

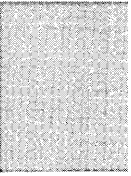
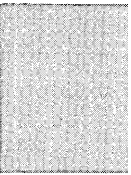
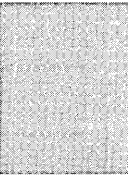
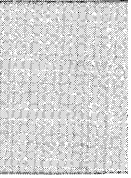
## Systems Reference Library

### OS Tape Labels

#### Release 21

This publication describes how the IBM System/360 Operating System processes magnetic tape volumes with IBM standard labels, American National Standard labels, nonstandard labels, or no labels. Included for the guidance of programmers and operations analysts are discussions of:

- Label formats and contents
- Tape volume layouts
- Label processing for input, output, multiple data set, and multiple volume conditions
- Tape characteristics
- Nonstandard label processing routines and volume label editor routines
- Component considerations
- Using tape volumes created by other systems
- External labels



### **Sixth Edition (April 1973)**

This edition replaces the previous edition (numbered GC28-6680-4) and makes that edition obsolete.

This edition applies to Release 21.7 of OS and to all subsequent releases unless otherwise indicated in new editions or technical newsletters.

Significant system changes are summarized under "Summary of Amendments for Release 21.7" and "Summary of Amendments for Release 21" following the list of figures. Each technical change is marked by a vertical line to the left of the change.

Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest *IBM System/360 and System/370 Bibliography*, GA22-6822, to learn which editions and technical newsletters are applicable and current.

Requests for copies of IBM publications should be made to the IBM branch office that serves you.

Forms for readers' comments are provided at the back of the publication. If the forms have been removed, comments may be addressed to IBM Corporation, Programming Center—Publishing, Department D58, Monterey and Cottle Roads, San Jose, California 95193. All comments and suggestions become the property of IBM.

© Copyright International Business Machines Corporation 1968, 1970, 1971, 1972, 1973

## PREFACE

This publication explains, for programmers and operations analysts, how the Operating System (OS) processes magnetic tape labels. The four main sections of the publication describe the specifications and processing of tapes with *IBM standard labels*, tapes with *American National Standard labels*, tapes with *nonstandard labels*, and *unlabeled tapes*. You need read only the section pertaining to the type of labels used by your installation. Before proceeding to a particular section, you should read the introduction to familiarize yourself with tape processing in the operating system and the terminology used in this book.

Other sections and the appendixes describe subjects of special interest, such as volume label editor routines, use of tape volumes from other systems, and external labels.

To fully understand this manual, you should be familiar with the following publications:

*OS Introduction*, GC28-6534

*OS Job Control Language Reference*, GC28-6704

*OS Data Management Services Guide*, GC26-3746

The following documents provide additional information on code and magnetic tape labeling standards approved by the American National Standards Institute (ANSI) and adopted by the Federal Government as Federal ADP Standards:

*American National Standard Code for Information Interchange*, X3.4-1968. This is a revision of the 1963 and the 1967 versions of the code, and is given the acronym ASCII.

*American National Standard Recorded Magnetic Tape for Information Interchange, 9-Track, 800 CPI, NRZI*, X3.22-1967.

*American National Standard Magnetic Tape Labels for Information Interchange*, X3.27-1969.

Copies of the American National Standards may be ordered from:

American National Standards Institute  
1430 Broadway  
New York, New York 10018

The following publications provide information about magnetic tape units:

*System/360 Component Description 2400-Series Magnetic Tape Units*, GA22-6866

*3803/3420 Magnetic Tape Subsystems Component Description*, GA32-0020

*3410/3411 Magnetic Tape Subsystems Component Summary*, GA32-0015

The following publications are referred to in the text:

*OS System Control Blocks*, GC28-6628

*OS System Generation*, GC28-6554

*OS Data Management for System Programmers*, GC28-6550

*OS Utilities*, GC28-6586

*OS MFT Guide*, GC27-6939

*OS MVT Guide*, GC28-6720



# CONTENTS

<b>Preface</b> .....	iii
<b>Figures</b> .....	ix
<b>Summary of Amendments</b> .....	xi
Summary of Amendments for Release 21.7 .....	xi
Summary of Amendments for Release 21 .....	xi
<b>Introduction to Tape Processing</b> .....	1
Describing the Labels .....	2
Describing the Data Sets .....	3
Completing the Data Control Block .....	3
Cataloged Data Sets .....	4
Generation Data Groups .....	4
Concatenated Data Sets .....	5
Passed Data Sets .....	5
Multiple Data Sets and Multiple Volumes .....	5
Processing Methods and Routines .....	6
Data Management Routines .....	6
Checkpoint/Restart .....	7
Automatic Volume Recognition .....	7
Tape Disposition .....	7
Tape Characteristics .....	8
Nine-Track Tapes .....	8
Nine-Track Dual-Density Feature .....	8
Seven-Track Tapes .....	9
Tapemarks and Reflective Strips .....	10
<b>IBM Standard Labels</b> .....	11
Label Definitions and Organization .....	11
Volume Label .....	11
Data Set Header Labels .....	12
User Header Labels .....	14
Data Set Trailer Labels .....	14
User Trailer Labels .....	14
Additional Labels .....	14
Tapemarks .....	14
IBM Standard Label Processing .....	15
Opening an Input Data Set .....	16
Volume Serial Number .....	17
Positioning the Volume to the Data Set .....	17
Data Set Name .....	18
Expiration Date .....	18
Security Protection .....	18
Block Count .....	19
Data Set Characteristics .....	19
User Header Labels .....	19
Read Backward .....	19
End of Data or End of Volume on Input .....	20
Block Count .....	20
The EOVS/EOF2 Label .....	20
User Trailer Labels .....	20
Determining Volume Switch .....	21
Checking the Next Volume .....	22

Closing an Input Data Set .....	22
Creating a Volume Label .....	23
Opening an Output Data Set .....	23
Volume Serial Number .....	24
Positioning the Volume to the Data Set .....	24
Expiration Date on Existing Label .....	25
Protection and Data Set Name on Existing Label .....	25
Writing Data Set Header Labels .....	26
Writing User Header Labels .....	26
Permanent I/O Error .....	26
End of Volume on Output .....	26
Writing Data Set Trailer Labels .....	26
Writing User Trailer Labels .....	27
Labels on New Volume .....	27
Special End-of-Volume Conditions .....	27
Closing an Output Data Set .....	27
Writing Data Set Trailer Labels .....	27
Writing User Trailer Labels .....	27
Restarting from a Checkpoint .....	28
Format of the IBM Standard Volume Label (VOL1) .....	28
Format of the IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1) .....	31
Format of the IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2) .....	36
Format of User Label (UHL1-UHL8, UTL1-UTL8) .....	40
<b>American National Standard Labels .....</b>	<b>43</b>
Label Definition and Organization .....	43
American National Standard Volume Label .....	46
Data Set Header Labels .....	46
Data Set Trailer Labels .....	46
User Header and Trailer Labels .....	46
Tapemarks .....	47
ANSI Standard Label Processing .....	47
Opening an Input Data Set .....	48
Volume Serial Number .....	49
Positioning the Volume to the Data Set .....	49
Data Set Name .....	50
Expiration Date .....	50
Accessibility .....	50
Block Count .....	51
Data Set Characteristics .....	51
User Header Labels .....	51
Read Backward .....	51
End of Data or End of Volume on Input .....	51
Block Count .....	52
The EOVS/EOF2 Label .....	52
User Trailer Labels .....	52
Determining Volume Switch .....	52
Checking the Next Volume .....	53
Closing an Input Data Set .....	54
Creating a Volume Label .....	54
Opening an Output Data Set .....	55
Volume Serial Number .....	55
Positioning the Volume to the Data Set .....	55
Expiration Date on Existing Label .....	56
Protection and Data Set Name on Existing Label .....	56
Writing Data Set Header Labels .....	57

Writing ANSI User Header Labels .....	57
Permanent I/O Error .....	57
End of Volume on Output .....	58
Writing Data Set Trailer Labels .....	58
Writing User Trailer Labels .....	58
Labels on New Volume .....	58
Special End-of-Volume Conditions .....	58
Closing an Output Data Set .....	59
Writing Data Set Trailer Labels .....	59
Writing User Trailer Labels .....	59
Restarting from a Checkpoint .....	59
Format of the ANSI Standard Volume Label (VOL1) .....	60
Format of the ANSI Standard Data Set Label 1 (HDR1/EOV1/EOF1) .....	62
Format of the ANSI Standard Data Set Label 2 (HDR2/EOV2/EOF2) .....	68
Format of ANSI User Header and Trailer Labels (UHL/UTL) .....	72
Other File Header Labels .....	73
<b>Nonstandard Labels.....</b>	<b>75</b>
Processing Tapes with Nonstandard Labels .....	76
Input Header Label Routines .....	76
Input Trailer Label Routines .....	77
Output Header Label Routines .....	78
Restart Label Processing Routine .....	78
Output Trailer Label Routines .....	79
Writing Nonstandard Label Processing Routines .....	79
Programming Conventions .....	79
Program Functions .....	80
Explanation of Logic Blocks—Figure 23, 24 and 25 .....	83
Explanation of Logic Blocks—Figure 26 .....	88
Inserting Nonstandard Label Routines Into the Control Program .....	90
Automatic Volume Recognition (AVR) Nonstandard Label Processing Routine .....	91
Entry Conditions .....	91
Conventions .....	91
Inserting AVR Nonstandard Label Routines into the Control Program .....	91
Volume Verification and Dynamic Device Reconfiguration .....	93
<b>Unlabeled Tapes.....</b>	<b>95</b>
Opening an Input Data Set .....	97
Positioning the Volume to the Data Set .....	97
Read Backward .....	98
End of Data or End of Volume on Input .....	98
Determining Volume Switch .....	98
Opening an Output Data Set .....	99
BLP (Bypass Label Processing) Option .....	99
Volume Serial Number .....	100
Positioning the Volume to the Data Set .....	100
End-of-Volume on Output .....	101
Closing an Output Data Set .....	101
Restarting from a Checkpoint .....	101
<b>Volume Label Verification and Volume Label Editor Routines.....</b>	<b>103</b>
Verification of First Record .....	103
Volume Label Editor Routines .....	106
Programming Conventions .....	107
Program Functions .....	108
Logic Block Explanations .....	112

Inserting Your Label Editor Routines Into the Control Program .....	114
<b>Using Tape Volumes Created by Other Systems:</b> .....	117
IBM System/360 Standard Labels .....	117
Nonstandard Labels .....	118
Unlabeled Tapes .....	118
<b>Appendix A: Component Considerations</b> .....	119
<b>Appendix B: External Labels</b> .....	121
Reel Label .....	121
Contents Label .....	121
<b>Appendix C: Restart Work Areas</b> .....	123
Table Entry .....	123
Work and Control Block Area .....	126
<b>Index</b> .....	129



# FIGURES

Figure 1.	Basic Tape Layouts .....	2
Figure 2.	Sources and Sequence for Completing the Data Control Block .....	4
Figure 3.	DEN Parameter Codes for Specifying Tape Density .....	8
Figure 4.	Volume Organizations with IBM Standard Labels (Single Data Set) ....	12
Figure 5.	Volume Organizations with IBM Standard Labels (Multiple Data Sets) 13	
Figure 6.	IBM Standard Label Processing by Data Management Routines .....	16
Figure 7.	Format of IBM Standard Volume Label .....	29
Figure 8.	Format of IBM Standard Data Set Label 1 .....	32
Figure 9.	Format of IBM Standard Data Set Label 2 .....	37
Figure 10.	Format of User Label .....	41
Figure 11.	Volume Organization with ANSI Standard Labels (Single Data Set) ....	44
Figure 12.	Volume Organization with ANSI Standard Labels (Multiple Data Sets) 45	
Figure 13.	ANSI Standard Label Processing by Data Management Routines .....	48
Figure 14.	Format of ANSI Standard Volume Label .....	61
Figure 15.	Format of ANSI Header 1 and Trailer 1 Labels .....	63
Figure 16.	Format of ANSI Header 2 and Trailer 2 Labels .....	69
Figure 17.	Format of ANSI User Labels .....	72
Figure 18.	Hypothetical ASCII Tape from a Non-IBM System .....	73
Figure 19.	Examples of Tape Organizations with Nonstandard Labels .....	75
Figure 20.	Status of Control Information and Pointers .....	81
Figure 21.	Format of Combined Work and Control Block Area .....	82
Figure 22.	Status of Control Information and Pointers from the Control Program's Restart Routine .....	83
Figure 23.	General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Open Routine .....	84
Figure 24.	General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Close Routine .....	85
Figure 25.	General Flow of a Nonstandard Label Processing Routine after Receiving Control from the EOVS Routine .....	86
Figure 26.	General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Restart Routine .....	89
Figure 27.	Organizations for Unlabeled Tapes .....	96
Figure 28.	Verification of First Record When Standard Labels are Specified .....	104
Figure 29.	Verification of First Record When Nonstandard Labels are Specified 105	
Figure 30.	Verification of First Record When Unlabeled Tape is Specified .....	106
Figure 31.	Editor Routine Entry Conditions from the EOVS Routine .....	109
Figure 32.	General Flow of an Editor Routine after Receiving Control from the Open Routine .....	110
Figure 33.	General Flow of an Editor Routine after Receiving Control from the EOVS Routine .....	111
Figure 34.	Component Support of Label Processing Features .....	119
Figure 35.	External Label for Reel Identification .....	121
Figure 36.	External Labels for Contents Identification .....	122
Figure 37.	Restart Table Entry .....	123
Figure 38.	Restart Work and Control Block Area .....	125



# SUMMARY OF AMENDMENTS

## Summary of Amendments for Release 21.7

### *Macro Definitions Deleted*

The macro definitions for IECDSECT, IEFUCBOB, and IEFJFCBN and the directions for adding them to the libraries have been deleted from the appendixes. Information about these macros may now be obtained in *OS Data Management For System Programmers*.

### *Other Changes*

Many minor technical and editorial changes have been made, particularly in the area of non-standard labels.

## Summary of Amendments for Release 21

### *Addition of New Tape Drives—New Programming Feature*

Character of the 3400 series and 2401 Model 8 tape drives are described in the Introduction.

