33	IFG0551T	77	
		IGG0550F	77	IFG0551T
EOV Tape Output WTO Function	None	IFG0552P	80	
FEOV Executor Function	None	IGC0003A	71	
SYNAD Executor Function	None	IFG0551B	71	
		IFG0551D	71	

Common Functions

Function Name	Method of Operation Figure	Module Name	Program Organization Figure	Alias Name For
ABEND Interpretation and Recovery ABEND Exit Function	42	IFG0199E	90	
		IFG0209E	90	IFG0199E
		IFG0239E	90	IFG0199E
		IFG0559E	90	IFG0199E
ABEND Interpretation and Recovery ABEND Trace Function	42	IFG0199D	90	
		IFG0209D	90	IFG0199D
		IFG0239D	90	IFG0199D
		IFG0559D	90	IFG0199D
ABEND Interpretation and Recovery Initialization Function	42	IFG0190P	90	
		IFG0200P	90	IFG0190P
		IFG0230P	90	IFG0190P
		IFG0550P	90	IFG0190P
		IGG0196M	90	IFG0190P
ABEND Interpretation and Recovery Write-to-Programmer Function	42	IFG0199B	90	
		IFG0209B	90	IFG0199B
		IFG0239B	90	IFG0199B
		IFG0550B	90	IFG0199B
Display DSNAME WTO Function	None	IFG0190R	89	
		IFG0200R	89	IFG0190R
		IFG0550R	89	IFG0190R
Optional Module Trace Initialization/Termination Function	43	IFG0199R	91	
		IFG0209R	91	IFG0199R
		IFG0239R	91	IFG0199R
		IFG0559R	91	IFG0199R
Optional Module Trace Function	43	IFG019TR	91	
		IFG020TR	91	IFG019TR
		IFG023TR	91	IFG019TR
		IFG055TR	91	IFG019TR
Resident Routine Function	None	IFG019RA	88	

SECTION 5: DATA AREAS

O/C/EOV Work Area

Each time a DCB is to be processed by an Open, Close, or EOVS operation, a work area from the dynamic area of the user's region of main storage is obtained for the DCB by the operation in control. When the operation that obtained the work area has finished processing that DCB, it frees the DCB's work area.

The Open, Close, and EOVS operations request the storage for each DCB's work area from subpool 0. With MVT, however, storage requested by a program operating in supervisor state is assigned from subpool 252 in the region allocated to the processing program.

Subpool 252 is owned by the job step task and is shared by its subtasks. If a task other than the job step task abnormally terminates during one of the three operations, the work areas for each DCB obtained for that operation are not released, but remain allocated within subpool 252. More information about the subpools is in *OS Supervisor Services and Macro Instructions*, GC28-6646.

The length of the work area depends on the operation being performed. The first 488 bytes of the work area are used by all three operations; the EOVS operation uses an additional 48 bytes. To define the symbolic names of the fields used in the first 464 bytes of the work area, the three operations use the IECDSECT macro instruction.

Many of the fields in the O/C/EOVS work area are copies of fields in the JFCB, DSCB, and volume label that are associated with the data set being processed. Other fields in the work area (all of which are constructed by the Open, Close, and EOVS operations to read in the JFCB, DSCB, and volume label, and to position tapes) are copies of fields in the DEB, IOB, and CCWs.

Figures 92, 93, and 94 list the fields of the control blocks and labels that are stored in the O/C/EOVS work area during the processing of each DCB. The location relative to the beginning of the work area is given in decimal with the hexadecimal equivalent in parentheses (work area + 0 in Figure 92 through work area + 218 in Figure 94). The length in bytes of each field is given in decimal, in parentheses, with the field name.

Figure 94 includes a description of the fields listed. These fields are not associated with the control block fields and label information as are those in Figures 92 and 93. The fields in Figure 94 make up a work area that is used to manipulate other fields in the DCB work area. They are used to indicate certain conditions during the processing of the DCB.

More information about the control block fields is in *OS System Control Blocks*, GC28-6628.

Offset	Volume Label Fields	File Label 1 Fields	File Label 2 Fields	Identifier (Format 1) DSCB Fields	Extension (Format 3) Key DSCB Fields	Extension (Format 3) Record DSCB Fields	Message Area Fields
00 (00)	VOLLABI (3)	FL1LABI (3)		DSCFMTID (1)	DSCBF3C (4)	DSCBFMID (1)	REPLY (1)
01 (01)				DSCFILSR (6)		DSCBEX5 (10)	REPLYADR (3)
03 (03)	VOLNO (1)	FL1NO (1)					
04 (04)	VOLSERNO (6)	FL1ID (17)	FL2RECFM (1)		DSCBEXTY (1)		REPLYECB (4)
05 (05)			FL2BLKL (5)		DSCBXSQ (1)		
06 (06)					DSCBLLMT (4)		
07 (07)				DSVOLSR (2)			
08 (08)							MSGLSTSZ (2)
09 (09)				DSCCREDIT (3)			
10 (0A)	VOLSEC (1)		FL2LRECL (5)		DSCBULMT (4)		MCSFLAGS (2)
11 (0B)	VOLVTOC (5)					DSCBEX6 (10)	
12 (0C)				DSCXPDT (3)			MSGIOSUP (3)
14 (0E)					DSCBEX2 (10)		
15 (0F)			FL2DEN (1)	DSCNOEXT (1)			MSGSER (3) MSGINSTR (6)
16 (10)				DSCBLDBL (1)			
17 (11)			FL2JOB (8)				MSGSERLO (1)
18 (12)				DSCSYSCD (13)			
20 (14)							MSGACTN (1)
21 (15)		FL1FILSR (6)				DSCBEX7 (10)	
22 (16)							MSGUN (3)
24 (18)					DSCBEX3 (10)		

Figure 92 (Part 1 of 3). Open, Close, and EOVS operations obtain an O/C/EOV work area for each DCB to be processed.

Offset	Volume Label Fields	File Label 1 Fields	File Label 2 Fields	Identifier (Format 1) DSCB Fields	Extension (Format 3) Key DSCB Fields	Extension (Format 3) Record DSCB Fields	Message Area Fields
25 (19)			FL2JSSP (1)				
26 (1A)			FL2STEPD (8)				MSGVOLSR (6)
27 (1B)		FL1VOLSQ (4)					
31 (1F)		FL1FILSQ (4)				DSCBEX8 (10)	
33 (21)							MSGTEXT (47)
34 (22)			FL2TRTCH (2)		DSCBEX4 (10)		
35 (23)		FL1GNO (4)					
36 (24)			FL2CNTRL (1)				
38 (26)			FL2BLKA (1)	DSCFILTY (2)			
39 (27)		FL1VNG (2)	FL2RES (41)				
40 (28)				DSCRECFM (1)			
41 (29)	VOLOWNER (10)	FL1CREDIT (6)		DSCOPTCD (1)		DSCBEX9 (10)	
42 (2A)				DSCBLKL (2)			
44 (2C)				DSCRECL (2)			
46 (2E)				DSCKEYL (1)			
47 (2F)		FL1EXPDT (6)		DSCRKP (2)			
49 (31)				DSCSIND (1)			
50 (32)				DSCSCALO (4)			
51 (33)						DSCBEXA (10)	
53 (35)		FL1FSEC (1)					
54 (36)		FL1BLKCT (6)		DSCLSTAR (5)			

Figure 92 (Part 2 of 3). Open, Close, and EOVS operations obtain an O/C/EOV work area for each DCB to be processed.

Offset	Volume Label Fields	File Label 1 Fields	File Label 2 Fields	Identifier (Format 1) DSCB Fields	Extension (Format 3) Key DSCB Fields	Extension (Format 3) Record DSCB Fields	Message Area Fields
59 (3B)				DSCTRBAL (2)			
60 (3C)		FL1SYSCD (13)					
61 (3D)				DSCEXTYP (1)		DSCBEXB (10)	
63 (3F)				DSCLOWLM (4)			
67 (43)				DSCUPPLM (4)			
71 (47)				DSCEXT1 (10)		DSCBEXC (10)	
73 (49)		FL1RES (7)					
74 (4A)		FL1RES1 (6)					
80 (50)							DSCODE (2)
81 (51)				DSCEXT2 (10)		DSCBEXD (10)	
82 (52)							ROUTCODE (2)
84 (54)							REPLY (12)
91 (5B)				DSCNEXT (5)		DSCBNEXT (5)	
96 (60)				DSCCORE (4)			

Figure 92 (Part 3 of 3). Open, Close, and EOVS operations obtain an O/C/EOV work area for each DCB to be processed.

Offset	JFCB Fields	ECB Fields	IOB Fields	DEB/DCB Fields	CCW Fields
100 (64)	JFCBDSNM (44)				
144 (90)	JFCBELNM (8)				
152 (98)	JFCBTSDM (1)				
153 (99)	JFCBSYSC (13)				
166 (A6)	JFCBLTYP (1)				
167 (A7)	JFCBOTTR (3) JFCBUFOF (1)				
168 (A8)	JFCBFLSQ (2)				
170 (AA)	JFCBVLSQ (2)				
172 (AC)	JFCBMASK (8)				
180 (B4)	JFCBCRDT (3)				
183 (B7)	JFCBXPDT (3)				
186 (BA)	JFCBIND1 (1)				
187 (BB)	JFCBIND2 (1)				
188 (BC)	JFCBUFNO (1) JFCBUFRQ (1)				
189 (BD)	JFCBFTEK (1) JFCBFALN (1)				
190 (BE)	JFCBUFL (2)				
192 (C0)	JFCEROPT (1)				
193 (C1)	JFCRTTCH (1) JFCKEYLE (1) JFCMODE (1) JFCCODE (1) JFCSTACK (1) JFCPRTSP (1)				
194 (C2)	JFCDEN (1)				
195 (C3)	JFCLINCT (3)				
198 (C6)	JFCDSORG (2)				
200 (C8)	JFCRECFM (1)				
201 (C9)	JFCOPTCD (1)				
202 (CA)	JFCBLKSI (2)				
204 (CC)	JFCLRECL (2)				
206 (CE)	JFCNCP (1)				
207 (CF)	JFCNTM (1)				
208 (D0)	JFCRKP (2)				
210 (D2)	JFCCYLOF (1)				

Figure 93 (Part 1 of 3). Some fields of the O/C/EOV work area are associated with the JFCB, ECB, IOB, DEB, and CCWs.

Offset	JFCB Fields	ECB Fields	IOB Fields	DEB/DCB Fields	CCW Fields
211 (D3)	JFCBUFN (1)				
212 (D4)	JFCINTVL (1)				
213 (D5)	JFCCPRI (1)				
214 (D6)	JFCSOWA (2)				
216 (D8)	JFCBNTCS (1)				
217 (D9)	JFCBNVOL (1)				
218 (DA)	JFCBVOLS (30)				
248 (F8)	JFCBEXTL (1)				
249 (F9)	JFCBEXAD (3)				
252 (FC)	JFCBPQTY (3)				
255 (FF)	JFCBCTRI (1)				
256 (100)	JFCBSQTY (3)				
259 (103)	JFCBIND3 (1)				
260 (104)	JFCBDQTY (3)				
263 (107)	JFCBSPNM (3)				
266 (10A)	JFCBABST (2)				
268 (10C)	JFCBSBNM (3)				
271 (10F)	JFCBDRLH (3)				
274 (112)	JFCBVLCT (1)				
275 (113)	JFCBSPTN				
276 (114)		DXECB (4)			
280 (118)			DXIOB (32) IOBFLAG1 (1)		
281 (119)			IOBFLAG2 (1)		
282 (11A)			IOBSENSE (2) IOBSENS0 (1)		
283 (11B)			IOBSENS1 (1)		
284 (11C)			IOBECBPT (1)		
287 (120)			IOBCSW (8) IOBCOMAD (4)		
291 (124)			IOBSTAT0 (1)		
292 (125)			IOBSTAT1 (1)		
293 (126)			IOBCNT (2)		
295 (128)			IOBSIOCC (1)		

Figure 93 (Part 2 of 3). Some fields of the O/C/EOV work area are associated with the JFCB, ECB, IOB, DEB, and CCWs

Offset	JFCB Fields	ECB Fields	IOB Fields	DEB/DCB Fields	CCW Fields
296 (129)			IOBSTART (3)		
299 (12C)			IOBWGHT (1)		
300 (12D)			IOBDCBPT (3)		
307 (134)			IOBINCAM (2)		
309 (136)			IOBERRCT (2)		
311 (138)			DXDAADDR (8)		
315 (13C)				DXDEB (48)	
319 (140)				DXDEBDEB (4) DXDCB (48)	
323 (144)				DXDEBOFL (1) DXDEBIRB (4)	
327 (148)				DXDEBSYS (4)	
331 (14C)				DXDEBUSR (4)	
335 (150)				DXDEBECB (4)	
339 (154)				DXDEBDCB (4)	
343 (158)				DXDEBAPP (4)	
347 (15C)				DXDEBMOD (1) DXDEBUCEB (4)	
351 (160)				DXDEBBIN (2)	
353 (162)				DXDEBSCC (2)	
355 (164)				DXDEBSHH (2)	
357 (166)				DXDEBECC (2)	
359 (168)				DXDEBEHH (2)	
361 (16A)				DXDEBNTR (2)	
363 (16C)				DXDCBDEB (4)	
367 (170)					DXCCW (96)

Figure 93 (Part 3 of 3). Some fields of the O/C/EOV work area are associated with the JFCB, ECB, IOB, DEB, and CCWs.