### *Use of DOS Tapes on OS—New Programming Feature*

Tapes created on DOS can be processed on OS. A new LABEL subparameter (LTM) has been created for DOS unlabeled tapes. Processing differences for DOS tapes are described in "Using Tape Volumes Created By Other Systems", in the Data Set Identifier field of IBM Standard Label 1, and in the discussion about Checkpoint/Restart in the Introduction.

### *Organization of Publication Changed*

Hexadecimal offsets have been added to the restart work areas (Appendix C).

### *Miscellaneous Changes*

- Corrections have been made to the descriptions of output security, password protection, and I/O error processing for IBM Standard and American National Standard labels.
- Corrections have been made to the descriptions of the Input Header Label routine, Restart Label Processing routine, and Dynamic Device Reconfiguration (DDR) routine for nonstandard labels.
- A correction has been made to indicate that the Bypass Label Processing (BLP) option can be specified in the LABEL parameter of a DD statement as well as at system generation.
- Other technical corrections have also been made throughout this publication.



# INTRODUCTION TO TAPE PROCESSING

Labels are used to identify magnetic tape volumes and the data sets they contain. With Operating System (OS), you can process tape volumes with IBM standard labels, American National Standard labels, nonstandard labels, or no labels. However, the use of labels is recommended as a basis for efficient control of your tape volumes.

IBM standard tape labels consist of *volume labels* and groups of *data set labels*. The volume label is the first record on the tape; it identifies the volume and its owner. The data set label groups precede and follow each data set on the volume, and identify and describe the data set.

- The data set labels that precede the data set are called *header labels*.
- The data set labels that follow the data set are called *trailer labels*. They are almost identical to the header labels.
- The data set label groups can include standard *user labels* at your option.

American National Standard tape labels are defined by the American National Standards Institute (ANSI). In general, ANSI labels are similar to IBM standard labels; and unless otherwise specified, the term “standard label,” as used in this manual, refers to both IBM standard labels and ANSI standard labels. ANSI labeled tapes are coded in the American National Standard Code for Information Interchange (ASCII), whereas IBM labeled tapes are coded either in the extended binary-coded-decimal interchange code (EBCDIC) or in binary coded decimal (BCD).

Nonstandard tape labels can have any format and are processed by routines provided by your installation. Unlabeled tapes contain only data sets and tapemarks.

Figure 1 shows the basic IBM standard, ANSI standard, nonstandard, and unlabeled tape layouts for a single data set on a single volume. Detailed layouts and variations for each type are illustrated and described in the appropriate sections of this manual.

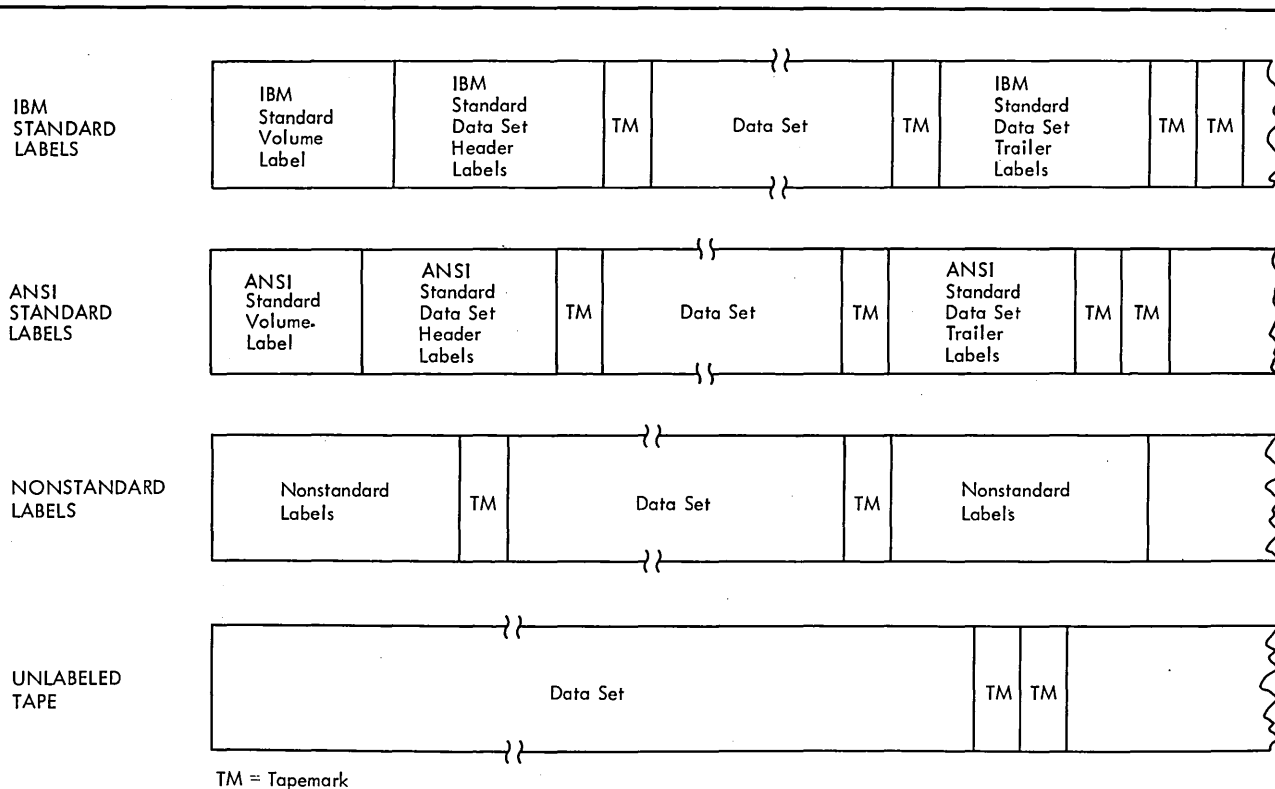


Figure 1. Basic Tape Layouts

## Describing the Labels

In the job control statements, you must provide a DD (data definition) statement for each data set to be processed. The LABEL parameter of the DD statement is used to describe the data set's labels. You specify the type of labels by coding one of the following subparameters of the LABEL parameter:

- SL IBM standard labels.
- AL American National Standard labels.
- SUL Both Standard and user labels.
- AUL Both Standard and user labels.
- NSL Nonstandard labels.
- NL No labels.
- BLP Bypass label processing. BLP may be used when a data set having no labels is to be written. BLP is treated in the same manner as NL, except that the system does not check for an existing volume label.
- LTM Bypass a leading tapemark, if encountered, on unlabeled tapes.

If you do not specify the label type, the operating system assumes that the data set has IBM standard labels.

The data set sequence subparameter of the LABEL parameter is used to specify the data set's relative position on the tape. If you do not specify the relative position, the operating system assumes that the data set is first in sequence on the reel.

When INOUT or OUTIN is specified as the processing method in the OPEN macro instruction, the LABEL parameter can be used to override this specification. If INOUT is specified and you want the data set processed for input only, code the subparameter IN in the LABEL parameter. If OUTIN is specified and you want the data set processed for output only, code the subparameter OUT in the LABEL parameter.

When new data sets are created, the LABEL parameter is used to record an expiration date and a security protection status in the label. If not otherwise specified, the expiration date is recorded as zeros (allowing the data set to be overwritten immediately), and security (password) protection is not provided.

## **Describing the Data Sets**

Other parameters of the DD statement identify the data set, give volume and unit information and volume disposition, and describe the data set's physical attributes. The information contained in the DD statement is read by the operating system and stored in a table called the job file control block (JFCB).

Each data set to be processed must also be represented by a data control block (DCB) that is created in storage by the processing program. When completed, the data control block contains full descriptive information about the data set, and is the connection between the data set, the processing program, and the operating system.

### ***Completing the Data Control Block***

Most of the information recorded in the data control block is obtained from:

- 1 — The DCB macro instruction in the processing program. The DCB macro instruction is used to construct a data control block and to provide information about the data set.
- 2 — The DD statement in the input stream (recorded in the job file control block).
- 3 — The data set label (if this is an existing data set).

The data control block is completed at execution time when it is opened. Figure 2 illustrates the sequence of filling in the data control block information. Steps 3 and 7 are bypassed if the tapes have nonstandard labels or no labels.

**Forward Merge (Steps 3 and 4):** Information from the standard data set label is merged into vacant fields of the job file control block. (Any fields that were already specified by the DD statement are not changed.) Then, in turn, information from the job file control block is merged into vacant fields of the data control block. (Any fields that were already specified by the DCB macro instruction are not changed.) When the forward merge is completed, your processing program can use a DCB exit routine to modify the data control block.

**Reverse Merge (Step 6):** After the data control block is completed, the merging process is reversed. For an input data set, information from the data control block is used to fill in any vacant fields of the job file control block. For an output data set, the data control block information overrides the job file control block information (except the data set organization field), and the updated job file control block provides the information for creating the new labels.

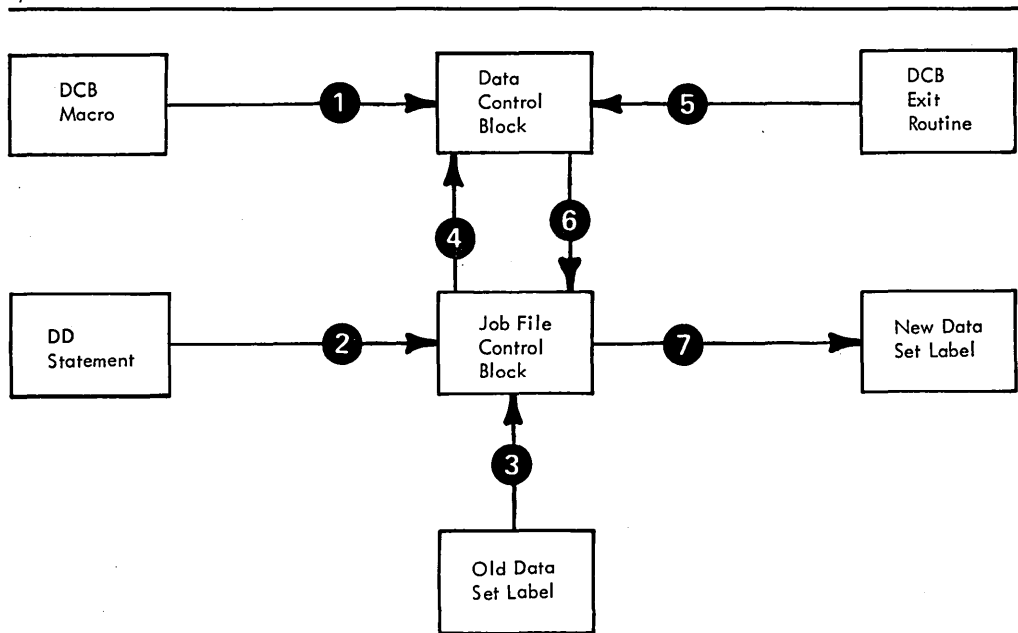


Figure 2. Sources and Sequence for Completing the Data Control Block

### Cataloged Data Sets

The operating system has facilities that can be used to automatically record the following information about each of your data sets:

- The data set name
- The serial numbers of the volume or volumes containing the data set
- The type of device on which the volumes should be mounted
- The data set's relative position on its first volume

The information is indexed by the data set name and recorded on a direct-access device in a logical structure called the catalog. You can retrieve a cataloged data set by specifying its name in the DD statement. The system finds the associated information in the catalog, and issues a mount message to the operator.

### Generation Data Groups

A cataloged data set that is frequently updated, such as a weekly payroll, can be grouped with its earlier generations to form a named generation group. A lower-level index in the catalog structure allows generation and version numbers to be included in the data set name. For example, the original generation of the data set group A.PAYROLL is named A.PAYROLL.G0000V00. The fourth update of the data set is identified as A.PAYROLL.G0004V00. The *absolute* generation and version numbers are in the form GxxxxVyy where:

xxxx

is a decimal number (0000-9999) showing the relationship to the original generation. The maximum number of generations that can be cataloged is established when the index is built for the particular generation data group.

yy

is a decimal number (00-99) identifying a version of the same generation. Only the latest version is cataloged.

You usually refer to a generation of a data set group by specifying its *relative* generation



number. For example, A.PAYROLL (0) refers to the latest cataloged generation; A.PAYROLL(-1) refers to the next-to-the-latest generation; and A.PAYROLL(+1) refers to a new generation to be added to the group.

When a generation data group index is established, a related model data set label must be built on the volume that contains the index. This model label may be used to supply uniform attributes for each generation. If you use a relative generation number to specify a new data set, attributes are taken from the model label. You can override the model label attributes with the DCB parameter of the DD statement.

Information on creating and retrieving generation data groups can be found in *OS Utilities*, and *OS Job Control Language Reference*.

### ***Concatenated Data Sets***

Several different data sets, each of which may reside on a separate volume, can be read as if they were a single data set through the technique of concatenation.

Concatenated data sets are read in the order of appearance of their DD statements in the input stream (the DD statements must follow one another and only the first DD statement is named). Each concatenated data set may be a single or multivolume data set. Concatenated data sets cannot be read backward.

Since only one data control block is associated with all the concatenated data sets, you must inform data management if the data sets have unlike characteristics (device type, block length, record format, etc.). To do this, your processing program must set a switch in the data control block, as explained in *OS Data Management Services Guide*.

### ***Passed Data Sets***

When a data set is used by two or more job steps in the same job, you can pass the data set from job step to job step. In this way, you can refer to the data set most conveniently in the DD statements for each of the later steps, which are called receiving steps. Device type, volume serial numbers, data set sequence number, and label type need not be coded in the DD statements for the receiving steps because this information is obtained from the passing step. However, the data set attributes (density, record format, etc.) are not automatically passed to the DD statements in the receiving steps. If the data set has standard labels, the receiving steps can obtain the attributes from the labels. If the data set does not have standard labels and the processing program does not define the data set attributes, then the DD statements in the receiving steps should restate the attributes.

### ***Multiple Data Sets and Multiple Volumes***

You can place a single data set on multiple volumes by coding multiple volume serial numbers in the related DD statement, or by requesting a nonspecific volume. If you request specific volumes and cataloging, all the specified volume serial numbers will be associated with the new data set in the catalog. If you use fewer volumes than you specify, you will not be able to retrieve the data set properly through use of the catalog.

You can place multiple data sets on a single volume by coding the same volume serial number on each of the related DD statements, or by using the VOLUME=REF parameter on the DD statements for the second and subsequent data sets. (VOLUME=REF=\*.ddname must not be used if the DD statement referred to requests a nonspecific volume.) You must use the LABEL parameter to specify the sequence number of each data set, both when you create it and when you retrieve it, except when retrieval is accomplished through the catalog.