Offset	Field Name	Field Use
456 (190)	DXDSNAME (17)	Save area for the data set name, taken from a tape data set label. This field is used when the data set name for a message is not obtained from the JFCB, in which case the message overlays the name field of the data set label.
480 (1A8)	DXREGSAV (32)	Register save area which has subfields DXREG9, DXREGA through DXREGE, DXREG0, and DXREG1. It is used by the resident routine to save the contents of registers 9 through 14, 0, and 1 (the registers that pertain to the DCB being processed when the resident routine is entered).
512 (1C8)	DXRESIND (1)	Contains resident routine indicators.
513 (1C9)	DXIDENT (7)	Includes the following two fields (DXCALLID and DXRETMOD).
513 (1C9)	DXCALLID (2)	Contains ID of the module that called ABEND interpretation and recovery.
515 (1CB)	DXRETMOD (5)	Includes the following two fields (DXRETID and DXRETTTR).
515 (1CB)	DXRETID (2)	Contains ID of the module to which ABEND interpretation and recovery will transfer control if the user has indicated that an error condition is to be ignored. This field may also contain the ID of the module to which a common module will transfer control. 1 Resident routine has been requested to issue WAIT macro for I/O started during processing of this DCB, but WAIT has not yet been issued. . 1 Processing of this DCB has reached a synchronization point in a module. . . x x x x x x Reserved bits.
516 (1CD)	DXRETTTR (3)	Contains the TTR (on SYS1.SVCLIB) of the module described by DXRETID.
516 (1CD)	DXABCODE (2)	Contains the left-justified 12-bit ABEND code to be issued by ABEND interpretation and recovery; overlays the first 2 bytes of DXRETTTR.
518 (1CF)	DXRETCODE (2)	Contains a return code issued by ABEND interpretation and recovery to accompany the ABEND code in DXABCODE; overlays the last byte of DXRETTTR.
519 (1D0)	DXXCTL (8)	XCTL work area. First word is address of module name at start of WTG table.
<i>The fields described above are common to Open, Close, and EOJ. The following fields are for use by Open and Close.</i>		
527 (1D8)	DXVOLMT1 (6)	Contains the volume serial number of the volume to be mounted by the Open operation.
533 (1DE)	DXVOLMT2 (6)	Volume serial number save area for the Open operation.
533 (1DE)	DXVOLSEQ (6)	Open work area that overlays the first 2 bytes of DXVOLMT2.
535 (1E0)	DXWORK1 (4)	Open work area that overlays the last 4 bytes of DXVOLMT2.
539 (1E4)	DXWORK2 (4)	Open work area.
<i>The following fields are used only by EOJ and follow field DXXCTL.</i>		
527 (1D8)	DXXAREA (40)	Contains a WTG table for EOJ that consists of a single entry and is nearly identical to the WTG table for Open and Close. The two tables have identical field names with the exception of the first 3 characters, which are either DXX or WTG. One field of DXXAREA, DXOPCLSW, has no corresponding field in WTGAREA.
567 (1F5)	DXOPCLSW (1)	See DXXAREA above. This field overlays the last byte of DXXAREA, which is not required by EOJ. EOJ's work area is of a fixed length. This field is used as a switch between EOJ and Open or Close.
577 (200)	DXVOLSER (18)	Includes the following three fields (DXVOLSR1, DXVOLSR2, and DXVOLSR3).

Figure 94 (Part 1 of 2). The fields beginning at offset X'190' in the O/C/EOV work area make up a work area that is used to manipulate other fields in the O/C/EOV work area.

Offset	Field Name	Field Use
577 (200)	DXVOLSR1 (6)	Contains the serial number of the volume which is at EOVS for read forward and output; contains the serial number of the mount-ahead volume for read backward.
583 (206)	DXVOLSR2 (6)	Contains the serial number of the next volume required.
589 (20C)	DXVOLSR3 (6)	Contains the serial number of the mount-ahead volume for read forward and output; contains the serial number of the volume at EOVS for read backward.
595 (212)	DXNOUNIT (2)	Contains the number of units allocated to the data set being processed.
597 (214)	DXDCBLST (4)	Contains a dummy DCB parameter list constructed by the EOVS Initial Work Area Initialization Function so that the resident routine can communicate with EOVS as it does with Open and Close. The parameter list is 1 word long because only one DCB at a time is processed by EOVS. The 3 low-order bytes contain the address of the DCB, and the high-order bit of the first byte is on to indicate that this is the last (and only) entry in the list.

Figure 94 (Part 2 of 2). The fields beginning at offset X'190' in the O/C/EOVS work area make up a work area that is used to manipulate other fields in the O/C/EOVS work area.

JFCBMASK Field

A JFCB is constructed and written on auxiliary storage by the job management routines for each dd name specified in a job step. A JFCB is brought into main storage when a DCB with the corresponding dd name is to be processed.

The JFCBMASK field, located in the JFCB, is used by the Open, Close, and EOVS operations to indicate DCB field modification 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 are modified during the Open operation. One bit corresponds to one DCB field (or more than one field if the fields are related). If a field is to be modified, Open sets its bit to 1. At the end of the Open operation, the Open final JFCB to DSCB merge function stores these 4 bytes of the JFCBMASK field in the DEBDCBMK field of the DEB and sets these 4 bytes of the JFCBMASK field to 0 before writing the JFCB back to its original location. See the "DEBDCBMK Field" section for more information on this field.

The Open, Close, and EOVS operations use the remaining 4 bytes of the JFCBMASK field to flag certain conditions necessary to process the DCB.

The first three columns of the following list show each byte of the JFCBMASK field, each bit within a byte, and what the bit indicates when set to 1. The fourth column shows which operations use the bit.

<u>Byte</u>	<u>Bit</u>	<u>Indicates</u>	<u>Operation(s)</u>
JFCBMASK+0	xxxx xxxx	Reserved for future use.	
JFCBMASK+1	1	DCBOPTCD field modified to force 2321 write checking.	Open
	. 1	DCBUFOF field merged from JFCBUFOF field of the JFCB for BSAM or QSAM.	Open
	. . 1	DCBNTM field modified for ISAM. DCBPCI field modified for TCAM.	Open Open

<u>Byte</u>	<u>Bit</u>	<u>Indicates</u>	<u>Operation(s)</u>
JFCBMASK+1 (Continued)	. . . 1 DCBCYLOF field modified for ISAM.	Open
	1 . . . DCBRKP field modified for ISAM.	Open
 DCBRESER field modified for TCAM.	Open
 1 . . DCBDBUFN field modified for ISAM.	Open
 1 . DCBNCP field modified for ISAM.	Open
 1 DCBEROPT field modified for ISAM or QSAM.	Open
JFCBMASK+2	1 DCBLRECL field modified for ISAM or QSAM.	Open
	. 1 DCBLIMCT field modified for BDAM.	Open
	. . 1 DCBNCP field modified for BSAM.	Open
	. . . 1 DCBBLKSI and DCBBUFSI fields modified for TCAM.	Open
	1 . . . DCBOPTCD field modified.	Open
 1 . . DCBRECFM field merged from JFCRECFM field of the JFCB.	Open
 1 . DCBDEN field modified.	Open
 1 DCBFTEK and DCBBFALN fields modified, and DCBGNCP field merged from JFCNCP field of the JFCB. DCBBUFMA field modified for TCAM.	Open Open
JFCBMASK+3	1 DCBBUFL field modified.	Open
	. 1 DCBBUFNO field modified.	Open
	. . 1 DCBKEYLE field merged from JFCBKEYLE field in the JFCB, and DCBTRTCH field modified.	Open
	. . . 1 DCBSOWA field merged from JFCSOWA field of the JFCB for QTAM.	Open
	1 . . . DCBCPRI field merged from JFCCPRI field of the JFCB for QTAM. DCBTHRES field modified for TCAM.	Open Open
 1 . . DCBINTVL field merged from JFCINTVL field in the JFCB for QTAM.	Open
 1 . DCBLRECL field merged from JFCLRECL field in the JFCB.	Open
 1 DCBDSORG field merged from JFCDSORG field of the JFCB.	Open
JFCBMASK+4	1 JFCB has been modified and is to be written back, unless bit 7 of JFCBTSDM field is set to 1. JFCB has been read.	Open Close

<u>Byte</u>	<u>Bit</u>	<u>Indicates</u>	<u>Operation(s)</u>
JFCBMASK+4 (Continued)	0	JFCB has not been modified; however, the Open operation assumes, at the end of its processing, that the bit is set to 1 and writes the JFCB back.	Open
	. 1	DSCB has been modified and is to be written back.	Open
	. . 1	Data set being processed is a dummy (null) data set.	Open
	. . . 1	BPAM concatenation.	Open
 1	Direct access parallel mounting for ISAM or BDAM.	Open
 1	Password required.	Open
 1	Delayed ABEND pending (DCBOFLGS field bit 7 set to 0), or the processing program recovered from an ABEND (DCBOFLGS field bit 7 set to 1).	Open, Close, EOVS
 1	Volume sequence number modified by DISP=MOD direct-access recovery volume sequence number set to 0 after DEB built.	Open
	JFCBMASK+5	1	Output magnetic tape label conflict.
. 1		IBM standard or American National Standard labeled tape required for output magnetic tape.	
. . 1		Nonstandard or nonlabeled tape required for output magnetic tape.	Open, EOVS
. . . 1		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	Open, EOVS
. . . . 1		Merge during standard label or nonstandard label processing to be bypassed after a remount of output magnetic tape.	Open
. 00		User-label routine called for Close operation with TYPE=T.	Open, EOVS
. 10		User-label routine called for Open operation.	Open, EOVS
. 11		User-label routine called for EOVS operation.	Open, EOVS
. x		Reserved for future use.	

<u>Byte</u>	<u>Bit</u>	<u>Indicates</u>	<u>Operation(s)</u>
JFCBMASK+6	1	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.	Open
	. 1	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.	Open
Note: Both bits 0 and 1 are set to 1 during the Open operation if the mode is being overridden.			
	. . 0	Direct-access rotational position sensing (RPS) is being used.	Open
	. . 1	Direct-access RPS is not being used.	Open
	. . . 1	DISP=NEW is to be changed to DISP=MOD.	Open
 0	Search direct is not being used.	Open
 1	Search direct is being used.	Open
 1	Work areas associated with this DCB are to be traced by the optional GTF routine each time a branch is made to the resident routine for an XCTL macro to be issued. The bit is set to 1 by the job scheduler if DCB=DIAGNS=TRACE is specified on the DD statement.	Open, Close, EOV
 0	Open operation, after it was called, stored a TTR in the JFCBOTTR field.	Open
 1	JFCBUFOF field contains, before the Open operation has been called, a buffer offset or invalid information resulting from a JFCB-to-JFCB merge.	Open
 1	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.	Open
JFCBMASK+7	1	Second pass through the Open Tape Volume Reference Function is necessary.	Open
	. xxx	Reserved for future use.	

DEBDCBMK Field

The DEB is an extension of the information in the DCB. Each DEB is associated with a DCB, and the two blocks contain pointers to each other. The DEB contains information about the physical characteristics of the data set and other information that is being used by the control program.

The DEBDCBMK field, located in the prefix section of the DEB, is a DCB modification mask, which is built by Open to indicate which DCB fields are modified during its operation. One bit corresponds to one or more DCB fields. If a DCB field is to be modified, Open sets its bit to 1. The mask is then stored in the DEB by the Open operation.

The Close operation examines the DEBDCBMK field to determine which fields of the DCB are to be restored to the status they had before they were processed by the Open operation. (Close restores the fields whose corresponding bits are set to 1.)

The three columns of the following list show each byte of the DEBDCBMK field, each bit within a byte, and which fields in the DCB may be modified.