You can place multiple data sets on multiple volumes by coding a set of volume serial numbers on each of the related DD statements, or you can use the VOLUME=REF

parameter. (VOLUME=REF must not be used if the data set referred to actually used fewer volumes than you specified. VOLUME=REF=\*.ddname must not be used if the DD statement referred to requests a nonspecific volume.) If you code a set of volume serial numbers for each of the data sets, the first number must be the serial number of the last volume occupied by the preceding data set.

For multiple data sets on multiple volumes, you must use the LABEL parameter to specify the sequence number of each data set, both when you create it and when you retrieve it, except when retrieval is accomplished through the catalog. The sequence number specified for each data set must indicate the relative position of the data set on the group of multiple volumes. If unlabeled data sets are retrieved in an order that differs from the order in which they were written, the specified sequence number must indicate the relative position of the data set on the first volume that occupies. Therefore, you must not use the catalog to retrieve an unlabeled data set that is out of order.

## Processing Methods and Routines

The method of processing (INPUT, OUTPUT, RDBACK, INOUT, OUTIN) is specified as an operand of the OPEN macro instruction. If you do not specify the method, INPUT is assumed.

A data set can be processed as either input or output (INPUT or OUTPUT). A data set on magnetic tape can also be read backward (RDBACK). If the basic sequential access method (BSAM) is used, a data set can also be processed as a combination of input and output (INOUT or OUTIN). For INOUT, the data set is an input data set first and then, without reopening, an output data set. For OUTIN, the data set is an output data set first and then, without reopening, an input data set.

## Data Management Routines

The input/output support routines of data management perform the label processing. These routines are Open, EOVS, and Close.

**Opening a Data Set:** The Open routine is entered when the processing program issues an OPEN macro instruction. The Open routine completes the specified data control block, and prepares and positions the data set for processing. The Open routine analyzes input header labels (or trailer labels if tape is read backward), or creates output header labels.

**End of Data Set or Volume:** The EOVS routine is entered when a tapemark is read, when the end of reel (reflective strip) is encountered, or when the processing program issues an FEOVS (force-end-of-volume) macro instruction. If you use the EXCP (execute channel program) technique, your processing program must issue an EOVS macro instruction to give control to the EOVS routine after your program recognizes a tapemark or end of reel. The EOVS routine processes trailer labels on the current volume (or header labels if the tape is read backward), and determines if additional volumes are needed to continue the data set. If another volume is needed, the EOVS routine handles the volume switching and processes the labels on the new volume. Otherwise, if the current volume is the last or only volume needed, EOVS gives control to the user's end-of-data routine that is specified in the data control block.

**Closing a Data Set:** The Close routine is entered when the processing program issues a CLOSE macro instruction. If the processing program terminates without closing the data set, the operating system calls the Close routine. The Close routine restores the fields of the data control block to the conditions that existed before the data set was opened, and logically disconnects the data set from the processing program. The Close routine creates output trailer labels, and provides for tape disposition.

## ***Checkpoint/Restart***

When a job step is restarted from a checkpoint, the Restart routine repositions tape volumes containing data sets that were open at the time the checkpoint was taken. The Restart routine also restores the applicable control blocks to the conditions that existed when the checkpoint was taken.

The Restart routine can handle tapes with IBM standard labels, ANSI standard labels, nonstandard labels, or no labels.

All DOS tapes having either a leading tapemark and/or embedded checkpoint records can be handled by Checkpoint/Restart with the exception of DOS 7-track tapes written in translate mode that contain embedded checkpoint records.

## ***Automatic Volume Recognition***

In systems with the AVR (automatic volume recognition) option, the operator can premount volumes on any unused drives. The volumes must be labeled (standard or nonstandard). The system records the volume and unit information, and assigns the drives to later job steps.

AVR checks the tape label during allocation by the scheduler, and records the volume serial number. This action merely determines which volumes are mounted on which devices—AVR does not verify or reject the volumes on the basis of their serial numbers. AVR is part of the job scheduler (as opposed to data management).

## ***Tape Disposition***

Tape disposition at end of data set or end of volume can be influenced by the DISP parameter of the DD statement. This implied disposition can be overridden by a positioning parameter of the OPEN or CLOSE macro instruction. The OPEN macro instruction controls positioning after an end-of-volume condition (multivolume data sets). The CLOSE macro instruction controls positioning at the end of the data set.

The positioning parameters of the OPEN and CLOSE macro instructions are:

- |        |   |
|--------|---|
| LEAVE  | Position the volume at the logical end of the data set just read or written. (If the data set has been read backward, the logical end is the physical beginning of the data set.)   |
| REREAD | Position the volume at the logical beginning of the data set just read or written. (When the data set exists on more volumes than there are units available, the REREAD parameter should not be used with the OPEN macro instruction—it may adversely affect the time required to mount the tapes.) |
| DISP   | Perform the disposition indicated by the TIOT. This can be REWIND or UNLOAD, depending on the volume attributes.  |

If none of the above are specified, DISP is assumed.

If necessary, the specified volume disposition can be overridden by the system. However, you need not be concerned; the system automatically requests the mounting and demounting of volumes depending on the availability of devices at a particular time.

## Tape Characteristics

The following paragraphs describe the data-recording characteristics of magnetic tape. The discussion includes density, parity, number of tracks, translation, conversion, tapemarks, etc., as related to the operating system and to IBM 2400 and 3400 Magnetic Tape Units. The error conditions that can result from conflicting tape characteristics are explained in the "Volume Label Verification and Volume Label Editor Routines" section of this publication.

The term "IBM 2400 Magnetic Tape Units" refers to the IBM 2401 Models 1-6 and Model 8, the IBM 2420 Models 5 and 7 Magnetic Tape Units, and the IBM 2415 Models 1-6 Magnetic Tape Unit, and Controls. The term "IBM 3400 Magnetic Tape Units" refers to the IBM 3420 Models 3, 5, and 7 Magnetic Tape Units and Controls, and the IBM 3410 Models 1, 2, and 3 Magnetic Tape Units and Controls. These units can write forward on a tape, and read both forward and backward.

### *Nine-Track Tapes*

Tape units without special features (except the 2401 Model 8) read and write data in 9 tracks. The operating system supports 9-track tape in densities of 800 BPI (bits per inch per track) or 1600 BPI. All 2400 Models 1-3 and 3400 drives with the dual-density feature read and write a density of 800 BPI in the NRZI (non-return-to-zero-inverted) mode. In this mode, a 1-bit is recorded on tape as a magnetic signal, and a 0-bit is recorded as the absence of a signal. All 2400 Models 4-7, 3410 Models 1-3, and 3420 Models 3, 5, and 7 read and write a density of 1600 BPI in the PE (phase encoding) mode. In this mode, a 1-bit and a 0-bit are recorded as signals of opposite polarity. Regardless of the mode, tape characters are always recorded with odd parity (a parity bit is set to 1 or 0 so that there is always an odd number of 1-bits in a 9-bit character). Tape characters are always recorded in the EBCDIC (extended binary coded decimal interchange code) or ASCII (American National Standard Code for Information Interchange) character code.

You can use the DEN parameter of the DD statement or DCB macro instruction to specify a density in bits per inch per track. Figure 3 shows the DEN parameter codes. If you do not code DEN, the system assumes the highest allowable density. If the DEN parameter specifies a density incompatible with the tape unit (for example, DEN=2 on a 1600 BPI tape unit), the job step will be abnormally terminated.

### *Nine-Track Dual-Density Feature*

The operating system supports a special feature called dual density. This feature can be installed on 2400 Models 4 and 6 and all 3400 models to allow them to read and write in either 800 BPI (NRZI mode) or 1600 BPI (PE mode). You specify the density with the DEN parameter of the DD statement or DCB macro instruction (Figure 3 shows the DEN parameter codes). If not specified, the default value is 1600 BPI.

DEN Value	7-Track	Recording Density		
		9-Track	9-Track (phase encoded)	9-Track (dual density)
0	200	—	—	—
1	556	—	—	—
2	800	800	—	800 <sup>1</sup>
3	—	—	1600	1600 <sup>2</sup>

1. Non-return-to-zero/inverted (NRZI) mode

2. Phase encoding (PE) mode

Figure 3. DEN Parameter Codes for Specifying Tape Density

For input with dual density, the density you specify is ignored. The system automatically reads the tape in its existing density.

For output with dual density, the tape is written in the density you specify or in the default density of 1600 BPI. If you request an unlabeled or a nonstandard labeled tape, you may specify either 800 or 1600 BPI. If your request is for a standard labeled tape, and the label of the mounted volume is in a density that differs from the one you specified, the system will rewrite the label to agree with your specifications, provided that you are opening the first data set on the volume. (Or your installation can provide a volume label editor routine to resolve this conflict to your own specifications.) If this is not the first output data set on a volume with a density conflict, the volume will be rewound and unloaded and a mount message will be issued for one with the correct density.

Tapes created with the dual-density feature and those created without it are interchangeable.

### ***Seven-Track Tapes***

The 7-track feature can be installed on all 2400 Models 1-3 and all 3400 models to allow them to read and write 7-track tapes instead of 9-track tapes. This special feature consists of a 7-track read/write head (replacing the 9-track head) and control unit changes, including a translator. Data can be read or written in densities of 200 BPI (2400 only), 556 BPI, or 800 BPI, with odd or even parity. The 2401 Model 8 reads and writes 7-track tapes only.

The American National Standards Institute (ANSI) does not specify support of 7-track magnetic tape for information interchange. The 7-track feature is therefore not applicable for tapes recorded in the American National Standard Code for Information Interchange (ASCII).

The translator writes 8-bit EBCDIC characters from your buffer as 6-bit BCD (binary coded decimal) tape characters, and translates the opposite way during reading. Density and parity can be set, and the translator can be turned on and off, by mode setting control commands. When the translator is off, only the six low-order bits of the characters from your buffer are written on the tape; and during reading, the two high-order bits of the characters are set to zeros.

The data conversion feature can also be installed with the 7-track feature. The data conversion feature makes it possible to write binary data on 7-track tape. It writes three main storage characters as four tape characters, and converts the opposite way during reading. Conversion is turned on and off by mode setting control commands and is mutually exclusive with translation. You must use the data conversion feature to process format-V (variable length) tape records because the length field of such records contains binary data. You cannot use the data conversion feature with the read backward (RDBACK) processing method.

The operating system supports the various densities of the 7-track feature. You specify the density with the DEN parameter of the DD statement or DCB macro instruction (Figure 3 shows the DEN parameter codes). If not specified, the default value is 800 BPI.

If the DEN parameter specifies a density incompatible with the tape unit, the job step will be abnormally terminated.

**Note:** For 7-track 3400 models, if DEN=0 is specified in the DCB macro or the DCB parameter of the DD statement, the recording density is 556 bits-per-inch. Labels and messages will, however, indicate 200 bits-per-inch.

If you use densities other than 800 BPI for 7-track system input tapes (SYSIN), system output tapes (SYSOUT), or tapes to be handled by the AVR option, you must establish the particular density for each during the system generation process.

Mode information other than density is specified with the TRTCH (tape recording technique) parameter of the DD statement or DCB macro instruction. The codes for this parameter are:

T	Odd parity with translation
C	Odd parity with conversion
E	Even parity with no translation or conversion
ET	Even parity with translation
null (parameter omitted)	Odd parity with no translation or conversion (same as 9-track)

You use the DEN and TRTCH parameters (or their default values) to specify the density and mode of the data to be read or written. If the tape contains standard labels, the DEN parameter also specifies the density of the labels. IBM recommends that all data sets on a tape containing standard labels be written in the same density. IBM standard labels on 7-track tape are always written in BCD, with the translate bit on, and even parity, regardless of the value of the TRTCH parameter.

Nonstandard labels on 7-track tape can be read or written in any code with any parity. The density of the labels need not be the same as the density of the data, but the density of associated tapemarks should be carefully planned. System recognition of tapemarks is ensured only when they are read in the density in which they were written.

### ***Tapemarks and Reflective Strips***

A data set or label group on tape is usually followed by a tapemark delimiter. A tapemark is a special character written by a control command. The tape drive recognizes a tapemark during a read operation and signals a unit exception condition. The condition is displayed by the unit exception bit in the channel status word (CSW) where it is recognized by the operating system. The tape mark is not read into main storage (unless the record read contained both the tapemark and data).

A reflective strip marks the physical end of the tape. If the reflective strip is encountered during a write operation, the hardware signals a unit exception condition in the CSW. When a tape is positioned at its load point, an indicator bit is set in either the CSW or a sense byte, and is recognized by the operating system.

## IBM STANDARD LABELS

If you specify SL in the LABEL parameter of the DD statement, or if you do not specify a label type, the data management routines of the operating system perform IBM standard label processing. If you specify SUL, data management processes both IBM standard labels and IBM standard user labels.

This section describes the organization, formats, and contents of IBM standard labels, and explains how they are processed or created.