<u>Byte</u>	<u>Bit</u>		<u>Field(s) Modified</u>
DEBDCBMK+0	xxxx	xxxx	Not used
DEBDCBMK+1	x	Not used
	. 1	DCBIFLG
	. . 1	DCBNTM, DCBPCI
	. . . 1	DCBCYLOF
	1 . . .	DCBRKP, DCBRESER
 1 . .	DCBDBUFN
 1 .	DCBNCP (ISAM)
DEBDCBMK+2 1	DCBEROPT
	1	DCBLRECL
	. 1	DCBLIMCT
	. . 1	DCBNCP (BSAM)
	. . . 1	DCBBLKSI, DCBBUFSI
	1 . . .	DCBOPTCD
 1 . .	DCBREFCM
DEBDCBMK+3 1 .	DCBDEN
 1	DCBFTEK, DCBBFALN, DCBGNCP, DCBBUFMA
	1	DCBBUFL
	. 1	DCBBUFNO
	. . 1	DCBKEYLE, DCBTRTCH
	. . . 1	DCBSOWA
	1 . . .	DCBCPRI, DCBTHRES
DEBDCBMK+3 1 . .	DCBINTVL
 1 .	DCBLRECL
 1	DCBDSORG

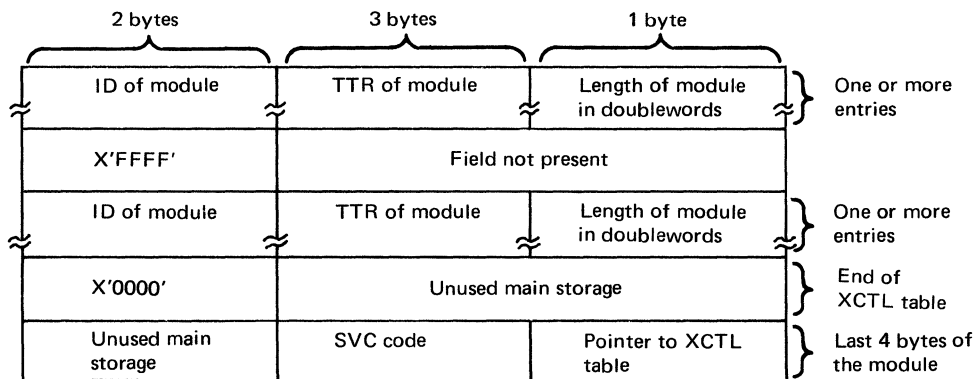
DCBOFLGS Field

This 1-byte field is located in the foundation segment of the DCB. The field contains input and output information necessary for the Open, Close, and EOVS operations to process the DCB. More information about this field and the DCB is in *OS System Control Blocks*, GC28-6628.

<u>Bit</u>	<u>Meaning</u>
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 EOVS operation took a user exit and the user's processing has not been completed. (This bit setting prevents other Open, Close, and EOVS operations from processing this DCB.)
. 1	An Open, Close, or EOVS operation took a user exit and the user's processing has been completed.
. 1	This DCB is to be processed by the Open, Close, or EOVS operation that set this bit to 1.

Transfer Control (XCTL) Table

If a module name consists of IFG or IGG, followed by 019, 020, 023, or 055, followed by a numeric character, an XCTL table appears in the module. (IGC0001I, IGC0002B, IGC00020, IGC0002C, IGC0003A, and IGC0005E also have XCTL tables.) The XCTL table in each such module lists the other modules to which that module can transfer control. The format of the table is shown in Figure 95.



Note: IDs before X'FFFF' indicate modules with names beginning with "IFG." IDs after X'FFFF' indicate modules with names beginning with "IGG."

Figure 95. Format of the XCTL table.

The XCTL table starts on a doubleword boundary and consists of entries, each in the form IDTTRL, for the other modules to which control can be transferred. Each entry consists of a module ID, the module's relative disk address (TTR) in SYS1.SVCLIB, and the module's length (L) in doublewords. The TTRLs are inserted by the IEHIOSUP utility program when the system is generated or modified.

The last 4 bytes of each module consist of a 3-byte EBCDIC supervisor call (SVC) code and a 1-byte pointer to the module's XCTL table. The SVC code is the fourth, fifth, and sixth letters of the module name.

When a module has finished its processing for a DCB, the ID of the next module to gain control is put in the WTG table entry and a branch is taken to the resident routine, which issues the XCTL macro.

Where-To-Go (WTG) Table

Each Open or Close operation sets up its own WTG table to indicate the sequence of modules needed to complete the operation. Figure 96 shows the WTG table format.

The first 29 bytes (bytes 0-28) of the WTG table consist of a parameter list for the XCTL and LOAD macro instructions. Bytes 30 and 31 indicate by bit setting (see the "WTGPATH Field" section for the bit settings) the sequence of modules needed to complete the Open or Close operation.

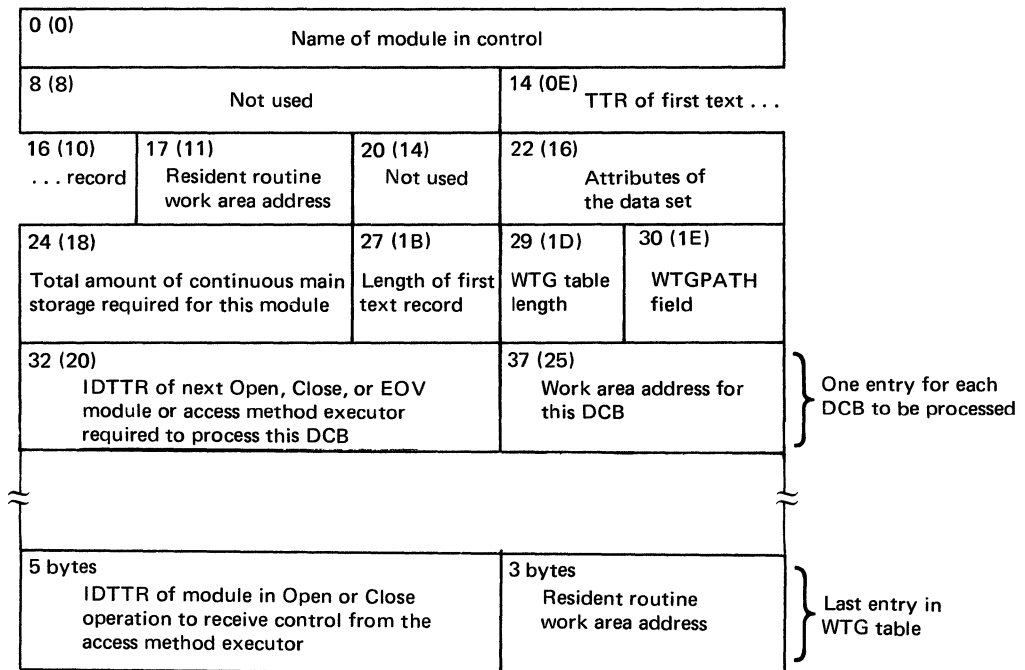


Figure 96. Each Open or Close operation sets up its own WTG table.

The length of the rest of the WTG table depends on the number of DCBs to be processed. This part of the table indicates the next module required to process each DCB. Open, Close, and EOV operations indicate this by providing each DCB with an entry in the table when it is built. The Open, Close, and EOV operations issue the IECRES macro instruction, which copies from the XCTL table to the first 5 bytes of the DCB's 8-byte entry the ID and TTR of the next Open, Close, or EOV module or first access method executor required to process the DCB. The last 3 bytes of the 8-byte entry contain the address of the work area assigned to the DCB.

The last 8-byte entry of the WTG table contains the IDTTR (5 bytes) of the module in the Open or Close operation to which the access method executors return control and the address (3 bytes) of the resident routine work area.

All of the 8-byte DCB entries except the last are used by both the resident routine and the access method executors. The last entry is used by the access method executors only.

Dumps issued by systems with MFT contain only the last 4 characters of the module name in the Request Block (RB) APSW field. After an OPEN macro instruction has been issued, the Open WTG table should be checked for the module name. Possible duplications may occur in this field because of modules with names of the form IFG0xxxx or IGG0xxxx.

WTGPATH Field

The WTG table contains a field called WTGPATH. The settings of bits in this field are used to pass information between modules processing DCBs in parallel.

The first three columns of the following list show each byte of the WTGPATH field, each bit setting within a byte, and what the bit indicates when set to 1. The fourth column shows which operations use the bit.

<u>Byte</u>	<u>Bit</u>	<u>Indicates</u>	<u>Operation</u>
WTGPATH+0	1	Direct-access partial release was requested.	Close
	. 1	Nonstandard label tape input processing requires the display DSNAME WTO function.	Open
	. . 1	NSL routine is needed. NSL processing is deferred.	TCLOSE EOV
	. . . 1	SMF data set information is required.	Close, EOV
 xxxx	Reserved for future use.	
WTGPATH+1	1	Optional GTF routine is loaded or is about to be loaded.	Open, Close, EOV
	. xxx . xx . .	Reserved for future use.	
 1 .	Nonstandard label volume must be remounted and checked.	Open
 x	Reserved for future use.	Open

SECTION 6: DIAGNOSTIC AIDS

ABEND-Function-Module Cross-Reference

This cross-reference table lists the ABEND codes and return codes that are issued by the Open, Close, and EOVS operations and by the access method executors to which control is given during DCB processing.

The list is in hexadecimal order by ABEND code with the name of the function (or access method executor) issuing the ABEND and return codes listed beside the return code. The names of specific modules issuing the ABEND and return codes are listed beside the function names.

To find more information about a particular function and module, refer to the "Function-to-Module Directory," which indicates where descriptions of the function and module appear.

Information about the access method executors is in the logic manual for the access method being used.

ABEND	Return Code	Function	Module
013	04	Open Tape Initial Common Function	IFG0193B
	08	Open Tape Initial Common Function	IFG0193B
	0C	BDAM Executor	IGG0193E
	1C	SAM Executor	IGG0191B
	10	SAM Executor	IGG0191C
	14	SAM Executor	IGG0191B
	18	SAM Executor	IGG0191B
	20	SAM Executor	IGG0191A
	24	SAM Executor	IGG0191A
	28	SAM Executor	IGG0191A
	2C	SAM Executor	IGG0191E
	30	SAM Executor	IGG0191F
	34	SAM Executor	IGG0191I
	38	SAM Executor	IGG0191H
	3C	SAM Executor	IGG0191D
	40	SAM Executor	IGG01910
	44	SAM Executor	IGG0191K
	48	SAM Executor	IGG01911
	4C	SAM Executor	IGG0196B
	50	SAM Executor	IGG0196B
54	SAM Executor	IGG0196A	

ABEND	Return Code	Function	Module
013 (Continued)	58	SAM Executor	IGG01912
	5C	SAM Executor	IGG01915
		SAM Executor	IGG01916
	60	SAM Executor	IGG0191A
	64	Open Merge DCB To JFCB Function	IFG0196M
		Open BPAM Concatenation Function	IFG0195M
	68	Open Merge DCB Exit Function	IFG0196L
	6C	Open Merge DCB Exit Function	IFG0196L
	70	Open Merge DCB To JFCB Function	IFG0196M
	74	SAM Executor	IGG0197A
	78	SAM Executor	IGG0197A
	7C	SAM Executor	IGG0197A
	80	SAM Executor	IGG0197A
	84	SAM Executor	IGG0197B
03D	04	Open Direct-Access Parallel Mounting Function	IFG0195O, IFG0195P
	08	Open Direct-Access Parallel Mounting Function	IFG0195O, IFG0195P
113	04	Open Initial Volume Serial Function	IFG0193A
		Open Tape Volume Reference Function	IFG0194D
	08	Open Initial Volume Serial Function	IFG0193A
		Open Direct-Access DISP=MOD Error Recovery Function	IFG0195E
	0C	Open Initial Volume Serial Function	IFG0193A
	10	Open Final Write JFCB Function	IFG0198N
	14	Open Direct-Access Parallel Mounting Function	IFG0195O, IFG0195P
	18	Open Direct-Access Parallel Mounting Function	IFG0195P
117	04	TCLOSE Final Function	IFG0232Z (IFG0202U)
	08	TCLOSE Tape Volume Positioning Function	IFG0232S
	0C	TCLOSE Tape Volume Positioning Function	IFG0232S
	10	TCLOSE Tape Volume Positioning Function	IFG0232S
	14	TCLOSE Tape Volume Positioning Function	IFG0232S
	18	TCLOSE Tape Volume Positioning Function	IFG0232S
	1C	TCLOSE Tape Volume Positioning Function	IFG0232S

ABEND	Return Code	Function	Module
117 (Continued)	20	TCLOSE Tape Volume Positioning Function	IFG0232S
	24	TCLOSE Tape Volume Positioning Function	IFG0232S
	28	TCLOSE Tape Volume Positioning Function	IFG0232S
	2C	TCLOSE Tape Standard Trailer Label Function	IFG0232G, IFG0232M
	30	TCLOSE Tape Standard Trailer Label Function	IFG0232G, IFG0232M
137	04	EOV Tape Output Trailer Label Function (EOV1)	IFG0551P
	08	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	0C	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	10	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	14	EOV Tape Input Standard Header Label Function	IFG0553F
	18	EOV Tape Input Standard Header Label Function	IFG0553F
	1C	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	20	EOV Tape Input Standard Header Label Function	IFG0553F
140	04	RDJFCB Macro Instruction	IGC0006D
213	04	Open Direct-Access 213 ABEND Exit Function	IFG0195G
	08	Open Security Initialization Function	IFG0195T
	0C	Open Direct-Access Parallel Mounting Function	IFG0195O, IFG0195P
	10	Open Direct-Access Parallel Mounting Function	IFG0195O, IFG0195P
	18	Open Final JFCB to DSCB Merge Function	IFG0196X
	1C	SAM Executor	IGG0197F
214	04	Close Tape Standard User Label Function	IFG0202A
	08	Close Tape Volume Disposition Function	IFG0202F, IFG0202G
217	04	TCLOSE Initial Function	IGC0002C
237	04	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	08	EOV Tape Input Standard Header Label Function	IFG0553F
240	04	RDJFCB Macro Instruction	IGC0006D
	08	RDJFCB Macro Instruction	IGC0006D
	0C	RDJFCB Macro Instruction	IGC0006D
	10	RDJFCB Macro Instruction	IGC0006D

ABEND	Return Code	Function	Module
313	04	Open Direct-Access Read DSCB Function	IFG0194E, IFG0195A
314	04	Close Direct-Access Read DSCB Function	IFG0200V
	08	Close Direct-Access Output User Labels Function	IFG0201R, IFG0202D
317	04	TCLOSE Direct-Access Input Function	IFG0232D
		TCLOSE Direct-Access Output Function	IFG0232D
337	04	EOV End-of-Data Function	IFG0552X
		EOV Access Method Executor Function	IFG0551L, IFG0551N
413	04	Open Direct-Access Volume Verification Function	IFG0194A, IFG0194C
		Open Tape Mount Verification Function	IFG0194F
	08	Open Tape Initial Common Function	IFG0193B
		Open Tape Volume Mounting Function	IFG0194G
		Open Tape Volume Verification Function	IFG0194H
		Open Tape Final Common Function	IFG0194I
		Open Tape Nonstandard Label Input Interface Function	IFG0195D (IGG0190B)
	0C	Open Tape Volume Verification Function	IFG0194H
		Open Tape Final Common Function	IFG0194I
		Open Tape Nonstandard Label Input Interface Function	IFG0195D (IGG0190B)
	14	Open Tape Final Common Function	IFG0194I
	18	Open Tape Volume Reference Function	IFG0194D
		Open Direct-Access Unit Selection Function	IFG0194E
	1C	Open Initial Volume Serial Function	IFG0193A
	20	Open Direct-Access Volume Verification Function	IFG0194A, IFG0194C
	24	Open Initial Function	IGC0001I, IGC0002B
414	04	Close Direct-Access Write DSCB Function	IFG0201R
417	04	TCLOSE Direct-Access Input Function	IFG0232D
		TCLOSE Direct-Access Output Function	IFG0232D
437	04	EOV Initial Read JFCB Function	IFG0551F
513	04	Open Tape Final Common Function	IFG0194I
613	04	Open Tape Standard Label Positioning Function	IFG0195B
		Open Tape No Label Positioning Function	IFG0195C

ABEND	Return Code	Function	Module
613 (Continued)	04	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195H
		Open Tape Standard Label Input/MOD Header Label 2 Function	IFG0195H, IFG0195K
		Open Tape Standard Label Input User Label Function	IFG0195N
		Open Tape Standard Label Output Security Function	IFG0195N
	08	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195H
		Open Tape Standard Label Input/MOD Header Label 2 Function	IFG0195H, IFG0195K
		Open Tape Standard Label Output Security Function	IFG0195N
	0C	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195H
		Open Tape Standard Label Input/MOD Header Label 2 Function	IFG0195H, IFG0195K
		Open Tape Standard Label Output Security Function	IFG0195N
	10	Open Tape Standard Label Rewrite Volume Label Function	IFG0195Q, IFG0195T
		Open Tape Standard Label Header Label Writing Function	IFG0195T
	14	Open Tape Standard Label Header Label Writing Function	IFG0195T
		Open Tape Standard Label Output User Label Function	IFG0196U
614	04	Close Direct-Access Write File Mark Function	IFG0202E
637	04	EOV Tape Output Volume Disposition Function	IFG0551T (IGG0550F)
		EOV Tape Output Label Verification Function	IFG0551X (IGG0550P), IFG0551Z
	08	EOV Tape Input Standard Trailer Label Function	IFG0552R, IFG0552T
	0C	EOV End-of-Data Function	IFG0552X
	10	EOV Tape Input Data Set Positioning Function	IFG0553D
	14	EOV Tape Input Volume Positioning Function	IFG0552V
	18	EOV Tape Input Volume Positioning Function	IFG0552V
	1C	EOV Tape Input Volume Positioning Function	IFG0552V
	20	EOV Tape Input Volume Positioning Function	IFG0552V
	24	EOV Tape Input Volume Positioning Function	IFG0552V
	28	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D

ABEND	Return Code	Function	Module
637 (Continued)	2C	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
	30	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
	34	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
	38	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
	3C	EOV Tape Input Data Set Positioning Function	IFG0553D
	40	EOV Tape Input Data Set Positioning Function	IFG0553D
	44	EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
	48	EOV Tape Input Next Volume Mounting Function	IFG0553H (IGG0550D)
	4C	EOV Tape Input Standard Header Label Function	IFG0553F
713	04	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195H
	08	Open Direct-Access Expiration Date Error Function	IFG0195G
714	04	Close Tape Standard Trailer Label Function	IFG0200Z
	08	Close Tape Standard Trailer Label Function	IFG0200Z
	0C	Close Tape Standard Trailer Label Function	IFG0200Z
		Close Tape Volume Disposition Function	IFG0202F, IFG0202G
717	04	TCLOSE Tape Standard Trailer Label Function	IFG0232G, IFG0232M
	08	TCLOSE Tape Standard Trailer Label Function	IFG0232G, IFG0232M
	0C	TCLOSE Tape Volume Positioning Function	IFG0232S
	10	TCLOSE Tape Volume Positioning Function	IFG0232S
737	04	EOV Direct-Access Output B37 ABEND Function	IFG0554T
		EOV Direct-Access Output Next Volume Mount Function	IFG0554V
		EOV Direct-Access Output IBM 2321 Next Volume Mount Function	IFG0554X
		EOV Direct-Access Output Volume Disposition Function	IFG0554Z
		EOV Direct-Access Output JFCB Update Function	IFG0555D
		EOV Direct-Access Output Read DSCB Function	IFG0555F
		EOV Direct-Access Output Current DEB Function	IFG0555H
	08	EOV Direct-Access Input Mount Function	IFG0553R, IFG0553T

ABEND	Return Code	Function	Module
737 (Continued)	08	EOV Direct-Access Output IBM 2321 Next Volume Mount Function	IFG0554X
		EOV Direct-Access Input FEOV Repositioning Function	IFG0554B, IFG0554D
	0C	EOV Direct-Access Input Exit Function	IFG0553Z
	14	EOV Direct-Access Input Initial Function	IFG0553P
	18	EOV Direct-Access Input Mount Function	IFG0553R, IFG0553T
		EOV Direct-Access Input FEOV Repositioning Function	IFG0554B, IFG0554D
	1C	EOV Direct-Access Input DEB Function	IFG0553X
	20	EOV Direct-Access Input Mount Ahead Function	IFG0553V
813	04	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195H
		Open Tape Standard Label Input/MOD Header Label 2 Function	IFG0195H, IFG0195K
837	04	EOV Initial Read JFCB Function	IFG0551F
		EOV Initial JFCB Extension Function	IFG0551J
		EOV Tape Output Exit Function	IFG0552H
	08	EOV Initial JFCB Extension Function	IFG0551J
913	04	Open Tape Volume Verification Function	IFG0194H
	08	Open Tape Standard Label Input/MOD Header Label 1 Function	IFG0195B, IFG0195K
		Open Tape Standard Label Output Security Function	IFG0196N
	0C	Open Security Search Function	IFG0195U
A13	04	Open Tape Standard Label Positioning Function	IFG0195B
		Open Tape No Label Positioning Function	IFG0195C
A14	04	Close Direct-Access Partial Release Interface Function	IFG0200Y (IGG0200F) (IGG0200G)
B13	04	SAM Executor	IGG0191U
	08	SAM Executor	IGG0191U
	0C	SAM Executor	IGG0191V
	10	SAM Executor	IGG0197U
	14	SAM Executor	IGG0197U
	18	SAM Executor	IGG0197F
	1C	SAM Executor	IGG0197F
	20	SAM Executor	IGG0191T

ABEND	Return Code	Function	Module
B13 (Continued)	24	SAM Executor	IGG0191T
	28	SAM Executor	IGG0191T
B14	04	SAM Executor	IGG0201B
		SAM Executor	IGG0201Z
	08	SAM Executor	IGG0201B
		SAM Executor	IGG0201Z
	0C	SAM Executor	IGG0201B
		SAM Executor	IGG0201Z
	10	SAM Executor	IGG0201B
		SAM Executor	IGG0201Z
B37	04	EOV Tape Output New Volume Function	IFG0551V
		EOV Tape Input New Volume Mounting Function	IFG0552Z, IFG0553D
		EOV Direct-Access Input Mount Function	IFG0553R, IFG0553T
		EOV Direct-Access Output IBM 2321 Get Space Current Volume Function	IFG0554R (IGG0552K)
		EOV Direct-Access Output B37 ABEND Function	IFG0554T
C13	04	Open Direct-Access BPAM Concatenation Function	IFG0195M
	08	Open Direct-Access BPAM Concatenation Function	IFG0195M
	0C	Open Direct-Access BPAM Concatenation Function	IFG0195M
	10	Open Direct-Access BPAM Concatenation Function	IFG0195M
	14	GAM Executor	IGG01934
D13	04	GAM Executor	IGG01934
D37	04	EOV Direct-Access Output Get Space Current Volume Function	IFG0554P
		EOV Direct-Access Output IBM 2321 Get Space Current Volume Function	IFG0554R (IGG0552K)
		EOV Direct-Access Output Next Volume Mount Function	IFG0554V
		EOV Direct-Access Output IBM 2321 Next Volume Mount Function	IFG0554X
E13	04	GAM Executor	IGG01934
E37	04	EOV Direct-Access Output Get Space Current Volume Function	IFG0554P
		EOV Direct-Access Output IBM 2321 Get Space Current Volume Function	IFG0554R (IGG0552K)
		EOV Direct-Access Output Next Volume Mount Function	IFG0554V

ABEND	Return Code	Function	Module
E37 (Continued)	04	EOV Direct-Access Output IBM 2321 Next Volume Mount Function	IFG0554X
		EOV Direct-Access Output Volume Disposition Function	IFG0554Z

DCB Error Handling

There are two situations in which a DCB to be processed by an Open, Close, or EOVS operation may be ignored.

1. The processing program has requested that a DCB be opened, but the DCB is already open and cannot be opened again.
2. Control is returned from problem determination with a request to ignore that DCB.

At four stages in processing a DCB the two situations are taken into consideration to ensure that each ignored DCB's work area is freed.

Control Given to First Module of the Operation

If no DCBs are to be processed, control is returned to the user.

For a DCB in situation 1:

- An O/C/EOVS work area is not obtained.
- The O/C/EOVS work area address for the DCB in the WTG table entry is set to 0.
- The module ID in the DCB's WTG table entry is set to X'0000'.

For a DCB in situation 2:

- Bit 7 of the DCBOFLGS field is set to 0.
- The IDTTR of the last module of the operation is inserted in the DCB's WTG table entry.
- A branch-table offset of 4 is added to the contents of register RET. This branch-table offset corresponds to the last entry of the WTG table.

Control Given to Executors

If bit 7 of the DCBOFLGS field is set to 0, the module ID in the DCB's WTG table entry is set to X'0000'.

Control Returned from Executors

If the module ID in the DCB's WTG table entry is X'0000', the work area pointer in the entry is not 0, and bit 7 of the DCBOFLGS field is set to 0, the ID in the last 8-byte entry of the WTG table is moved to the DCB's WTG table entry and a branch-table offset of 4 is added to the contents of register RET.

If bit 7 of the DCBOFLGS field is set to 1, the ID of the current module in the first entry of the WTG table is copied into the DCB's entry.

Control Given to Module Whose ID is in the Last 8-byte Entry of WTG Table

A branch is taken to the address specified in the branch-table at the offset in the last 8-byte entry to handle the DCB that was ignored because of the second situation.

Internal ABEND Codes

When an error that requires the ABEND Interpretation and Recovery Initialization Function occurs during an Open, Close, or EOF operation, the operation passes the function an internal code in register 0. This internal code indicates to the function what ABEND code and return code are to be issued.