### Label Definitions and Organization

IBM standard labels are 80-character records that are recorded in EBCDIC and odd parity on 9-track tape, or are in BCD and even parity, with the translate bit on, on 7-track tape. The first four characters are always used to identify the labels:

Label Identifier	Label Description
VOL1	Volume label
HDR1 and HDR2	Data set header labels
EOV1 and EOV2	Data set trailer labels (end-of-volume)
EOF1 and EOF2	Data set trailer labels (end-of-data-set)
UHL1 and UHL8	User header labels
UTL1 and UTL8	User trailer labels

The header and trailer labels use identical formats; therefore, there are only four different label formats. These formats are described in detail later in this section. The four types are:

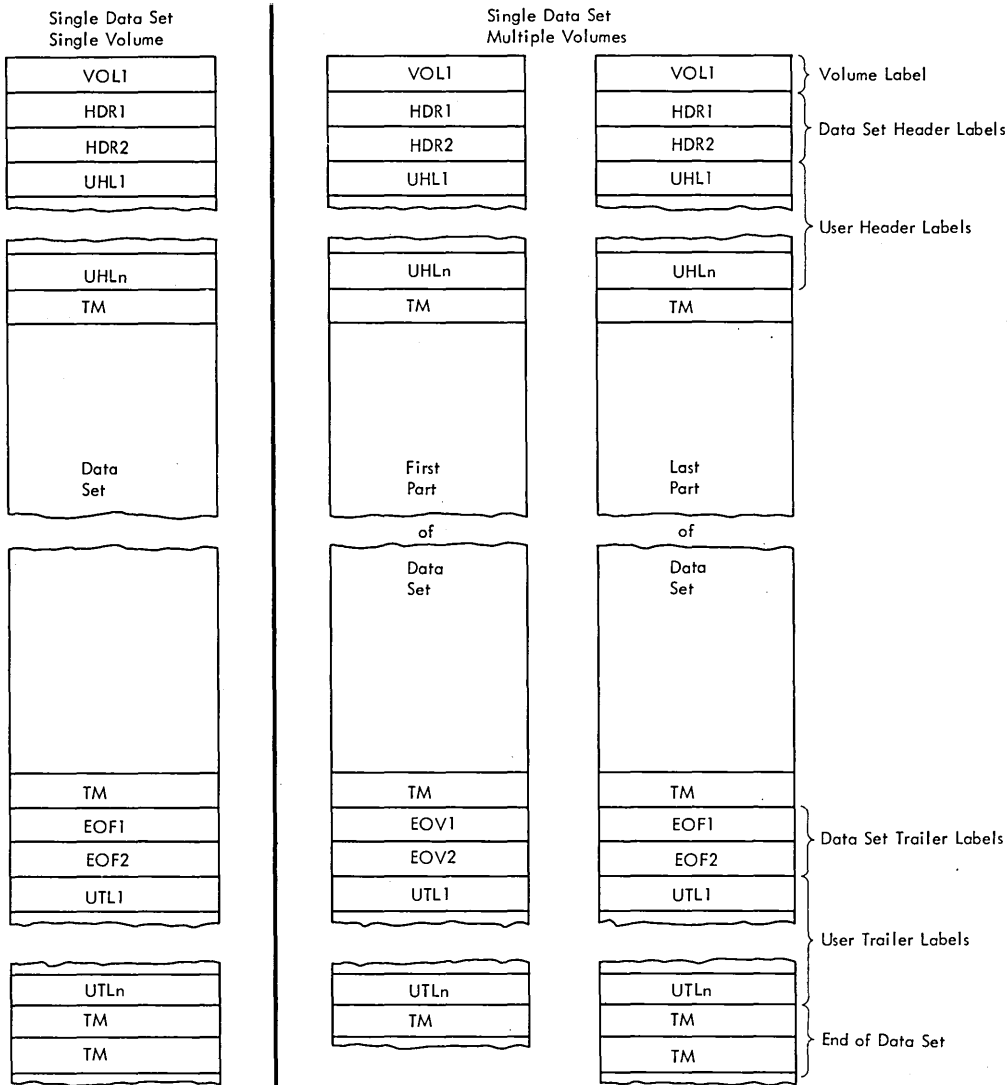
1. Standard Volume Label (identified as VOL1)
2. Standard Data Set Label 1 (identified as HDR1, EOV1, or EOF1)
3. Standard Data Set Label 2 (identified as HDR2, EOV2, or EOF2)
4. Standard User Label (identified as UHL1-UHL8 or UTL1-UTL8)

Figures 4 and 5 show the positions of the labels with various tape volume organizations. A tape with IBM standard labels must contain a volume label and data set labels. User labels are optional.

#### *Volume Label*

The IBM standard volume label (VOL1) appears at the beginning of each tape. The volume label identifies the volume and its owner and is used to verify that the correct volume is mounted.

The volume label is created by either a utility program or user's program when the tape is first received at an installation.



**Single Data Set/Single Volume:** The volume label is followed by the data set header labels and optional user header labels. The data set is preceded and followed by a tapemark. The data set trailer labels are identified as EOF and followed by optional user trailer labels. Two tapemarks follow the trailer label group to indicate that the data set is the last data set on the volume and is not continued on another volume.

**Single Data Set/Multiple Volumes:** More than one volume is needed to contain the data set. The last volume is organized the same as a single volume. On the other volumes, the data set trailer labels are identified as EOVS instead of EOF, and the trailer label group is followed by one tapemark instead of two. The data set and user labels are repeated on each volume, and there is a separate volume label for each tape.

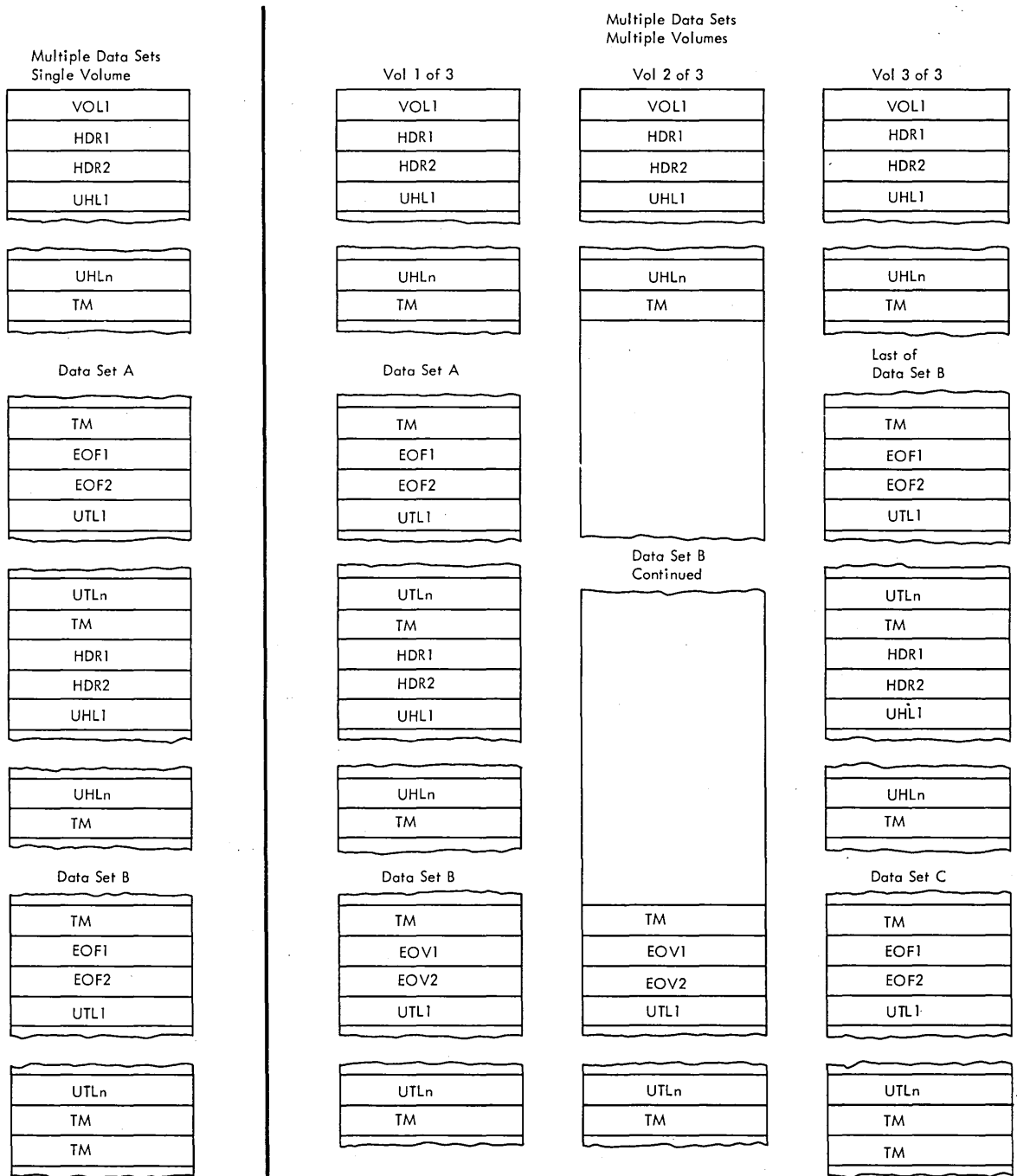
Figure 4. Volume Organizations with IBM Standard Labels (Single Data Set)

### Data Set Header Labels

The data set header label group consists of IBM standard data set label 1 (HDR1) and IBM standard data set label 2 (HDR2). The HDR1 label contains operating system and device-dependent data that relates to the data set. The HDR2 label contains additional data set characteristics. These labels are used to identify and describe the data set and to protect it from unauthorized use.

These labels are created automatically by data management each time a data set is recorded on tape.





**Multiple Data Sets/Single Volume:** The tape begins with a volume label. Each data set is preceded by a header label group and a tapemark, and is followed by a tapemark and a trailer label group. The data set trailer labels are identified as EOF. Each trailer label group is followed by a tapemark; the trailer label group for the last data set on the volume is followed by two tapemarks.

**Multiple Data Sets/Multiple Volumes:** More than one volume is needed to contain the multiple data set aggregate. The last volume is organized the same as a multiple data set/single volume layout. On the other volumes, the last data set trailer labels are identified as EOVI instead of EOF, and the last trailer label group is followed by one tapemark instead of two. There is a separate volume label for each tape.

Figure 5. Volume Organizations with IBM Standard Labels (Multiple Data Sets)

## ***User Header Labels***

Optionally, a maximum of eight user header labels (UHL1-UHL8) can appear on the tape immediately following the data set header labels. These labels contain user-specified data that can be made available to your program for processing.

If you want data management to write user header labels or to make user header labels available to your program, you must specify `SUL` on the `DD` statement and specify the address of a user header label routine in the DCB exit list. (The exit list can address several user header label routines, that is, routines that process input user header labels and create output user header labels.) The exit list (`EXLST`) is described in *OS Data Management Services Guide*.

## ***Data Set Trailer Labels***

The data set trailer label group consists of IBM standard data set label 1 (EOV1 or EOF1) and IBM standard data set label 2 (EOV2 or EOF2). These labels duplicate the IBM data set header labels so that the tape can be read backward. The trailer labels are identical to the header labels except that:

- The identifier is EOV or EOF instead of HDR.
- A block count is recorded in the first trailer label (EOV1 or EOF1) and used on input to verify that all blocks of the data set are processed. The block count field in the HDR1 label contains zeros (EBCDIC or BCD).

These labels are created automatically by data management when the data set is recorded on tape.

## ***User Trailer Labels***

Optionally, a maximum of eight user trailer labels (UTL1-UTL8) can immediately follow the data set trailer labels. These labels contain user-specified data that can be made available to your program for processing.

If you want data management to write user trailer labels or to make user trailer labels available to your program, you must specify `SUL` on the `LABEL` parameter of the `DD` statement and specify the address of a user trailer label routine in the DCB exit list. (The exit list can address several user trailer label routines, that is, routines that process input user trailer labels and create output user trailer labels.) The exit list (`EXLST`) is described in *OS Data Management Services Guide*.

## ***Additional Labels***

The operating system does not support any additional labels in the groups described above. This applies to labels identified as `VOL2-VOLn`, `HDR3-HDRn`, `UHL9-UHLn`, etc. If such labels exist on an input tape, they are bypassed. They are omitted on output tapes.

## ***Tapemarks***

Each data set and each data set label group to be processed by data management must be followed by a tapemark.

- There is no tapemark between the volume label and the first header label group on the volume.
- The tapemark that marks the end of the header label group also indicates the beginning of the data set to be processed.

- The tapemark that follows the data set also indicates the beginning of the trailer label group.
- A tapemark marks the end of the trailer label group. A second tapemark follows the trailer label group of the last data set on the volume, provided the data set does not continue on another volume.

When the operating system is used to create a data set with IBM standard labels, data management writes the necessary tapemarks.

## IBM Standard Label Processing

Label processing is handled by the I/O support routines of data management (Open, EOVS, and Close). This processing consists of three basic functions.

- Checking the labels on input tapes to ensure that the correct volume is mounted, and to identify, describe, and protect the data set being processed
- Checking the existing labels on output tapes to ensure that the correct volume is mounted, and to prevent overwriting of vital data
- Creating and writing new labels on output tapes

These processing functions are summarized in Figure 6. The table shows the specific labels that are processed for each function and which routines perform the functions. The summary in Figure 6 is the general basis for the discussions of label processing that follow.

When a data set is opened for input, the volume label and the header labels are processed. For an input end-of-data condition, the trailer labels are processed, unless deferred user input trailer label processing is specified in the data control block (DCB) exit list. For an input end-of-volume condition, the trailer labels on the current volume are processed, and then the volume label and header labels on the next volume are processed. (When the FEOV macro instruction is issued for an input tape, the trailer labels on the current volume are not processed, but the volume labels and header labels on the next volume are processed.) No label processing is performed when an input data set is closed, unless deferred user input trailer label processing is specified in the data control block (DCB). If deferred user input trailer label processing was specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

When a data set is opened for output, the existing volume label and HDR1 label are checked, and a new volume label and new header labels are written. For an output end-of-volume condition (including FEOV), trailer labels are written on the current volume, the existing volume and header labels on the next volume are checked, and then a new volume label and new header labels are written on the next volume. When an output data set is closed, trailer labels are written.

Processing	Volume Label VOL1	Header Labels <sup>1</sup>			Trailer Labels <sup>1</sup>		
		HDR1	HDR2	UHL1-8	EOF1 or EOVS	EOF2 or EOVS	UTL1-8
<b>First or only volume:<sup>2</sup></b>							
Checks Labels on input tape.	Open	Open	Open	Open	EOV	bypassed	EOV
Checks existing labels on output tape before overwriting.	Open	Open	not read	not read	not read	not read	Open <sup>5</sup>
Writes new labels on output tape.	Open or user <sup>4</sup>	Open	Open	Open	Close or EOVS	Close or EOVS	Close or EOVS
<b>Second or subsequent volumes:<sup>3</sup></b>							
Checks labels on input tape.	EOV	EOV	bypassed	EOV	EOV	bypassed	EOV
Check labels on output tape before overwriting.	EOV	EOV	not read	not read	not read	not read	not read
Writes new labels on output tape.	EOV or user <sup>4</sup>	EOV	EOV	EOV	Close or EOVS	Close or EOVS	Close or EOVS

1. For read backward operations, the action on header and trailer labels is reversed.
2. Includes the first volume of concatenated data sets with unlike characteristics. (Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program. Any exception in processing makes the data sets unlike.)
3. Includes the first volume of concatenated data sets with like characteristics.
4. User can create the label with the IEHINITT utility program or a user program. Subsequently, the label may be rewritten by the Open and EOVS routines.
5. If DISP MOD is specified on the DD statement, the Open routine positions the tape at the end of the existing data set and allows an input user trailer label routine to process user trailer labels (prior to overwriting the existing labels).