The following list is divided into the Open, Close, Close (TYPE=T), and EOF operations with the specific internal codes that each passes listed under it. The list gives the internal code, its associated ABEND code and return code, and the reason for the error.

Open Operation

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
1	413	04	No unit available to mount volume
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
5	413	14	I/O error during writing of EOF1 or EOF2 on SYSOUT data set
6	513	04	Second OPEN was issued for a magnetic tape volume
7	113	04	I/O error during reading of JFCB, or JFCB does not exist
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	DS name 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 labels

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
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 security-protected volume
21	213	08	Password DSCB not found
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
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
29	113	08	I/O error during reading of JFCB extension block with DISP=MOD
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	I/O error during reading of JFCB extension block
33	113	04	I/O error during reading of JFCB
34	113	0C	OPEN TYPE=J issued but no JFCB exit specified
35	113	10	I/O error during writing back of JFCB
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
38	C13	0C	I/O error during reading of DSCB of concatenated partitioned organization (PO) data set
39	C13	08	I/O error during reading of JFCB of concatenated PO data set
40	C13	04	DSCB of concatenated PO data set not found
42	C13	10	Illegal concatenation of PO data sets for OUTPUT specified
43	013	64	Dummy (null) data set specified but invalid for EXCP or if DSORG, then PS data set use dummy data set for BSAM or QSAM
44	013	68	Blocksize is greater than 32767
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
49	113	14	I/O error during reading of ISAM concatenated JFCB

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
50	113	18	I/O error during reading of ISAM or BDAM JFCB extension block
51	013	0C	Dynamic buffering requested for BDAM data set with BUFL=0
52	013	10	No BUFL or BUFCB specified for a dummy (null) data set
53	013	14	DCB specified PO data set but DSCB specified different type
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
56	013	20	Blocksize not a multiple of logical record length for blocked data set with fixed-length records
57	013	24	OPEN option 1 conflicts with MACRF parameter in DCB
58	013	28	OPEN option 1 conflicts with MACRF parameter in DCB
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
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
70	B13	08	I/O error during execution of BLDL instruction for UCS image
71	B13	04	UCS image not found in SYS1.SVCLIB by BLDL instruction
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
78	140	04	I/O error during reading of JFCB
79	240	04	RDJFCB issued but no foundation block present in DCB
80	240	08	RDJFCB issued but no exit list address specified in DCB
81	240	0C	RDJFCB issued but no JFCB exit specified

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
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
84	013	84	Number of buffers in buffer pool control block not equal to DCBBUFNO
85	013	74	Optical character recognition (OCR) data set not open for INPUT
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 loading of forms control buffer for IBM 3211 printer
90	B13	1C	Operator replied 'CANCEL' to message IEC128I
91	C13	14	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	DCB does not specify TSO and requested unit is telecommunications device
95	B13	20	Storage not available for SYS1.IMAGELIB
96	B13	24	Volume containing SYS1.IMAGELIB not mounted
97	B13	28	I/O error during search of SYS1.IMAGELIB
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
217	213	10	I/O error during reading of extension (format 3) DSCB for BDAM or ISAM data set

Close Operation

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
61	714	04	I/O error during writing of trailer label 1
62	714	0C	I/O error during writing of tapemark
63	714	08	I/O error during writing of trailer label 2
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

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
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
111	514	04	I/O error during reading of JFCB
112	Not Applicable	Not Applicable	Invalid control block(s) found
120	D14	04	Graphic device opened by a task other than graphics task
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

Close (TYPE=T) Operation

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
121	217	04	I/O error during reading of JFCB
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 trailer label 2
125	717	0C	I/O error during writing of tapemark after trailer labels (SL) or after data (NL)
126	717	0C	I/O error during writing of 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 being 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 DSCB
133	417	04	I/O error during writing of updated DSCB
134	117	20	I/O error during positioning of tape volume for NL INPUT or SL OUTPUT data set

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
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 positioning of tape volume when RDBACK and LEAVE or INOUT and REREAD are specified
138	717	10	I/O error during reading of trailer label 1 to updated block count in DCB
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

EOV Operation

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
141	137	04	I/O error during writing of EOV label or tapemark
142	B37	04	New volume cannot be mounted because all units are reserved
143	837	04	I/O error during reading or writing of JFCB
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 or preparing for label processing
147	137	0C	I/O error during reading of trailer label or header label if data set opened for RDBACK
148	237	04	Block count in DCB does not match count in trailer label
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
154	637	20	I/O error during positioning of tape volume when OPEN option 2 is not specified, the volume is private, and DISP is not equal to PASS

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
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
157	637	28	I/O error during positioning of tape volume when wrong volume mounted
158	637	2C	I/O error during rewinding of next volume before verification of volume label
159	637	30	I/O error during unloading when wrong volume mounted in response to message IEC001A
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 concatenated data set
163	637	40	I/O error during positioning of SL or NL volume to data
164	137	14	I/O error during reading of header label or trailer label during opening for RDBACK
165	237	08	DS name in tape label does not match that in JFCB
166	B37	04	Next volume for IBM 2321 Data Cell Drive cannot be mounted because no unit available
167	737	18	I/O error during reading of volume label on IBM 2321 Data Cell Drive
168	B37	04	Next volume cannot be mounted because no unit available
169	737	08	I/O error during reading of volume label on direct-access volume
170	737	20	I/O error during reading of DSCB during a mount ahead of next volume
171	737	0C	I/O error during reading of DSCB for concatenated partitioned data set
172	737	08	I/O error during reading of volume label on direct-access
173	737	18	I/O error during reading of volume label on IBM 2321 Data Cell Drive
174	337	04	No EODAD exit 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
178	737	1C	I/O error during reading of extension (format 3) DSCB
179	737	1C	I/O error during reading of extension (format 3) DSCB
180	137	1C	Invalid trailer label read
184	E37	04	No more space available on volume, all volumes used

<u>Internal Code</u>	<u>ABEND Code</u>	<u>Return Code</u>	<u>Reason</u>
185	B37	04	New volume cannot be mounted because all units are reserved
186	D37	04	Primary space exhausted, no secondary space specified
187	737	10	I/O error during writing of file mark
188	737	04	I/O error during reading or writing of JFCB or extension block
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
193	637	48	I/O error during unload during lookahead mount
194	637	10	I/O error in positioning tape volume when data set opened for RDBACK
195	137	20	Invalid header label read
196	637	4C	I/O error during positioning of tape volume following user header label processing
197	837	08	New JFCB extension block not found

Message-to-Function Cross-Reference

This cross-reference list indicates the messages issued by the Open, Close, and EOVS operations. The list includes the ABEND and return codes associated with the message if the message is issued because of an error condition, and the name of the function(s) or access method executor(s) that issue the message.

To find more information about each function, refer to the "Function-to-Module Directory," which will direct you to a description of each function. Information about the executors is in the OS logic manuals for the access methods, listed in the Preface of this manual.

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC001A M		EOV Tape Input New Volume Mounting Function
		EOV Tape Output New Volume Function
		EOV Tape Output Error Recovery And Nonstandard Label Function
		EOV Direct Access Input Mount Ahead Function
		EOV Direct Access Input FEOV Repositioning Function
		EOV Direct Access Output JFCB Update Function
		EOV Direct Access Output Volume Disposition Function
		EOV Direct Access Output Next Volume Mount Function

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC001A (Continued)	M	EOV Direct-Access IBM 2321 Next Volume Mount Function
IEC001E	M	EOV Tape Input Next Volume Mounting Function
IEC002E	K	EOV Tape Input Volume Positioning Function
		EOV Tape Input New Volume Mounting Function
		EOV Tape Input Next Volume Mounting Function
		EOV Tape Output Volume Disposition Function
		EOV Tape Output Label Rewrite Function (VOL 1)
		EOV Tape Output Exit Function
		EOV Direct-Access Input Initial Function
		EOV Direct-Access Input FEOV Repositioning Function
		EOV Direct-Access Output Volume Disposition Function
IEC0003E	R	EOV Tape Input New Volume Mounting Function
		EOV Tape Input Next Volume Mounting Function
		EOV Tape Output Volume Disposition Function
		EOV Tape Output New Volume Function
		EOV Tape Output Exit Function
		EOV Direct-Access Input Initial Function
		EOV Direct-Access Input IBM 2321 Next Volume Mount Function
		EOV Direct-Access Input FEOV Repositioning Function
		EOV Direct-Access Output Get Space Current Volume Function
		EOV Direct-Access Output IBM 2321 Get Space Current Volume Function
		EOV Direct-Access Output Volume Disposition Function
IEC004E	D	EOV Tape Input New Volume Mounting Function
		EOV Tape Input Next Volume Mounting Function
		EOV Tape Output Volume Disposition Function
		EOV Tape Output Exit Function
IEC007D	E	EOV Tape Output Label Rewrite Function (VOL 1)
IEC009A	F	EOV Tape Output Label Verification Function
IEC010D	F	EOV Tape Input New Volume Mounting Function
IEC012I		EOV Tape Output Label Rewrite Function (VOL 1)
		EOV Tape Output Label Destroy (EMODVOL1) Function
		EOV Tape Output Error Recovery And Nonstandard Label Function

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC014E	D	EOV Tape Output Error Recovery And Nonstandard Label Function
IEC017I	A-1	EOV Tape Output Label Verification Function
	A-2	EOV Tape Input Standard Header Label Function
		EOV Tape Output Label Rewrite Function (VOL 1)
		EOV Tape Output Label Destroy (EMODVOL1) Function
IEC022I	137 - 04	EOV Tape Output Trailer Label Function (EOV1)
	137 - 08	EOV Tape Input Standard Trailer Label Function
	137 - 0C	EOV Tape Input Standard Trailer Label Function
	137 - 10	EOV Tape Input Standard Trailer Label Function
	137 - 14	EOV Tape Input Standard Header Label Function
	137 - 18	EOV Tape Input Standard Header Label Function
	137 - 1C	EOV Tape Input Standard Trailer Label Function
	137 - 20	EOV Tape Input Standard Header Label Function
IEC023I	237 - 04	EOV Tape Input Standard Trailer Label Function
	237 - 08	EOV Tape Input Standard Header Label Function
IEC024I	337 - 04	EOV End-of-Data Function
		EOV Executor Function
IEC025I	437 - 04	EOV Initial Read JFCB Function
IEC026I	637 - 04	EOV Tape Output Volume Disposition Function
		EOV Tape Output Label Verification Function
	637 - 08	EOV Tape Input Standard Trailer Label Function
	637 - 0C	EOV End-of-Data Function
	637 - 10	EOV Tape Input Data Set Positioning Function
	637 - 14	EOV Tape Input Volume Positioning Function
	637 - 18	EOV Tape Input Volume Positioning Function
	637 - 1C	EOV Tape Input Volume Positioning Function
	637 - 20	EOV Tape Input Volume Positioning Function
	637 - 24	EOV Tape Input Volume Positioning Function
	637 - 28	EOV Tape Input New Volume Mounting Function
	637 - 2C	EOV Tape Input New Volume Mounting Function
	637 - 30	EOV Tape Input New Volume Mounting Function
	637 - 34	EOV Tape Input New Volume Mounting Function
	637 - 38	EOV Tape Input New Volume Mounting Function
	637 - 3C	EOV Tape Input Data Set Positioning Function
	637 - 40	EOV Tape Input Data Set Positioning Function
	637 - 44	EOV Tape Input New Volume Mounting Function

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>	
IEC026I (Continued)	637 - 48	EOV Tape Input Next Volume Mounting Function	
	637 - 4C	EOV Tape Input Standard Header Label Function	
IEC027I	737 - 04	EOV Direct-Access Output FEOV Function	
		EOV Direct-Access Output B37 ABEND Function	
		EOV Direct-Access Output Next Volume Mount Function	
		EOV Direct-Access Output IBM 2321 Next Volume Mount Function	
		EOV Direct-Access Output Volume Disposition Function	
		EOV Direct-Access Output JFCB Update Function	
		EOV Direct-Access Output Read DSCB Function	
		EOV Direct-Access Output Construct DEB Function	
		737 - 08	EOV Direct-Access Input Mount Function
			EOV Direct-Access Output IBM 2321 Next Volume Mount Function
			EOV Direct-Access Input FEOV Repositioning Function
		737 - 0C	EOV Direct-Access Input Exit Function
		737 - 10	EOV Direct-Access Output FEOV Function
		737 - 18	EOV Direct-Access Input Mount Function
EOV Direct-Access Input FEOV Repositioning Function			
737 - 20	EOV Direct-Access Input Mount Ahead Function		
IEC028I	837 - 04	EOV Initial Read JFCB Function	
		EOV Initial JFCB Extension Function	
		EOV Tape Output Exit Function	
	837 - 08	EOV Initial JFCB Extension Function	
IEC030I	B37 - 04	EOV Tape Output New Volume Function	
		EOV Tape Input New Volume Mounting Function	
		EOV Direct-Access Input Mount Function	
		EOV Direct-Access IBM 2321 Get Space Current Volume Function	
		EOV Direct-Access Output B37 ABEND Function	
IEC031I	D37 - 04	EOV Direct-Access Output Get Space Current Volume Function	
		EOV Direct-Access IBM 2321 Get Space Current Volume Function	
		EOV Direct-Access Output Next Volume Mount Function	
		EOV Direct-Access Output IBM 2321 Next Volume Mount Function	
IEC032I	E37 - 04	EOV Direct-Access Output Get Space Current Volume Function	

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC032I (Continued)	E37 - 04	EOV Direct-Access IBM 2321 Get Space Current Volume Function EOV Direct-Access Output Next Volume Mount Function EOV Direct-Access Output IBM 2321 Next Volume Mount Function EOV Direct-Access Output Volume Disposition Function
IEC101A	M	Open Tape Mount Verification Function Open Tape Nonstandard Input Interface Function Open Direct-Access Volume Verification Function Open Tape Label Editor Function Open Tape Volume Mounting Function
IEC104E	D	Open Tape Mount Verification Function
IEC105I		Open EXCP Appendage Function
IEC106E	R	Open Tape Mount Verification Function Open Direct-Access Volume Verification Function
IEC107D	E	Open Tape Standard Label INPUT/MOD Header Label 1 Function Open Tape Standard Label Date Protection Function Open Tape Destroy Label Function Open Direct-Access Expiration Date Error Function
IEC108I		Open Direct-Access Volume Verification Function
IEC109A	F	Open Tape Volume Verification Function
IEC110D	F	Open Tape Volume Verification Function
IEC111E	D	Open Direct-Access Volume Verification Function
IEC112I		Open Tape Standard Label Header Label Writing Function Open Tape Standard Label Rewrite Volume Label Function Open Tape Create Label Function Open Tape Destroy Label Function
IEC113A		Open Security Read Password (READPSWD) Function
IEC114E	D	Open Tape Error Demount Function Open Tape Label Editor Function Open Tape Volume Mounting Function
	D-1	Open Tape Volume Verification Function
	D-2	Open Tape Standard Label INPUT/MOD Header Label 1 Function Open Tape Standard Label Output Security Function

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC114E	D-3	Open Tape Initial Common Function
	D-4	Open Tape Initial Common Function
		Open Tape Error Demount Function
IEC115I		Open Security Read Password (READPSWD) Function
IEC116A		Open Security Read Password (READPSWD) Function
IEC117I		Open Security Search Function
		Open Security Scratch/Rename Function
IEC118I		Open Security Search Function
IEC130I		Open Initial Function
IEC134D	A	Open Tape Destroy Label Function
IEC141I	013 - 04	Open Tape Initial Common Function
	013 - 08	Open Tape Initial Common Function
	013 - 0C	IGG0193E (Basic Direct Access Method Executor)
	013 - 10	IGG0191C (Sequential Access Method Executor)
	013 - 14	IGG0191B (Sequential Access Method Executor)
	013 - 18	IGG0191B (Sequential Access Method Executor)
	013 - 1C	IGG0191B (Sequential Access Method Executor)
	013 - 20	IGG0191A (Sequential Access Method Executor)
	013 - 24	IGG0191A (Sequential Access Method Executor)
	013 - 28	IGG0191A (Sequential Access Method Executor)
	013 - 2C	IGG0191E (Sequential Access Method Executor)
	013 - 30	IGG0191F (Sequential Access Method Executor)
	013 - 34	IGG0191I (Sequential Access Method Executor)
	013 - 38	IGG0191H (Sequential Access Method Executor)
	013 - 3C	IGG0191D (Sequential Access Method Executor)
	013 - 40	IGG01910 (Sequential Access Method Executor)
	013 - 44	IGG0191K (Sequential Access Method Executor)
	013 - 48	IGG01911 (Sequential Access Method Executor)
	013 - 4C	IGG0196B (Sequential Access Method Executor)
	013 - 50	IGG0196B (Sequential Access Method Executor)
	013 - 54	IGG0196A (Sequential Access Method Executor)
	013 - 58	IGG01912 (Sequential Access Method Executor)
	013 - 5C	IGG01915 (Sequential Access Method Executor)
		IGG01916 (Sequential Access Method Executor)
	013 - 60	IGG0191A (Sequential Access Method Executor)

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>	
IEC141I (Continued)	013 - 64	Open Merge JFCB To DCB Function Open BPAM Concatenation Function	
	013 - 68	Open DCB Exit Function	
	013 - 6C	Open DCB Exit Function	
	013 - 70	Open Merge DCB To JFCB Function	
	013 - 74	IGG0197A (Sequential Access Method Executor)	
	013 - 78	IGG0197A (Sequential Access Method Executor)	
	013 - 7C	IGG0197A (Sequential Access Method Executor)	
	013 - 80	IGG0197A (Sequential Access Method Executor)	
	013 - 84	IGG0197B (Sequential Access Method Executor)	
	013 - 88	IGG0197B (Sequential Access Method Executor)	
	IEC142I	113 - 04	Open Initial Volume Serial Function Open Tape Volume Reference Function
		113 - 08	Open Initial Volume Serial Function Open Direct-Access DISP=MOD Error Recovery Function
		113 - 0C	Open Initial Volume Serial Function
		113 - 10	Open Final Write JFCB Function
113 - 14		Open Direct-Access Parallel Mounting Function	
113 - 18		Open Direct-Access Parallel Mounting Function	
IEC143I	213 - 04	Open Direct-Access Missing DSCB 213 ABEND Exit Function	
	213 - 08	Open Security Initialization Function	
	213 - 0C	Open Direct-Access Parallel Mounting Function	
	213 - 10	Open Direct-Access Parallel Mounting Function	
	213 - 18	Open Final Write JFCB Function	
IEC144I	313 - 04	Open Direct-Access Read DSCB Function	
IEC145I	413 - 04	Open Direct-Access Volume Verification Function Open Tape Mount Verification Function	
	413 - 08	Open Tape Initial Common Function Open Tape Volume Mounting Function Open Tape Volume Verification Function Open Tape Final Common Function	
	413 - 0C	Open Tape Nonstandard Input Interface Function Open Tape Volume Verification Function Open Tape Final Common Function Open Tape Nonstandard Input Interface Function	

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>	
IEC145I (Continued)	413 - 10		
	413 - 14	Open Tape Final Common Function	
	413 - 18	Open Tape Volume Reference Function Open Direct-Access Unit Selection Function	
	413 - 1C	Open Initial Volume Serial Function	
	413 - 20	Open Direct-Access Volume Verification Function	
	IEC146I	513 - 04	Open Tape Final Common Function
IEC147I	613 - 04	Open Tape Standard Label Positioning Function Open Tape No Label Positioning Function Open Tape INPUT/MOD Header Label 1 Function Open Tape INPUT/MOD Header Label 2 Function	
		Open Tape Standard Label Input User Label Function Open Tape Standard Label Output Security Function	
		613 - 08	Open Tape INPUT/MOD Header Label 1 Function Open Tape INPUT/MOD Header Label 2 Function Open Tape Standard Label Output Security Function
	613 - 0C	Open Tape INPUT/MOD Header Label 1 Function Open Tape INPUT/MOD Header Label 2 Function Open Tape Standard Label Output Security Function	
		613 - 10	Open Tape Standard Label Rewrite Volume Label Function Open Tape Standard Label Header Label Writing Function
		613 - 14	Open Tape Standard Label Header Label Writing Function Open Tape Standard Label Output User Label Function
	IEC148I	713 - 04	Open Tape Standard Label INPUT/MOD Header Label 1 Function
		713 - 08	Open Direct-Access Expiration Date Error Function
	IEC149I	813 - 04	Open Tape Standard Label INPUT/MOD Header Label 1 Function Open Tape Standard Label INPUT/MOD Header Label 2 Function
	IEC150I	913 - 04	Open Tape Volume Verification Function
		913 - 08	Open Tape Standard Label INPUT/MOD Header Label 1 Function Open Tape Standard Label Output Security Function
913 - 0C			Open Security Search Function
IEC151I	A13 - 04	Open Tape Standard Label Positioning Function Open Tape No Label Positioning Function	

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC152I	B13 - 04	IGG0191U (Sequential Access Method Executor)
	B13 - 08	IGG0191U (Sequential Access Method Executor)
	B13 - 0C	IGG0191V (Sequential Access Method Executor)
	B13 - 10	IGG0197U (Sequential Access Method Executor)
	B13 - 14	IGG0197U (Sequential Access Method Executor)
	B13 - 18	IGG0197F (Sequential Access Method Executor)
	B13 - 1C	IGG0197F (Sequential Access Method Executor)
	B13 - 20	IGG0191T (Sequential Access Method Executor)
	B13 - 24	IGG0191T (Sequential Access Method Executor)
	B13 - 28	IGG0191T (Sequential Access Method Executor)
IEC153I	C13 - 04	Open Direct-Access BPAM Concatenation Function
	C13 - 08	Open Direct-Access BPAM Concatenation Function
	C13 - 0C	Open Direct-Access BPAM Concatenation Function
	C13 - 10	Open Direct-Access BPAM Concatenation Function
IEC154I	140 - 04	IGC0006D (RDJFCB)
IEC155I	240 - 04	IGC0006D (RDJFCB)
	240 - 08	IGC0006D (RDJFCB)
	240 - 0C	IGC0006D (RDJFCB)
	240 - 10	IGC0006D (RDJFCB)
IEC156I	03D - 04	Open Direct-Access Parallel Mounting Function
	03D - 08	Open Direct-Access Parallel Mounting Function
IEC157I	C13 - 14	IGG01934 (Graphic Access Method Executor)
IEC158I	D13 - 04	IGG01934 (Graphic Access Method Executor)
IEC159I	E13 - 04	IGG01934 (Graphic Access Method Executor)
IEC202E	K	Close Tape Volume Disposition Function
IEC210I	214 - 04	Close Tape Standard User Label Function
	214 - 08	Close Tape Volume Disposition Function
IEC211I	314 - 04	Close Direct-Access Read DSCB Function
	314 - 08	Close Direct-Access Output User Label Function
IEC212I	414 - 04	Close Direct-Access Write DSCB Function
IEC213I		Close Initialization Function
IEC214I	614 - 04	Close Direct-Access Write File Mark Function
IEC215I	714 - 04	Close Tape Standard Trailer Label Function
	714 - 08	Close Tape Standard Trailer Label Function
	714 - 0C	Close Tape Standard Trailer Label Function Close Tape Volume Disposition Function

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>	
IEC216I	A14 - 04	Close Direct-Access Partial Release Interface Function	
IEC217I	B14 - 04	IGG0201B (Sequential Access Method Executor)	
		IGG0201Z (Sequential Access Method Executor)	
	B14 - 08	IGG0201B (Sequential Access Method Executor)	
		IGG0201Z (Sequential Access Method Executor)	
	B14 - 0C	IGG0201B (Sequential Access Method Executor)	
		IGG0201Z (Sequential Access Method Executor)	
	B14 - 10	IGG0201B (Sequential Access Method Executor)	
		IGG0201Z (Sequential Access Method Executor)	
IEC218I	117 - 04	TCLOSE Final Function	
	117 - 08	TCLOSE Tape Positioning Function	
	117 - 10	TCLOSE Tape Positioning Function	
	117 - 14	TCLOSE Tape Positioning Function	
	117 - 18	TCLOSE Tape Positioning Function	
	117 - 1C	TCLOSE Tape Positioning Function	
	117 - 20	TCLOSE Tape Positioning Function	
	117 - 24	TCLOSE Tape Positioning Function	
	117 - 28	TCLOSE Tape Positioning Function	
	117 - 2C	TCLOSE Tape Standard Trailer Label Function	
	117 - 30	TCLOSE Tape Standard Trailer Label Function	
	IEC219I	217 - 04	TCLOSE Initial Function
	IEC220I	317 - 04	TCLOSE Direct-Access Input Function
TCLOSE Direct-Access Output Function			
IEC221I	417 - 04	TCLOSE Direct-Access Input Function	
		TCLOSE Direct-Access Output Function	
IEC222I	717 - 04	TCLOSE Tape Standard Trailer Label Function	
	717 - 08	TCLOSE Tape Standard Trailer Label Function	
	717 - 0C	TCLOSE Tape Positioning Function	
	717 - 10	TCLOSE Tape Positioning Function	
IEC223I		CLOSE Initialization Function	
IEC224I	D14 - 04	IGG02034 (Graphics Access Method Executor)	
IEC301A	S	Open Security Read Password (READPSWD) Function	
IEC704A	L	Open Tape Create Label Function	
		EOV Tape Output Label Create Function	
IEC705I		Open Tape Standard Label Rewrite Volume Label Function	
		Open Tape Destroy Label Function	

<u>Message ID</u>	<u>ABEND/ Return Code</u>	<u>Function</u>
IEC705I (Continued)		EOV Tape Output Label Rewrite Function
		EOV Tape Output Label Destroy (EMODVOL1) Function
IEC900I		Problem Determination Initial Function
IEC901I		Problem Determination User ABEND Function

Register Usage

The use of all registers is consistent throughout DCB 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 0-1 and 9-14 are saved and restored by the resident routine.