Figure 6. IBM Standard Label Processing by Data Management Routines

## Opening an Input Data Set

If IBM standard labels are specified, the first record on the input tape must be an IBM standard volume label (VOL1). At the time the data set is opened, data management checks the first record on the tape to determine whether the record is 80 bytes long and contains the identifier VOL1 in the first 4 bytes. The various error conditions that can occur during verification of the first record are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

## ***Volume Serial Number***

Data management uses the VOL1 label to ensure that the correct tape is mounted. The volume serial number in the label is compared to the volume serial number that you specify. You can specify the serial number either directly in the DD statement or indirectly through the catalog facility. Serial numbers are required when the processing method is INPUT, INOUT, or RDBACK.

If the volume serial number is correct, data management resets a mount switch in the unit control block to indicate that volume mounting is verified (the switch is initially set when the mount message is issued to the operator). If the serial number is not correct, data management rejects the tape and issues another mount message.

## ***Positioning the Volume to the Data Set***

When the volume is mounted and verified, data management positions the tape in front of the header label group of the data set to be processed. Usually there is only one data set on the reel, and the header label group immediately follows the volume label.

To retrieve a data set when more than one data set is on a single reel of tape, you specify a data set sequence number in the LABEL parameter of the DD statement, unless the data set is cataloged. You need not specify a data set sequence number for a cataloged data set, because the number can be obtained from the catalog along with the volume serial number.

- The sequence number can be from 1 to 9999, with 1 representing the first data set on the volume. If you specify a sequence number higher than the number of data sets on the volume, the tape is spaced through and removed from its reel.
- If the data set is not cataloged and you do not specify a sequence number, or you specify 0, data management assumes that the data set is the first in sequence on the volume.

To position the tape, data management uses the requested data set sequence number shown in the JFCB and the data set sequence number shown in the first HDR1 label on the tape, and maintains a logical data set sequence number in the unit control block (UCB). The number in the UCB represents the current position of the tape and is maintained as follows:

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened, the Open routine sets the data set sequence number in the UCB to 1. The exceptions are:
  - If the tape is still positioned from previous processing, such as for a LEAVE request, the Open routine does not reset the number in the UCB.
  - If the data set sequence number in the JFCB and the data set sequence number in the first HDR1 label on the tape are both greater than one, the Open routine sets the data set sequence number in the UCB to the value of the number in the first HDR1 label. (The data set sequence number in the first HDR1 label may be greater than one when the volume is part of a multiple-data-set/multiple-volume aggregate.)
3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the Open routine adjusts the data set sequence number in the UCB as the tape is spaced past each data set, until the number in the UCB equals the number in the JFCB.

4. When multiple tape units are used and a volume switch causes processing to be continued on a volume on a different unit, the EOV routine copies the data set sequence number from the previous UCB to the current UCB.

Only one data set on a tape volume may be open at any given time. An attempt to begin processing of a second data set on the same volume results in abnormal termination.

When the tape is positioned to the data set header label group of the first data set or the requested data set, data management checks the label identification. If the identifier HDR1 is not found, processing is abnormally terminated.

### ***Data Set Name***

To ensure that the correct data set is being opened, data management compares the data set name shown in the HDR1 label of the requested data set to the data set name specified by the user in the DD statement. This comparison is made on only the 17 least significant (rightmost) characters of the data set name (including 8 characters for the generation and version numbers if the data set is part of a generation data group).

If the comparison shows an incorrect data set name, processing is abnormally terminated.

### ***Expiration Date***

The expiration date shown in the HDR1 label is not verified for input data sets, unless the processing method is INOUT. For INOUT, if the expiration date has not been reached, data management notifies the operator and asks him to confirm use of the tape. If confirmation is not received, processing is abnormally terminated. (If you override the INOUT specification by coding LABEL=(,,IN) on the DD statement, the expiration date is not verified.)

### ***Security Protection***

An indicator in the HDR1 label shows whether the data set is protected against unauthorized use. If security protection is specified, data management asks the operator or the TSO terminal user to key in the correct password.

The password is verified in a user-established password data set. This password data set contains the data set name, the password, and a protection mode indicator. The protection mode indicator is set to permit either read/write or read-only operations. Processing is terminated if:

- The operator or TSO terminal user, in two attempts, does not supply the correct password.
- The password record for the data set to be opened does not exist in the password data set.
- The read-only protection mode is specified for the INOUT or OUTIN processing method.

*OS Data Management for System Programmers* describes the protection feature in detail and contains the information you need to create and maintain the password data set.

## ***Block Count***

The block count shown in the HDR1 label is always 0 (EBCDIC or BCD). This 0 is recorded in the data control block (in binary) and incremented during processing for comparison to the block count shown in the trailer label (EOV1 or EOF1).

For reading backward, the block count shown in the trailer label (EOV1 or EOF1) is recorded in the data control block and decremented during processing for comparison to the 0 block count in the HDR1 label.

The block count is verified at end-of-data or end-of-volume.

## ***Data Set Characteristics***

The HDR2 label immediately follows the HDR1 label. Data management uses the HDR2 label to determine certain data set characteristics, if these characteristics are not otherwise specified by the user. The characteristics that can be obtained from the HDR2 label are:

- Record format
- Block length
- Logical record length
- Tape recording technique (7-track tape only)
- Type of control characters

The above information is obtained from the label and recorded in the job file control block (JFCB) and the data control block, provided the appropriate fields in these control blocks contain zeros. The label information cannot override any characteristics previously specified in the processing program or the DD statement. This merging process is explained and illustrated in the Introduction to this manual.)

Unless user header labels are to be processed, data management spaces forward past the tapemark immediately after processing the HDR2 label. All labels that follow the HDR2 label are bypassed and the tape is positioned at the first data set record.

## ***User Header Labels***

Up to eight user header labels (UHL1-UHL8) may follow the HDR2 label. To make the user header labels available to your program, SUL must be coded on the DD statement and the address of an input user header label routine must be specified in the DCB exit list. If you omit one of these parameters, data management forward spaces past the tapemark immediately after processing the HDR2 label.

## ***Read Backward***

For the read backward (RDBACK) processing method, data management uses the data set's trailer labels as header labels, and vice versa. Each label group is read in the normal sequence; that is, EOF1 before EOF2, etc. The data records, however, are read in reverse sequence.

Multivolume data sets can be read backward. Concatenated data sets, 7-track tape with data conversion, and format V (variable length) records cannot be read backward.

## End of Data or End of Volume on Input

Data management's EOVS routine handles both end-of-data-set and end-of-volume conditions on input. These conditions occur when:

- A tapemark is read.
- An FEOV (force-end-of-volume) macro instruction is executed by the processing program.

After encountering a tapemark, data management checks the first 4 bytes of the first trailer label for the identifier EOVS1 or EOF1. If neither identifier is found, processing is abnormally terminated. When the FEOV macro instruction is executed, the trailer labels are not checked.

### ***Block Count***

To verify that all records on the input data set on the current volume have been read, data management compares the block count shown in the first trailer label (EOVS1 or EOF1) against the block count that was accumulated in the data control block. For reading backward, data management compares the 0 block count shown in the HDR1 label against the block count in the data control block.

If the block count in the label does not equal the block count in the data control block, the EOVS routine gives control to the appropriate entry in the user's DCB exit list. This entry in the exit list is identified as 0B (hexadecimal). The EOVS routine passes the following information to the exit routine:

Register 0      The block count shown in the label  
Register 1      The address of the data control block

After your exit routine analyzes the discrepancy (and possibly prints a message), your exit routine must return to the EOVS routine with one of the following return codes in register 15:

- 0 Abnormally terminate with completion code 237 (hexadecimal)
- 4 Continue processing

If you do not provide the appropriate user exit entry in the DCB exit list, a block count discrepancy will cause processing to abnormally terminate with a completion code of 237 (hexadecimal).

When the FEOV macro instruction is executed, the block count is not verified.

### ***The EOVS2/EOF2 Label***

Data management ignores the second trailer label (EOVS2 or EOF2) of an input data set, except when it is used as a header label for a read backward operation.

### ***User Trailer Labels***

If user trailer labels (UTL1-UTL8) are present on input, data management can make them available to your program. To make them available, SUL must be coded on the DD statement and the address of an input user trailer label routine must be specified in the DCB exit list.



## ***Determining Volume Switch***

For a multivolume input data set, you must specify the serial numbers of all the volumes to be processed. The serial numbers are specified either directly in the DD statement or indirectly through the catalog procedure. You specify the serial numbers in forward sequence, regardless of whether the tapes are to be read forward or backward.

- For noncataloged data sets, you specify the volume serial numbers in the **VOLUME** parameter of the DD statement. Data management processes the group of volumes in whatever order you specify and processes only the volumes you specify.
- For cataloged data sets, the group of volumes must be processed in sequential order. However, you can begin processing at any volume of the group by specifying a volume sequence number in the **VOLUME** parameter of the DD statement.

For input, the label identifier of the trailer labels determines whether data management continues processing the data set. When data management finds an EOJ label, it performs volume switching. When data management finds an EOF label, it passes control to the user's end-of-data routine, with one exception: If the DD statement specifies **OPTCD=B**, data management performs volume switching if an additional volume is available. If **OPTCD=B** and no volumes are available for switching, data management passes control to the user's end-of-data routine.

To determine whether additional volumes are required, data management maintains a volume sequence number in the data extent block (**DEB**) in storage.

- For read forward operations, the volume sequence number in the **DEB** is incremented as each volume is processed. This count is compared to the total number of volumes requested, as shown in the **JFCB**.
- For read backward operations, the volume sequence number in the **DEB** is initialized to the total number of volumes requested, as shown in the **JFCB**. The **DEB** count is decremented as each volume is processed until the count reaches 0.

If another volume is not required (end-of-data-set condition), control is given to the user's end-of-data routine that is specified in the data control block. Subsequently, the processing program or the operating system closes the data set.

- The user's end-of-data routine is not entered until the last specified volume or the last concatenated data set is processed.
- If an input data set is closed before the end of the data is reached, the user's end-of-data routine is not entered.

If another volume is required (end-of-volume condition), data management obtains the next volume serial number from the **JFCB** and performs volume switching. If the new volume is not already mounted, the EOJ routine issues a mount message to the operator.

When multiple tape units are being used, the EOJ routine also checks to see if a next-plus-one volume is specified, and if the volume just completed can be rewound and unloaded. If so, the EOJ routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid—the next-plus-one volume label is not verified at this time.

## Checking the Next Volume

When volume switching is performed for multiple volume input, the EOVS routine checks the volume and header labels on the new volume.

The VOL1 label is checked as if it were the first volume of the group; that is, the volume serial number is verified to ensure that the correct volume is mounted.

The method of locating and checking the HDR1 label varies according to the situation. The processing depends on whether the data set is a continuation of a multivolume data set or is a concatenated data set (Data sets with like characteristics can be processed using the same data control block (DCB), input/output block (IOB), and channel program.)

- *Multivolume data set:* The data set sequence number is irrelevant for the second and subsequent volumes of a multivolume data set. The EOVS routine assumes that the data set continues at the beginning of the new volume and, therefore, checks the first header label group on the tape. The HDR1 label is checked in the same manner as when the data set was opened on the first volume. If the data set name is not the same, or the password was not verified for each volume, processing is abnormally terminated.
- *Concatenated data sets:* The EOVS routine handles concatenated data sets with like characteristics. Such data sets are not necessarily the first on the volume, so the EOVS routine positions the tape according to the specified data set sequence number. This positioning is the same as for opening a data set. is checked in the same manner as when the first data set was opened, including verification of the password if protection is indicated.

The HDR2 label on the new volume is not processed. The data set characteristics that were established when the data set was opened apply to all subsequent volumes handled by the EOVS routine.

The data set's block count is not accumulated from volume to volume. It is initialized and verified separately for each volume.

## Closing an Input Data Set

The Close routine does not process trailer labels on an input data set. Usually, the trailer labels are processed by the EOVS routine before the data set is closed, unless deferred user input trailer label processing was specified in the data control block (DCB) exit list. If deferred user input trailer label processing was specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

If an input data set is closed before it reaches the end-of-data or the end-of-volume, or if the FEOVS macro instruction is executed, processing of trailer labels is omitted.

## Creating a Volume Label

The IBM standard volume label (VOL1) is usually written by a utility program when the reel of tape is first received at the installation. At that time, a permanent volume serial number is assigned to the reel, physically posted on the reel, and recorded in the VOL1 label.

You can use the IBM-supplied IEHINITT utility program to create IBM standard volume labels. IEHINITT initializes the tape by writing in the following order:

1. A volume label (VOL1) with the volume serial number and owner identification that you specify. You cannot specify any other fields of the VOL1 label.
2. A dummy header label (HDR1 followed by 76 EBCDIC zeros).
3. A tapemark.

The IEHINITT utility program can write a volume label on a labeled, unlabeled, or blank tape—it makes no checks to see what data, if any, previously exists on the tape. Detailed procedures for using the program are contained in *OS Utilities*.

Methods other than the IEHINITT utility program can be used to write volume labels. You can use a card-to-tape program, or you can replace the IBM-supplied volume label editor routine (see Volume Label Verification and Volume Label Editor Routines) with one that writes volume labels. If you use an editor routine to write the volume label, some data or a tapemark should already exist on the tape; otherwise, data management reads through the entire reel of blank tape looking for a label.

The VOL1 label is rewritten (that is, reproduced) by Open or EOVS routines if all the following conditions are met:

- OUTPUT or OUTIN is specified in the OPEN macro instruction.
- The tape is positioned to the first data set on the volume. (The tape cannot be blank.)
- The data set is not security-protected.

All VOL1 labels that can be successfully read under these conditions are rewritten.

If you request a standard labeled (SL, SUL) output volume and the tape that you are allocated is recorded in the wrong density and cannot be read, the VOL1 label is rewritten in the density that you specify. This facility allows you to make nonspecific requests (that is, you need not specify a volume serial number in your DD statement) for output tapes and allows the operator to mount any available scratch tape to answer your request.

If you make a nonspecific output volume request for a standard labeled (SL, SUL) tape and the mounted volume is an NL or NSL labeled tape, the Open or EOVS routine sends a message to the console operator requesting serial number and owner information (see Figure 7, Fields 3 and 8) and creates a volume label (VOL1).

## Opening an Output Data Set

If IBM standard labels are specified, the first existing record on the output tape must be a IBM standard volume label (VOL1). At the time the data set is opened, data management checks the first existing record on the tape to determine whether the record is 80 bytes long and contains the identifier VOL1 in the first 4 bytes. The various error conditions that can occur during verification of the first record are explained in “Unlabeled Tapes” in this publication.

## ***Volume Serial Number***

You are not required to specify volume serial numbers for output tapes. If none is specified, the mount message directs the operator to mount a scratch tape. Data management obtains the volume serial number from the VOL1 label and records it in the JFCB and the unit control block.

If you choose to specify the volume serial number, data management compares it with the volume serial number shown in the VOL1 label. If the number is correct, data management resets a mount switch in the unit control block to indicate that volume mounting is verified (the switch is initially set when the mount message is issued to the operator). If the volume serial number is incorrect, data management rejects the tape and issues another mount message. Volumes are requested for mounting in the order specified.

## ***Positioning the Volume to the Data Set***

When the volume is mounted and verified, data management positions the tape to receive the new data set. Usually the new data set will be the first and only data set on the tape, so the tape remains positioned immediately following the VOL1 label.

To create a data set that follows another data set already stored on the tape, you specify a data set sequence number in the LABEL parameter of the DD statement.

- The sequence number can be from 1 to 9999, with 1 representing the first data set on the volume. If you specify a sequence number that is two greater than the number of data sets existing on the volume, one of two things may happen: (1) the tape will be spaced through and removed from its reel or (2) the data set will be written but separated from the preceding data set by unusable (old) data.
- If you do not specify a sequence number, or specify zero, data management assumes that the data set is to be written as the first on the volume.

To position the tape, data management maintains a logical data set sequence number in the unit control block (UCB). The method of positioning is the same as that previously explained for opening an input data set.

Only one data set on a tape volume can be open at any given time. If you attempt to open another data set on the same volume, processing is abnormally terminated. This restriction includes system output (SYSOUT) tapes.

When the tape is positioned to receive the new data set, data management expects to find either an existing HDR1 label or a tapemark. If neither one is present, data management assumes that other data is recorded where the HDR1 label should be, and therefore processing is abnormally terminated. (If the last data set on a tape has EOF labels, another data set cannot be written to follow it.)

If a tapemark is found, it indicates that a HDR1 label does not exist at the position where the new data set is to be written. Data management bypasses all further label verification and accepts the tape for output. The conditions under which data management finds a tapemark instead of a HDR1 label are:

- When a tapemark immediately follows the VOL1 label. This may occur when the tape is initialized by means other than the IEHINIT utility program (IEHINIT writes a dummy HDR1 label following the VOL1 label). The tapemark is overwritten by the new HDR1 label.
- When, for multi-data-set organizations, the new data set is to be written after the last existing data set on the volume. In this case, data management encounters the second tapemark following the existing EOF trailer label group. The tapemark is overwritten by the new HDR1 label.

If data management finds an existing HDR1 label, it checks the label to determine whether the existing data set may be overlaid.

### ***Expiration Date on Existing Label***

The existing HDR1 label is inspected for the expiration date. If the expiration date has not been reached, the operator is asked to confirm use of the tape or to mount another tape.

If other data sets exist on the same volume, data management checks only the one expiration date and assumes that all following data sets expire on the same date.

### ***Protection and Data Set Name on Existing Label***

After checking the expiration date, data management inspects the security indicator in the existing HDR1 label. This indicator shows whether the existing data set is protected against unauthorized use.

If no protection is indicated, the tape is accepted for output. Data management does not request a password, and does not check the data set name.

If protection is indicated, data management compares the data set name shown in the existing HDR1 label to the name specified by the user in the DD statement. If the names are not the same, processing is abnormally terminated unless the data set is the first one on the first or only volume. In this case, even if you specify a specific volume, the operator will be requested to demount the tape and mount a new scratch tape. If a security protected data set is deleted, the data set security byte in the HDR1 label must be set to 0 before the volume can be written on again. This can be done by using either the IEHINITT utility or a user program to relabel the volume.

If the data set name is correct, data management requests the operator or TSO terminal user to key in the required password. The password is verified in a user-established password data set. This password data set contains the data set name, the password, and a protection mode indicator. The protection mode indicator is set to permit either read/write or read-only operations. The read/write mode is necessary for output data sets. Processing is terminated if:

- The operator or TSO terminal user, in two attempts, does not supply the correct password.
- The password record for the data set to be opened does not exist in the password data set.
- The read-only protection mode is specified.

*OS Data Management for System Programmers* describes the protection feature in detail, and contains the information you need to create and maintain the password data set.

**Note:** Verification of existing labels is considered complete after checking the HDR1 label. Any labels, data, data sets, or tapemarks following the HDR1 label are irrelevant and may be overlaid by the new output.

## ***Writing Data Set Header Labels***

When the tape is accepted by data management for output, data management creates the header labels (HDR1 and HDR2) for the new data set. These labels are created from information in the updated JFCB and other system control blocks.

The source of information for each field of each label is explained in the description of label formats. The process of updating the JFCB is explained in the introduction to this manual.

If no user header labels are to be written, data management writes a tapemark after the HDR2 label. The tape is then ready to receive the new data set.

## ***Writing User Header Labels***

When SUL is coded on the DD statement and the address of an output user header label routine is specified in the DCB exit list, data management can write as many as eight user header labels (UHL1-UHL8).

## ***Permanent I/O Error***

If a permanent I/O error occurs during label processing, and the data set is the first one on the first or only volume, the operator will be requested to demount the tape and mount a scratch tape, even if you request a specific volume. If the data set is not the first one on the volume or this is not the first volume of a multivolume data set, the job will be abnormally terminated.

## **End of Volume on Output**

Data management's EOVS routine automatically switches volumes when an end-of-volume condition occurs, that is, when a reflective strip is encountered or when a FEOV macro instruction is executed. This volume switching includes:

1. Writing trailer labels on the current volume
2. Checking existing labels on the new volume
3. Writing header labels on the new volume

When multiple tape units are being used, the EOVS routine also checks to see if a next-plus-one volume is needed, and if the volume just written can be rewound and unloaded. If so, the EOVS routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid—the next-plus-one volume label is not verified at this time.

## ***Writing Data Set Trailer Labels***

Trailer labels are always written at an end-of-volume condition on output tapes. These labels are identified as EOVS1 and EOVS2 (as opposed to EOF for end of data). These labels are created in the same manner and with the same content as the data set header labels, except for the label identifiers and the block count.

At end of volume, one tapemark is written following the data set trailer labels (as opposed to two tapemarks for end of data). If user trailer labels are to be written, the tapemark follows the user labels.

### ***Writing User Trailer Labels***

When SUL is coded on the DD statement and the address of an output user trailer label routine is specified in the DCB exit list, data management can write as many as eight user trailer labels (UTL1-UTL8).

### ***Labels on New Volume***

The EOVR routine handles label processing on the new volume (checking existing labels and writing new labels). The processing is the same as the Open routine's handling of the first volume.

When creating a multivolume data set, the data set sequence number is irrelevant for the second and subsequent volumes. The EOVR routine assumes that the data set continues at the beginning of the new volume.

### ***Special End-of-Volume Conditions***

When a reflective strip causes end-of-volume condition during the writing of data, the EOVR routine writes the trailer labels as described above. If the reflective strip is encountered while writing the trailer labels, the EOVR or the Close routine continues to write the trailer labels. In both cases, the data set can be read or overwritten normally even though it crosses the reflective strip.

If you add another data set to a tape (multi-data-set organization) on which the last existing trailer label group crossed the reflective strip, or on which the new header label group crosses the reflective strip, data management:

1. Writes the new header label group
2. Allows the user to write one record
3. Writes the new trailer label group
4. Performs volume switching

### ***Closing an Output Data Set***

The Close routine handles end-of-data-set processing on output tapes. When a write operation is the last operation that occurs before closing a data set (for OUTPUT, OUTIN, or INOUT) or when no output is written before closing (for OUTPUT or OUTIN), the Close routine creates data set trailer labels.

### ***Writing Data Set Trailer Labels***

The Close routine writes the data set trailer labels with the identifiers EOF1 and EOF2. These labels are created in the same manner and with the same content as the data set header labels, except for the label identifiers and the block count.

The Close routine writes two tapemarks following the trailer labels. If user labels are to be written, the tapemarks follow the user trailer labels. If another data set is added to the tape (multi-data-set organization), its HDR1 label overlays the second tapemark.

### ***Writing User Trailer Labels***

When SUL is coded on the DD statement and the address of an output user trailer label routine is specified in the DCB exit list, the Close routine can write as many as eight user trailer labels (UTL1-UTL8).

## Restarting from a Checkpoint

When a job step is restarted from a checkpoint, the Restart routine repositions tape volumes containing data sets that were open when the checkpoint was taken.

Specifically, the Restart routine:

1. Restores applicable control blocks to the conditions that existed when the checkpoint was taken.
2. Ensures that the first existing record on the tape is a standard volume label (VOL1), and verifies the volume serial number shown in the label.
3. Uses the data set sequence number shown in the JFCB to position the tape at the interrecord gap preceding the first record of the required data set. The method of positioning is the same as previously explained for opening an input data set. The data set labels are not reprocessed.
4. Uses the block count shown in the DCB to reposition the tape at the proper record within the data set. This positioning is always performed in a forward direction. If the block count is 0 or a negative number, the tape remains positioned at the interrecord gap preceding the first record.

If a SYSOUT data set was open when the checkpoint was taken, the data set written into during restart differs from the data set used originally. The system writes job separators at the beginning of the SYSOUT data set used during restart.

## Format of the IBM Standard Volume Label (VOL1)

The IBM standard volume label (VOL1) is 80 characters in length and is used to identify the tape volume and its owner. It is always the first record on an IBM standard labeled tape. It is recorded in EBCDIC on 9-track tape units, or in BCD on 7-track tape units.

Figure 7 shows the format of the volume label. The shaded areas represent fields that are recorded in the label, but are not used or verified during processing. The contents and processing of each field of the label are described below. The processing descriptions refer to following system control blocks:

- Job file control block (JFCB)
- Unit control block (UCB)

Location of areas within these control blocks can be found in *OS System Control Blocks*.

### 1 — Label Identifier (3 bytes)

- *Contents:* The characters VOL identify this label as a volume label.
- *Processing:* This field is read to verify that a standard labeled tape is mounted, and that this label is a volume label.

The labels are created initially by the IEHINITT utility program or a user's program.

In the following situations, Open or EOVR routines create a standard volume label based on specifications in your DD statement:

When you request a standard labeled (SL, SUL) output volume, and an NL or NSL labeled volume is mounted to satisfy your request

When you request a standard labeled (SL, SUL) output volume, and a volume that can be overwritten and that is recorded in the wrong density is mounted to satisfy your request



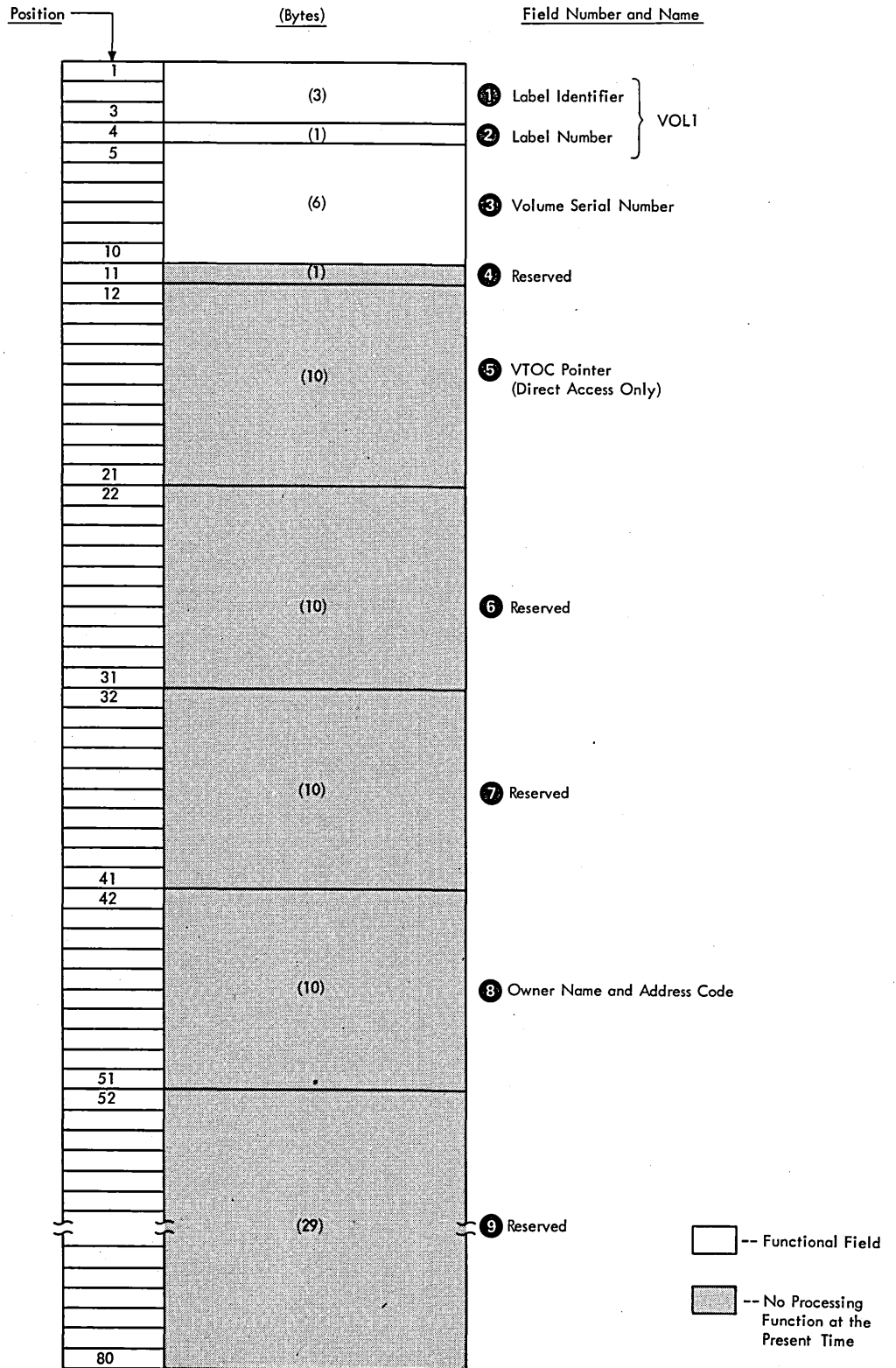


Figure 7. Format of IBM Standard Volume Label

## 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always a 1 for the IBM standard volume label.
- *Processing:* Verified in conjunction with Field 1 to identify this label as VOL1.