<u>Register</u>	<u>Name</u>	<u>Use</u>
0	R0	Work register and parameter register
1	R1	Work register and parameter register
2	RDCB	Address of current DCB
3	RBASE	Base register to be modified on entry to a module
4	RCORE	Current work area address
5	RES	Address of the resident routine
	RPAR	Address of DCB parameter list when a module controls parallel processing of DCBs
6	RWTG	Address of WTG table
	R6	Work register and parameter register
7	RPARC	Address of current entry in parameter list
	R7	Work register and parameter register
8	RWTGC	Address of current entry in WTG table
	R8	Work register and parameter register
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 the resident routine to issue a WAIT macro, offset of a branch-table entry in the next module to receive control when a module is branching to the resident routine to issue an XCTL macro, or work register
15	RF	Work register

How to Use Problem Determination for Debugging

There are two problem determination trace routines available to help in debugging. The ABEND trace routine traces control blocks associated with a specific ABEND condition. The optional work area trace routine, used if a job has failed repeatedly, traces the user's DCB, the O/C/EOV work area associated with that DCB, the beginning of the where-to-go table, and the current entry in the where-to-go table after every module of Open/Close/EOV that processes that DCB has finished.

Both traces require that the generalized trace facility (GTF) be operating in external mode while the job to be traced is running. In addition, the operator must respond "TRACE=USR" when the GTF message "SPECIFY TRACE OPTIONS" appears on the terminal. GTF performs the trace. The user must define the data set which is to be used as output from GTF. For information on how to use GTF, see the GTF chapter in *OS Service Aids*.

ABEND Trace

When an ABEND condition develops during the execution of the Open/Close/EOV operations, the ABEND trace routine of Open/Close/EOV issues a GTRACE macro instruction to call GTF and pass parameters on the data to be traced before ABEND is issued.

Optional Work Area Trace

The user requests the optional work area trace by specifying DCB=DIAGNS=TRACE on the DD statement of a data set for which the trace is desired. The trace takes place after each module of Open/Close/EOV has finished processing. Open/Close/EOV problem determination modules, initial and final modules of Open/Close/EOV, and access-method executor modules are not traced.

Because the optional work area trace increases execution time, it should probably be used for situations in which the same ABEND has occurred more than once and recovery (if permitted) has been repeatedly unsuccessful. A description of how to recover from some ABEND conditions is in *OS Data Management Services Guide*.

Formatting Traced Output

The results of either trace are dumped in hexadecimal unless formatted by the IMDPRDMP service aid. When a job that used either trace has finished processing, run the IMDPRDMP service aid against the trace data set you have defined as the output data set for GTF. Specify the following parameters on the edit statement of IMDPRDMP:

```
EDIT      DDNAME=name of trace data set, USR=DMA1
```

DMA1 is the name GTF uses to identify Open/Close/EOV data. A sample of the output produced by IMDPRDMP is in the *OS Guide to Reading System Dumps*. Complete description of the job control language is in the IMDPRDMP chapter of *OS Service Aids*.

Resident Routine Module Trace

The resident routine contains a module trace routine that, when active, lists each module to which the resident routine transfers control during execution of a processing program.

The module trace routine is located in the FE maintenance space of the resident routine. Unless you modify the code, the module trace routine is not operational. To activate it, you must 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 the resident routine. Its address is at offset X'110' in the CVT; that location is labeled CVTDMSVR.

Each time the resident routine transfers control, the module trace routine issues a write-to-operator message containing the name of the module receiving control. The 8-byte message is assigned routing codes 2 and 11, so it is sent to the console and is written on the system output device. You can change the routing by modifying the routing code, as described in *OS Supervisor Services and Macro Instructions*.

The module trace routine lists only those modules to which the resident routine transfers control. In some cases, the Open/Close/EOV modules transfer control themselves rather than going through the resident routine. In those cases, the modules to which control is transferred are not listed.

The module trace routine is not reentrant; if more than one task tries to use it at the same time, two problems may result. First, 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's message buffer, so that some messages are lost altogether.

If SMF is active, EOV may transfer control to Close modules to process SMF records. The result is that Close module names appear between EOV module names in the trace. Also, because SMF calls TCLOSE, TCLOSE module names may appear between the names of the Close modules that process SMF records.

When the module trace routine gets control, registers 2-8 are used as follows:

Register 2 contains the address of the first DCB to be processed after the transfer of control

Register 3 contains the beginning address, in the SVC transient area, of the module currently in control

Register 4 contains the address of the O/C/EOV work area of the DCB whose address is in register 2

Register 5 is the base register

Register 6 contains the address of the where-to-go table for the Open or Close operation being performed

Register 7 contains the address of the parameter list entry for the DCB whose address is in register 2

Register 8 contains the address of the where-to-go table entry for the DCB whose address is in register 2

If this routine is modified, registers 0, 1, 3, and 9-15 may be used without their contents being saved or restored.

SECTION 7: APPENDIX

Abbreviations

ABEND	abnormal end of task
ANSI	American National Standards Institute
ASCII	American National Standard Code for Information Interchange
AUL	American National Standard user label
BALR	branch-and-link register
BDAM	basic direct access method
BLP	bypass label processing
BPAM	basic partitioned access method
BPI	bits per inch
BSAM	basic sequential access method
BTAM	basic telecommunications access method
CCW	channel command word
CPU	central processing unit
CSW	channel status word
CVT	communications vector table
DASD	direct-access storage device
DCB	data control block
DD	data definition
DEB	data extent block
DSCB	data set control block
EBCDIC	extended binary-coded-decimal interchange code
ECB	event control block
EOD	end-of-data
EOF	end-of-file
EOV	end-of-volume
EOVC	end-of-volume concatenation
ERP	error recovery procedure
EXCP	execute channel program

GAM	graphics access method
GTF	generalized trace facility
HEX	hexadecimal
ID	identification
I/O	input/output
IOB	input/output block
ISAM	indexed sequential access method
JFCB	job file control block
MVT	multiprogramming with a variable number of tasks
NL	nonlabeled
NSL	nonstandard label
OCR	optical character recognition
PCI	program controlled interruption
PLM	program logic manual
PO	partitioned organization
PS	physical sequential
PSW	program status word
QSAM	queued sequential access method
QTAM	queued teleprocessing access method
RER	reduced error recovery
RPS	rotational position sensing
SL	IBM standard label
SMF	system management facility
SUL	IBM standard user label
SVC	supervisor call
TCB	task control block
TIOT	task input/output table
TSO	time sharing option
TTR	relative track address
UCB	unit control block
VTOC	volume table of contents

WTG	where-to-go
WTO	write to operator
WTOR	write to operator with reply

Glossary

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the *IBM Data Processing Glossary, GC20–1699*.

abnormal end of task (ABEND): Termination of a task before normal completion because of an error condition.

access method: A technique for moving data between main storage and input/output devices.

alternate track: For direct-access devices, a track designated to be used in place of a defective primary track.

asynchronous: Without regular time relationship; unexpected or unpredictable with respect to the execution of a program's instructions.

attribute: A characteristic; for example, attributes of a data set include record length, record format, data set name, associated device type and volume identification, use, and creation date.

auxiliary storage: Data storage other than main storage; for example, storage on magnetic-tape or direct-access devices. Synonymous with external storage, secondary storage.

basic direct access method (BDAM): An access method used to directly retrieve or update particular blocks of a data set on a direct-access device.

basic partitioned access method (BPAM): An access method that can create program libraries, in direct-access storage, for convenient storage and retrieval of programs.

basic sequential access method (BSAM): An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential-access or a direct-access device.

basic telecommunications access method (BTAM): An access method that permits read/write communications with remote devices.

buffer: An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. Synonymous with **I/O area**.

channel command word (CCW): A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs channel operations.

channel status word (CSW): A doubleword in main storage that provides information about the termination of input/output operations.

clear: To place one or more storage locations into a prescribed state, usually 0 or the space character. American National Standard Definition.

concatenated data sets: A group of logically connected data sets that are treated as one data set for the duration of a job step.

control block: A storage area used by an operating system to hold control information.

data control block (DCB): A control block used by access routines in storing and retrieving data.

data definition name (ddname): The name of a data definition (DD) statement, which corresponds to a data control block that contains the same name.

data definition (DD) statement: A job control statement that describes a data set associated with a particular job step.

data extension block (DEB): An extension of the data control block that contains information about the physical status of the data set being processed.

data set: The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set control block (DSCB): A data set label for a data set in direct-access storage.

direct access: Retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data.

direct-access storage device (DASD): A device in which the access time is effectively independent of the location of the data.

error recovery procedures (ERP): Standard procedures designed to ensure that all the routines that test particular devices provide a uniform type and quality of information.

event: An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as an input/output operation.

event control block (ECB): A control block used to represent the status of an event.

extended binary coded decimal interchange code (EBCDIC): A set of 256 characters, each represented by 8 bits.

extent: A continuous space on a direct-access storage device, occupied by or reserved for a particular data set or part of a data set.

fixed-length record: A record having the same length as all other records with which it is logically or physically associated. Contrast to **variable-length record**.

function: A unit of processing.

header label: A file or data-set label that precedes the data records on a unit of a recording medium.

job management: The functions of the job scheduler and master scheduler.

job scheduler: The part of the control program that reads and interprets job definitions, schedules the jobs for processing, initiates and terminates the processing of jobs and job steps, and records job output data.

job step task: A task that is initiated by an initiator/terminator in the job scheduler in accordance with specifications in an execute (EXEC) statement. In MVT, a job step task can initiate any number of other tasks.

main storage: All program-addressable storage from which instructions may be executed and from which data can be loaded directly into registers.

module: A unit of code.

nonresident portion (of a control program): Control-program routines that are loaded into main storage as they are needed and can be overlaid after their completion.

nonspecific volume request: In job control language (JCL), a request that allows the system to select a suitable volume.

nonstandard labels: Labels that do not conform to American National Standard or IBM System/360 standard label conventions.

nucleus: That portion of a control program that always remains in main storage.

parallel processing: Concurrent or simultaneous execution of two or more processes. Contrasts with serial processing.

partitioned data set: A data set in direct access storage that is divided into partitions, called members, each of which can contain a program or part of a program. Each partitioned data set contains a directory (or index) that the control program can use to locate a program in the library.

password: In systems with the time sharing option (TSO), a 1- to 8-character symbol that may be required at the time the user logs on the system. The password is confidential; the user identification is not. Users can also assign passwords to data sets.

problem determination: A procedure or process that a user can follow after receiving an error message to determine the cause of the error.

queued access method: Any access method that automatically synchronizes the transfer of data between the program using the access method and input/output devices, thereby eliminating delays for input/output operations.

queued sequential access method (QSAM): An extension of the basic sequential access method (BSAM), which produces a queue of input data blocks that have been processed and are awaiting transfer to auxiliary storage or to an output device.

queued telecommunications access method (QTAM): A method used to transfer data between main storage and remote terminals. Application programs use GET and PUT macro instructions to request the transfer of data, which is performed by a message control program. The message control program synchronizes the transfer, thus eliminating delays for input/output operations.

record length: A measure of the size of a record, usually specified in units such as words or characters. American National Standard definition.

simple buffering: A technique for controlling buffers in such a way that the buffers are assigned to a single data control block and remain so assigned until the data control block has been processed.

subpool: A set of 2K-byte storage blocks allocated for a particular task and identified by a label (called the subpool number).

subtask: A task that is initiated and terminated by a higher-order task.

supervisor call instruction (SVC): An instruction that interrupts the program being executed and passes control to the supervisor so that it can perform a specific service indicated by the instruction.

supervisor state: A state of the central processing unit, in which it can execute input/output and other privileged instructions.

SVC routine: A control-program routine that performs or begins a control program service specified by a supervisor call.

synchronous: Occurring with a regular or predictable time relationship.

SYSIN: A system input stream; also, a name used as the data definition name of a data set in the input stream.

SYSOUT: A system output stream; also, an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

SYS1.SVCLIB: The partitioned system data set that contains the nonresident SVC routines, nonresident error-handling routines, and access method routines.

system management facilities (SMF): An optional part of the control program that provides the means for gathering and recording information that can be used to evaluate system usage.

task: A unit of work for the central processing unit from the standpoint of the control program; therefore the basic multiprogramming unit under the control program.

task control block (TCB): The consolidation of control information related to a task.

telecommunications access method (TCAM): A method used to transfer data between main storage and remote or local terminals. Application programs use either GET and PUT or READ and WRITE macro instructions to request the transfer of data, which is performed by a message control program. The message control program synchronizes the transfer, thus eliminating delays for terminal input/output operations.

time sharing option (TSO): An optional part of OS that provides conversational time sharing from remote terminals.

transient area: A main-storage area used for temporary storage of transient routines, such as nonresident SVC or error-handling routines.

undefined record: A record having an unspecified or unknown length.

variable-length record: A record having a length independent of the length of other records with which it is logically or physically associated. Contrast to **fixed-length record**.

volume table of contents (VTOC): A table on a direct-access volume that describes each data set on the volume.