## 3 — Volume Serial Number (6 bytes)

- *Contents:* A unique identification code that is assigned through IEHINITT to the volume when it enters the system, or that is assigned by the operator when the Open or EOVS routines label the volume. This code may also appear on the external surface of the volume for visual identification. The code is normally numeric (000001 - 999999), but may be any six alphanumeric characters. It may be from one to six characters, but if fewer than six characters, the code must be left-justified.

If the volume serial number is assigned in the JCL statements, all national characters, the hyphen, and other special characters are accepted when enclosed by apostrophes. However, their use is not recommended because difficulties can occur in recognizing volume serial numbers when typewriter heads and print chains with nonalphanumeric characters are used.

If the volume serial number is assigned through the IEHINITT utility program, A through Z, 0 through 9, and the hyphen are the only valid characters that may be specified.

- *Processing:* The user-specified volume serial number is obtained from the JFCB and recorded in the UCB. Then the number in the UCB is compared to the number in this field of the label to ensure that the correct volume is mounted.

For scratch output tapes, the volume serial number is obtained from this field of the label and recorded in both the JFCB and the UCB.

The IEHINITT utility program can create this label with a volume serial number of up to six characters. The number is left-justified and the remainder of this field is padded with blanks.

## 4 — Reserved (1 byte)

- *Contents:* Reserved for future use—must be recorded as zero.
- *Processing:* Not used. The IEHINITT utility program writes a zero in this field.

## 5 — VTOC Pointer (10 bytes)

- *Contents:* Direct-access volumes only. This field is not used for tape volumes and should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field.

## 6 — Reserved (10 bytes)

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field.

## 7 — Reserved (10 bytes)

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field.

## 8 — Owner Name and Address Code (10 bytes)

- *Contents:* Indicates a specific customer, person, installation, department, etc., to which the volume belongs. Any code or name is acceptable.
- *Processing:* Not used or verified. The IEHINITT utility program writes the text specified by the user, and the Open and EOVS routines write the text specified by the operator. If the code is less than 10 bytes long, it is left-justified and the remainder of the field is padded with blanks.

**9 — Reserved (29 bytes)**

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field.

## Format of the IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1)

IBM standard data set label 1 is 80 characters in length and describes the associated data set. The format is used for header labels (HDR1), end-of-volume trailer labels (EOV1), and end-of-data-set trailer labels (EOF1). Data set label 1 is always followed by data set label 2. It is recorded in EBCDIC on 9-track tape units, or in BCD on 7-track tape units.

Figure 8 shows the format of data set label 1. The shaded areas represent fields that the operating system writes in the label, but that are not used or verified during processing. The contents and processing of each field of the label are described below. The processing descriptions refer to the following system control blocks:

- Communication vector table (CVT)
- Data control block (DCB)
- Data extent block (DEB)
- Job file control block (JFCB)
- Unit control block (UCB)

Location of areas within these control blocks can be found in *OS System Control Blocks*.

**1 — Label Identifier (3 bytes)**

- *Contents:* Three characters that identify the label are as follows:
 

HDR	Header label (at the beginning of a data set)
EOV	Trailer label (at the end of a tape volume, when the data set continues on another volume)
EOF	Trailer label (at the end of a data set)
- *Processing:* Data management checks this field to verify that the record is an IBM standard data set label.

For input data sets, data management checks the label identifier to determine whether data set processing is to be continued. When data management finds an EOV label, it performs volume switching. When data management finds an EOF label, it passes control to the user's end-of-data routine.

If the DD statement specifies OPTCD=B for an input data set, the trailer label identifier (EOV or EOF) is not used to determine whether a volume switch is necessary. If more volumes are available, data management performs the switching. If no volumes are available, data management passes control to the user's end-of-data routine.

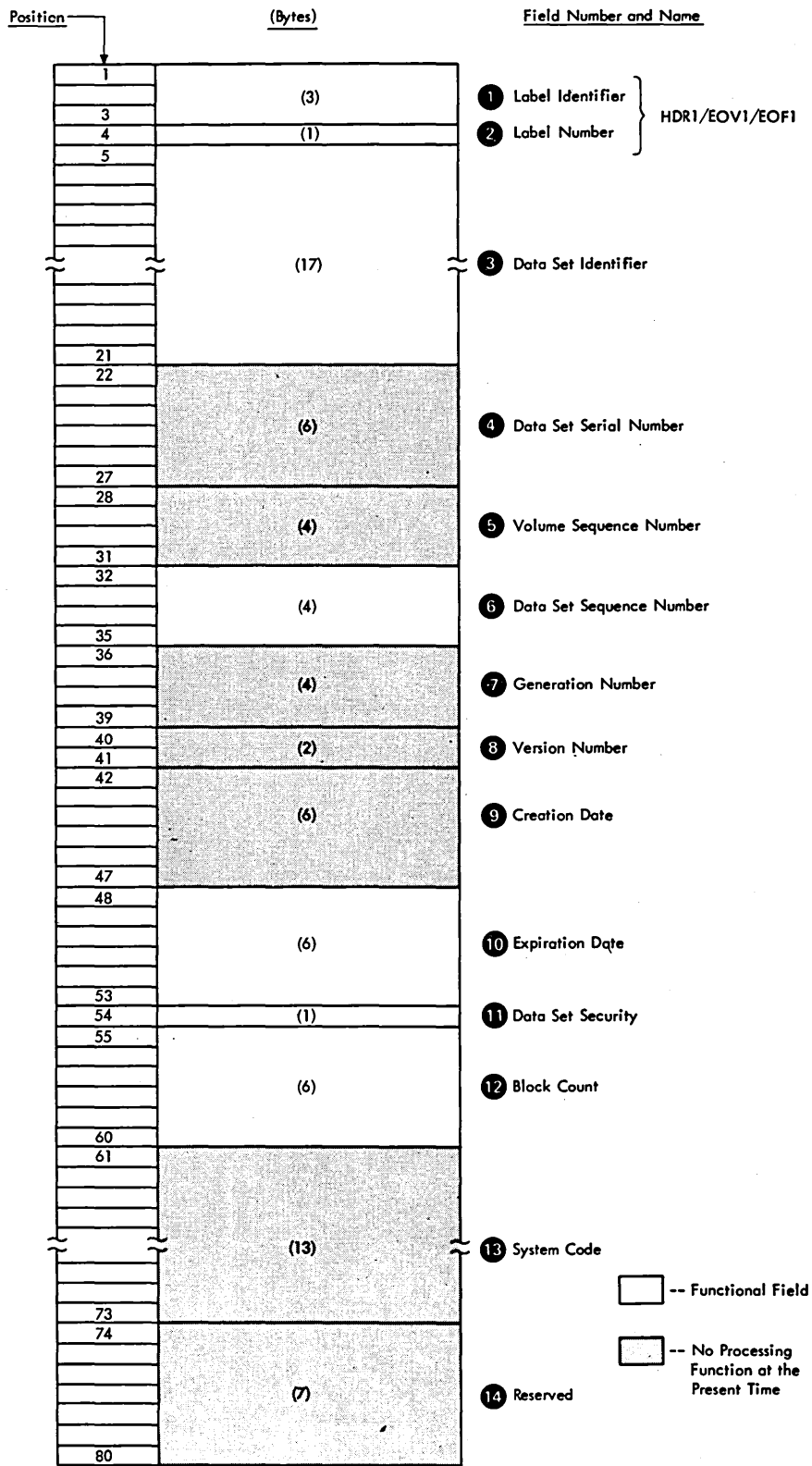


Figure 8. Format of IBM Standard Data Set Label 1

When creating trailer labels, the EOV routine writes EOV in this field, and the Close routine writes EOF.

## 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always a 1 for data set label 1.
- *Processing:* Verified and written in conjunction with Field 1 to identify this label as HDR1, EOV1, or EOF1.

## 3 — Data Set Identifier (17 bytes)

- *Contents:* The rightmost 17 bytes of the data set name (includes GnnnnVnn if the data set is part of a generation data group). If the data set name is less than 17 bytes, it is left-justified and the remainder of this field is padded with blanks. If the name contains embedded blanks or other special characters, you must enclose the name in apostrophes on the DD statement that requests this data set. The *OS Job Control Language Reference* lists the restrictions that apply to enclosing a data set name in apostrophes. The apostrophes do not appear in the data set identifier field.
- *Processing:* For input, this name is compared to the user-specified data set name found in the JFCB. This ensures that the correct data set is being processed.

For output, the data set name in the existing label is verified in conjunction with security protection to determine whether the existing data set can be overwritten. If protection is not specified, the data set name is not checked.

OS tape input files can accept generation data group members written in DOS format. That is, on input, the operating system can verify the file name field on a DOS tape, excluding the generation and version number as a level of qualification, and verify the generation and version number fields in addition to the file name field.

When creating labels for a new data set, the user-specified data set name is obtained from the JFCB and recorded in this field.

## 4 — Data Set Serial Number (6 bytes)

- *Contents:* The volume serial number of the tape volume containing the data set. For multivolume data sets, this field contains the serial number of the first volume of the aggregate created at the same time. The serial number can be any six alphameric characters, normally numeric (000001-999999). The number may be from one to six characters, but if fewer than six characters, the code must be left-justified and followed by blanks. Although all national characters, the hyphen, and other special characters are accepted when enclosed by apostrophes, their use is not recommended. This is because difficulties can occur in recognizing volume serial numbers when typewriter heads and print chains with nonalphameric characters are used.
- *Processing:* Not used or verified. When creating labels, the serial number is obtained from the UCB and recorded in this field.

## 5 — Volume Sequence Number (4 bytes)

- *Contents:* A number (0001-9999) that indicates the order of the volume within the multivolume group created at the same time. This number is always 0001 for a single volume data set.
- *Processing:* Not used or verified. When creating labels, the Open routine writes 0001 in this field; the EOV and Close routines obtain the current volume sequence number from the DEB.

## 6 — Data Set Sequence Number (4 bytes)

- *Contents:* A number (0001-9999) that indicates the relative position of the data set within a multi-data-set group. This number is always 0001 for a single data set organization.
- *Processing:* This number in the first HDR1 label on the tape is referred to when the Open routine positions the tape. If this number in the first HDR1 label and the requested data set sequence number in the JFCB are both greater than 1, the logical data set sequence number in the UCB is set to the number in the label. Otherwise, the logical data set sequence number in the UCB is set to 1.

When creating labels, the Open and Close routines obtain the user-specified data set sequence number from the JFCB (a 0 is changed to 1). The EOVS routine obtains this number from the logical data set sequence number in the UCB.

## 7 — Generation Number (4 bytes)

- *Contents:* If the data set is part of a generation data group, this field contains a number from 0001 to 9999 indicating the absolute generation number (the first generation is recorded as 0001). If the data set is not part of a generation data group, this field contains blanks.
- *Processing:* Not used or verified. The generation number is available as part of the data set name in Field 3 of this label.

When creating labels, data management checks the JFCB to determine whether the data set is part of a generation data group. If so, the generation number is obtained from the last part of the data set name in the JFCB. Otherwise, this field is recorded as blanks.

## 8 — Version Number of Generation (2 bytes)

- *Contents:* If the data set is part of a generation data group, this field contains a number from 00 to 99 indicating the version number of the generation (the first version is recorded as 00). If the data set is not part of a generation data group, this field contains blanks.
- *Processing:* Not used or verified. The version number is available as part of the data set name in Field 3 of this label.

When creating labels, data management checks the JFCB to determine whether the data set is part of a generation group. If so, the version number is obtained from the last part of the data set name in the JFCB. Otherwise, this field is recorded as blanks.

## 9 — Creation Date (6 bytes)

- *Contents:* Year and day of the year when the data set was created. The date is shown in the format byydd where:

b = blank

yy = year (00-99)

ddd = day (001-366)

- *Processing:* Not used or verified. When data management creates labels, the date is obtained from the JFCB. This is the date set by the operator at IPL when the reader that read the job was started and not necessarily the date when the label is created.

## 10 — Expiration Date (6 bytes)

- *Contents:* Year and day of the year when the data set may be scratched or overwritten. The data is shown in the format byydd where:

b = blank

yy = year (00-99)

ddd = day (001-366)

- *Processing:* For input, not used or verified. For output, the expiration date in the existing label is compared to the current date shown in the CVT. If the date in the label is greater than the current date, a message informs the operator and gives him the option of using the tape or mounting another. If any other data sets follow on the same volume, they are considered to expire on the same day.

When creating labels, data management obtains the expiration date from the JFCB. If you did not specify a retention period or expiration date, then the expiration date is recorded as zeros and the data set is considered expired.

## 11 — Data Set Security (1 byte)

- *Contents:* A code number indicating the security status of the data set is as follows:

0 No security protection.

1 Security protection. Additional identification of the data set is required before it can be read, written, or deleted.

3 Security protection. Additional identification of the data set is required before it can be written or deleted.

- *Processing:* For input, data management inspects this field on a single volume data set, on each concatenated data set, and on each volume of a multivolume data set. If protection is specified in this field, data management verifies the password furnished by the operator and sets a security indicator in the JFCB.

For output, data management inspects this field in the existing HDR1 label. If security is specified, the existing data set cannot be overwritten until data management verifies the password and the data set name in Field 3 of this label. If you specify a data set name different from the one in Field 3, and the data set is the first one on the first or only volume, the operator is requested to demount the tape and mount a scratch tape, even though you requested a specific volume. If the data set is not the first one on the volume or this is not the first volume of a multivolume data set, the job is abnormally terminated.

When data management creates labels, the user's request for security is determined from the indicator in the JFCB.