INDEX

Indexes to Systems Reference Library publications are consolidated in *OS Master Index to Logic Manuals*, GC28–6644. For additional information about any subject listed below, refer to other publications listed for the same subject in the *Master Index*.

A

abbreviations 199–202

ABEND

- called from problem determination 88
- codes, internal 178–185
- messages, cross-referenced to function and module 169–176

ABEND interpretation and recovery

- overview of processing 87–88

access method executor, exits to

- during close 38
- during open 30

AL processing

- (see SL processing)

B

B37 ABEND processing 80

BDAM

- parallel mounting 14

BLP processing

- during close 40
- during open 22

BPAM, concatenated data set 14

C

close 35

- overview of processing 35

concatenated data set processing

- BPAM 14
- direct-access volumes 66,78
- tape volumes 66

D

DADSM routines, exits to 26,80,82

data areas

- DCBOFLGS field 166
- DEBDCBMK field 164–165
- JFCBMASK field 161–164
- O/C/EOV work area 153–161
- Transfer control (XCTL) table 166–167

data areas (continued)

- Where-to-go (WTG) table 167–168
- WTGPATH field 168

DCB ABEND exit routine 12

DCB exit list routines

- ABEND exit 88
- B37 ABEND exit 80
- OPENJ 3
- TCAM exit 12
- user label routines
 - during close 40,44
 - during EOJ 66,68,70,74,78
 - during open 14,20,24,28

DCBOFLGS field 166

DEB

- built 76
- removed from TCB DEB chain 48
- restored for FEOV 78

DEBDCBMK field 164–165

directory, function to module 143–151

- open and OPENJ 143–146

- close and TCLOSE 146–148

EOV 148–150

common functions 151

E

EMODVOL1 74,130

- (see also directory)

end of data processing

- direct-access volumes 78
- tape volumes 66

end of file processing (see end of data processing)

EOD processing (see end of data processing)

EOF (see end of data processing)

EOV 61

- overview of processing 63

error processing

DCB error handling 177–178

during EOJ

- B37 ABEND exit 80
- recovery from invalid tape 74

during Open

- 213 ABEND exit 12
- DISP=MOD error recovery 12
- expiration date error recovery 12

(see also problem determination)

EXCP

- appendage processing 32
- requests, purged 38

F

function to module cross reference
for ABEND conditions 169–176

G

generalized trace facility, exit to 88,90
glossary 203–205
GTF (see generalized trace facility)

I

internal ABEND codes 178–185

IFG0190P 141
(see also directory)
IFG0190R 140
IFG0193A 96
(see also directory)
IFG0193B 97
(see also directory)
IFG0193C 99
(see also directory)
IFG0193D 99
(see also directory)
IFG0193E 99
(see also directory)
IFG0194A 105
(see also directory)
IFG0194C 105
(see also directory)
IFG0194D 97
(see also directory)
IFG0194E 103
(see also directory)
IFG0194F 97
(see also directory)
IFG0194G 98
(see also directory)
IFG0194H 98
(see also directory)
IFG0194I 98
(see also directory)
IFG0194J 99
(see also directory)
IFG0195A 103
(see also directory)
IFG0195B 100
(see also directory)
IFG0195C 101
(see also directory)
IFG0195D 101
(see also directory)
IFG0195E 104
(see also directory)
IFG0195F 105,134
(see also directory)

IFG0195G 104
(see also directory)
IFG0195H 100
(see also directory)
IFG0195J 103
(see also directory)
IFG0195K 100
(see also directory)
IFG0195M 103
(see also directory)
IFG0195N 100
(see also directory)
IFG0195O 104
(see also directory)
IFG0195P 104
(see also directory)
IFG0195T 106
(see also directory)
IFG0195U 107
(see also directory)
IFG0195V 106
(see also directory)
IFG0196J 108
(see also directory)
IFG0196K 108
(see also directory)
IFG0196L 108
(see also directory)
IFG0196M 108
(see also directory)
IFG0196N 102
(see also directory)
IFG0196O 101
(see also directory)
IFG0196P 109,138
(see also directory)
IFG0196Q 102
(see also directory)
IFG0196T 102
(see also directory)
IFG0196U 102
(see also directory)
IFG0196V 110
(see also directory)
IFG0196W 110,111
(see also directory)
IFG0196X 111
(see also directory)
IFG0198N 111
(see also directory)
IFG0199B 141
(see also directory)
IFG0199D 141
(see also directory)
IFG0199E 141
(see also directory)
IFG0199R 142
(see also directory)
IFG019RA 139
IFG0200P 141
(see also directory)
IFG0200R 140

IFG0200V 112,115
 (see also directory)
 IFG0200W 112,115
 (see also directory)
 IFG0200Y 112
 (see also directory)
 IFG0201R 115
 (see also directory)
 IFG0202B 114
 (see also directory)
 IFG0202C 116
 (see also directory)
 IFG0202D 116
 (see also directory)
 IFG0202E 116
 (see also directory)
 IFG0202F 114
 (see also directory)
 IFG0202G 114
 (see also directory)
 IFG0202H 118
 (see also directory)
 IFG0202I 118
 (see also directory)
 IFG0202J 117
 (see also directory)
 IFG0202K 117
 (see also directory)
 IFG0202L 117
 (see also directory)
 IFG0202U 121
 (see also directory)
 IFG0209B 141
 (see also directory)
 IFG0209D 141
 (see also directory)
 IFG0209E 141
 (see also directory)
 IFG0209R 142
 (see also directory)
 IFG0230P 141
 (see also directory)
 IFG0232D 120
 (see also directory)
 IFG0232G 119
 (see also directory)
 IFG0232J 120
 (see also directory)
 IFG0232M 119
 (see also directory)
 IFG0232S 119
 (see also directory)
 IFG0232Y 114
 (see also directory)
 IFG0232Z 121
 (see also directory)
 IFG0239B 141
 (see also directory)
 IFG0239D 141
 (see also directory)
 IFG0239E 141
 (see also directory)
 IFG0239R 142
 (see also directory)
 IFG0550P 141
 (see also directory)
 IFG0550R 140
 IFG0551B 122
 IFG0551F 123
 (see also directory)
 IFG0551H 123
 (see also directory)
 IFG0551J 123
 (see also directory)
 IFG0551L 124
 (see also directory)
 IFG0551N 124
 (see also directory)
 IFG0551P 128
 (see also directory)
 IFG0551R 128
 (see also directory)
 IFG0551T 128
 (see also directory)
 IFG0551V 128
 (see also directory)
 IFG0551X 129
 (see also directory)
 IFG0551Z 129
 (see also directory)
 IFG0552B 129
 (see also directory)
 IFG0552D 129
 (see also directory)
 IFG0552F 130
 (see also directory)
 IFG0552H 130
 (see also directory)
 IFG0552J 130
 (see also directory)
 IFG0552L 130
 (see also directory)
 IFG0552N 131
 (see also directory)
 IFG0552P 131
 (see also directory)
 IFG0552R 125
 (see also directory)
 IFG0552T 125
 (see also directory)
 IFG0552V 125
 (see also directory)
 IFG0552X 126
 (see also directory)
 IFG0552Z 126
 (see also directory)
 IFG0553B 132
 (see also directory)
 IFG0553D 127
 (see also directory)
 IFG0553F 127
 (see also directory)
 IFG0553H 118,127
 (see also directory)

IFG0553P 132
 (see also directory)
 IFG0553R 132
 (see also directory)
 IFG0553T 132
 (see also directory)
 IFG0553X 133
 (see also directory)
 IFG0553Z 133
 (see also directory)
 IFG0554B 133
 (see also directory)
 IFG0554D 133
 (see also directory)
 IFG0554J 134
 (see also directory)
 IFG0554L 105,134
 (see also directory)
 IFG0554N 135
 (see also directory)
 IFG0554P 135
 (see also directory)
 IFG0554R 136
 (see also directory)
 IFG0554T 136
 (see also directory)
 IFG0554V 136
 (see also directory)
 IFG0554X 137
 (see also directory)
 IFG0554Z 137
 (see also directory)
 IFG0555B 137
 (see also directory)
 IFG0555D 137
 (see also directory)
 IFG0555F 137
 (see also directory)
 IFG0555H 138
 (see also directory)
 IFG0555J 109,138
 (see also directory)
 IFG0556B 114
 (see also directory)
 IFG0559B 141
 (see also directory)
 IFG0559D 141
 (see also directory)
 IFG0559E 141
 (see also directory)
 IFG0559R 142
 (see also directory)
 IGC0001I 96
 IGC00020 112
 IGC0002B 96
 IGC0002C 119
 IGC0003A 122
 IGC0005E 122
 IGG0190A 99
 IGG0190B 101
 IGG0190R 101
 IGG0196M 141

IGG0200B 114
 IGG0200F 115
 IGG0200G 115
 IGG0550B 125
 IGG0550D 127
 IGG0550F 128
 IGG0550K 135
 IGG0550P 129
 IGG0550R 131
 IGG0550S 136
 IGG0552K 136
 introductory information
 close contents 35
 close—overview of processing 37
 closing a data set 2
 data areas 1
 EOVS contents 61
 EOVS overview of processing 63
 EOVS processing 2–3
 introduction to Open/Close/EOVS processing 1–3
 open contents 8
 open overview of processing 9
 opening a data set 1–2
 problem determination contents 85
 problem determination overview of processing 87

J

JFCBMASK field 161–164

L

label processing
 (see the following:
 AL processing
 BLP processing
 NL processing
 NSL processing
 SL processing
 user label routines)
 logical records written to SYSOUT data sets, computation of 32

M

merge processing
 DSCB to JFCB merge 14
 tape label to JFCB merge 20
 JFCB to DCB merge 28
 DCB to JFCB merge 28
 JFCB to DSCB merge 32
 module names (see specific module name)
 module trace
 problem determination 90
 resident routine 196

N

NL processing
 during close 40
 during open 22
NSL processing
 during close 40
 during open 22

O

O/C/EOV work area 153–161
OMODVOL1 18,97,99
 (see also directory)
open 8
 overview of processing 9
OPENJ processing 10
open options 11
optional module trace 90

P

problem determination 85
 overview of processing 87
problem determination, how-to-use information
 ABEND trace 196
 optional work area trace 196
 formatting traced output 196
 resident routine module trace 196–197
parallel mounting
 BDAM 14
 ISAM 14
partial release processing 44
password-protected data set processing 26

R

RDJFCB 10
read JFCB (SVC 64) 10
READPSWD 107
 (see also directory)
register usage 195
resident routine 5-7
 introduction 1

S

SAM executors, exit to 82
SECLOADA 107
 (see also directory)
security processing for password-protected data set 26

SL processing
 during close 40
 during EOVS 66,68
 during open 20,24
 during TCLOSE 54
SMF (see system management facility processing)
SYSOUT limit processing 32
system management facility processing 37,46

T

TCAM exit routine 12
TCLOSE processing (see close processing)
Traces
 ABEND interpretation and recovery 88
 optional module trace 90
 resident routine trace 196–197
transfer control (XCTL) table 166–167

U

user label routines
 during close 40,44
 during EOVS 66,68, 70,74,78
 during open 14,20,24,28
 during TCLOSE 54,56

W

where-to-go (WTG) table 167–168
WTGPATH field 168
WTG table (see where-to-go (WTG) table)

X

XCTL table (see transfer control (XCTL) table)

213 ABEND exit 12



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

READER'S COMMENT FORM

OS Open/Close/EOV Logic

Order Number GY28-6609-6

Your comments about this publication will help us to produce better publications for your use. If you wish to comment, please use the space provided below, giving specific page and paragraph references.

Please do not use this form to ask technical questions about the system or equipment or to make requests for copies of publications. Instead, make such inquiries or requests to your IBM representative or to the IBM Branch Office serving your locality.

Reply requested

Yes

No

Name _____

Job Title _____

Address _____

_____ Zip _____

No postage necessary if mailed in the U S A

YOUR COMMENTS, PLEASE . . .

This publication is one of a series which serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

fold

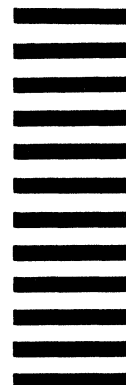
fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Monterey & Cottle Rds.
San Jose, California
95114



Attention: Programming Publications, Dept. D78

fold

fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)