## 12 — Block Count (6 bytes)

- *Contents:* This field in the trailer label shows the number of data blocks in the data set on the current volume. This field in the header label is always zero (000000).
- *Processing:* The DCB count is incremented as the data set is read. The final DCB count is compared with the count in the trailer label at end of data or end of volume. If the counts do not agree, a user exit entry in the DCB exit list determines whether processing will continue or abnormally terminate. If the appropriate user exit entry is not provided, a block count discrepancy causes processing to abnormally terminate.

For read backward, the verification process is reversed. The trailer label count is recorded in the DCB and decremented as the data set is read. The final DCB count should be zero, which equals the count in the header label.

When data management creates labels, the block count in the header label is set to zeros. The block count in the trailer label is obtained from the DCB.

### 13 — System Code (13 bytes)

- *Contents:* A unique code that identifies the system.
- *Processing:* Not used or verified. When creating labels, the code is obtained from the JFCB where it is always binary zeros.

### 14 — Reserved (7 bytes)

- *Contents:* Reserved for future use—contains blanks.
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field.

## Format of the IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2)

IBM standard data set label 2 always follows data set label 1 and contains additional information about the associated data set. The format is used for header labels (HDR2), end-of-volume trailer labels (EOV2), and end-of-data-set trailer labels (EOF2). The label is 80 characters in length. It is recorded in EBCDIC on 9-track tape units, or in BCD on 7-track tape units.

Figure 9 shows the format of data set label 2. The shaded areas represent fields that the operating system writes in the label, but that are not used or verified during processing. The processing descriptions refer to the following system control blocks:

- Data control block (DCB)
- Job file control block (JFCB)
- Task input/output table (TIOT)
- Unit control block (UCB)

Location of areas within these control blocks can be found in *OS System Control Blocks*.

### 1 — Label Identifier (3 bytes)

- *Contents:* Three characters that identify the label are as follows:

HDR	Header label (at the beginning of a data set)
EOV	Trailer label (at the end of a tape volume, when the data set continues on another volume)
EOF	Trailer label (at the end of a data set)
- *Processing:* Data management checks this field to verify that the record is an IBM standard data set label.

For input data sets, data management checks the label identifier to determine whether data set processing is to be continued. When data management finds an EOV label, it performs volume switching. When data management finds an EOF label, it passes control to the user's end-of-data routine.

If the DD statement specifies OPTCD=B for an input data set, data management accepts either EOV or EOF as the trailer label identifier, and the identifier is not



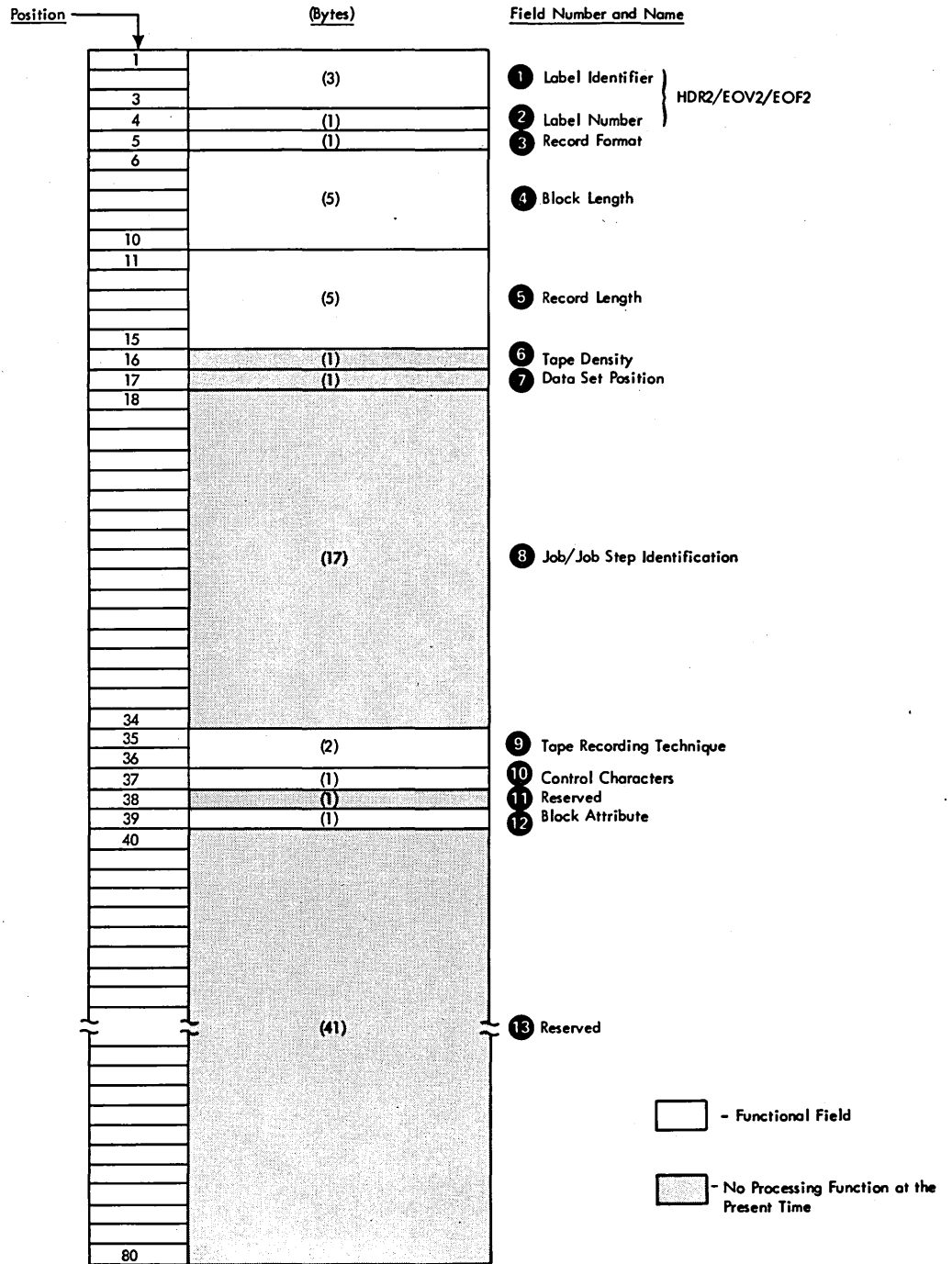


Figure 9. Format of IBM Standard Data Set Label 2

used to determine whether a volume switch is necessary. If more volumes are available, data management performs the switching. If no volumes are available, data management passes control to the user's end-of-data routine.

When creating trailer labels, the EOv routine writes EOv in this field, and the Close routine writes EOF.

## 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always a 2 for data set label 2.
- *Processing:* Verified and written in conjunction with Field 1 to identify this label as HDR2, EO2, or EOF2.

## 3 — Record Format (1 byte)

- *Contents:* An alphabetic character that indicates the format of the records in the associated data set:
  - F Fixed length
  - V Variable length
  - U Undefined length
- *Processing:* For input, the record format is obtained from this label and recorded in the JFCB (if the JFCB field is zero). Then the record format in the JFCB is recorded in the DCB (if the DCB field is zero). Note that this is a merging process in which existing specifications in the JFCB and DCB cannot be overridden.

When creating labels, a reverse merge follows the forward merge described above. The record format in the DCB overrides the record format in the JFCB, and the updated JFCB provides the information for the label.

This merging process is explained and illustrated in the introduction to this manual.

## 4 — Block Length (5 bytes)

- *Contents:* A number up to 32760 that indicates the block length, in bytes. Interpretation of the number depends on the associated record format in Field 3, as follows:
  - Format F Block length (must be a multiple of the logical record length in Field 5)
  - Format V Maximum block length (including the 4-byte length indicator in the blocks)
  - Format U Maximum block length
- *Processing:* The number in the label is converted to binary and merged with appropriate fields in the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

## 5 — Record Length (5 bytes)

- *Contents:* A number that indicates the record length, in bytes. Interpretation of the number depends on the associated record format in Field 3, as follows:
  - Format F Logical record length
  - Format V Maximum logical record length (including the 4-byte length indicator in the records)
  - Format U Zeros
- *Processing:* The number in the label is converted to binary code and merged with the appropriate fields in the JFCB and DCB. The merging process is the same as for the record format code in Field 3 of this label.

## 6 — Tape Density (1 byte)

- *Contents:* A code indicating the recording density of the tape, as follows:

DEN Value	Recording Density			
	7-Track	9-Track	9-Track (phase encoded)	9-Track (dual density)
0	200	—	—	—
1	556	—	—	—
2	800	800	—	800 <sup>1</sup>
3	—	—	1600	1600 <sup>2</sup>

1. Non-return-to-zero-inverted (NRZI) mode

2. Phase encoding (PE) mode

Note: For 7-track 3400 models, if DEN=0 was specified in the DCB macro or the DCB parameter of the DD statement, the recording density will be 556 bits-per-inch, but this field will indicate 200 bits-per-inch recording density.

- *Processing:* Not used or verified. When data management creates labels, the information for this field is obtained from the JFCB.

#### 7 — Data Set Position (1 byte)

- *Contents:* A code indicating a volume switch is as follows:

0 No volume switch has occurred

1 A volume switch previously occurred

- *Processing:* Not used or verified. When creating labels, the Open routine writes 0 in this field, and the EOVS routine writes 1. The Close routine determines which code to write by comparing the volume serial number in the JFCB to the number in the UCB. It writes 0 if the numbers are equal, and 1 if they are not equal.

#### 8 — Job/Job Step Identification (17 bytes)

- *Contents:* Identification of the job and job step that created the data set. The first 8 bytes contain the name of the job, the ninth byte is a slash (/), and the final 8 bytes contain the name of the job step.
- *Processing:* Not used or verified. When data management creates labels, the names of the job and job step are obtained from the TIOT.

#### 9 — Tape Recording Technique (2 bytes)

- *Contents:* A code or blanks indicating the tape recording technique used to create the data set (7-track tape only):

Tb Odd parity with translation

Cb Odd parity with conversion

Eb Even parity with no translation

ET Even parity with translation

bb Odd parity with no translation or conversion

This field is recorded as blanks for 9-track tape. The only technique available for 9-track tape is odd parity and no translation.

- *Processing:* For seven-track tape, the specification in the label is converted to a bit code and merged with the appropriate fields of the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

#### 10 — Control Characters (1 byte)

- *Contents:* A code indicating whether a control character set was used to create the data set and the type of control characters used:

A Contains ASCII control characters

M Contains machine control characters

b Contains no control characters

- *Processing:* The specification in the label is converted to a bit code merged to the appropriate fields of the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

**11 — Reserved (1 byte)**

- *Contents:* Reserved for future use—recorded as blanks.
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field.

**12 — Block Attribute (1 byte)**

- *Contents:* A code indicating the block attribute used to create the data set:
  - B Blocked records
  - S Spanned records
  - R Blocked and spanned records
  - b No blocked and no spanned records
- *Processing:* The specification in the label is converted to a bit code and merged with the appropriate fields of the JFCB and DCB. The merging process is the same as for the record format code in Field 3 of this label.

**13 — Reserved (41 bytes)**

- *Contents:* Reserved for future use—recorded as blanks.
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field.

## Format of User Label (UHL1-UHL8, UTL1-UTL8)

IBM Standard user labels contain user-specified information about the associated data set. User labels are optional within the standard label groups.

Figure 10 shows the format of user labels. The format is used for user header labels (UHL1-UHL10) and user trailer labels (UTL1-UTL8). The labels are 80 characters in length. They are recorded in EBCDIC on nine-track tape units, or in BCD on seven-track tape units. The contents and processing of each field of the label are described below.

**1 — Label Identifier (3 bytes)**

- *Contents:* Three characters that identify the label are as follows:
  - UHL User header label (at the beginning of a data set)
  - UTL User trailer label (at the end-of-volume or end-of-data-set)
- *Processing:* This field is read to verify that the record is a user label. Data management accepts either UHL or UTL.

**2 — Label Number (1 byte)**

- *Contents:* The relative position of this label within a set of labels of the same type; it can be a number from 1 to 8.
- *Processing:* This field is read to ensure that no more than eight user labels are processed. This field is read in conjunction with Field 1.

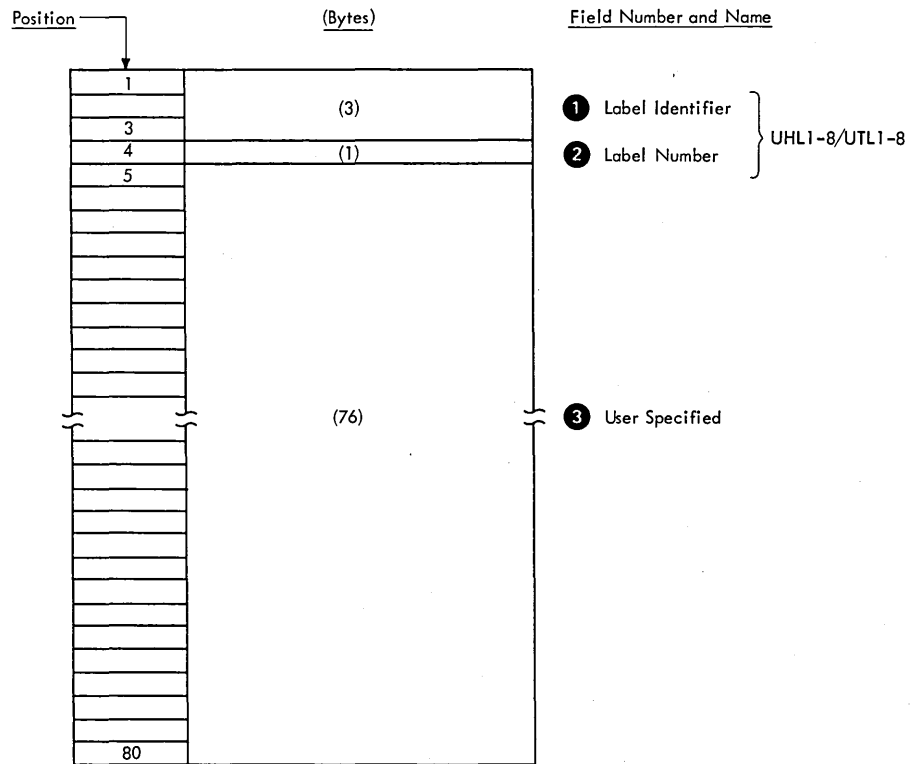


Figure 10. Format of User Label

**3 — User Specified (76 bytes)**

- *Contents:* Specified by the user.
- *Processing:* Specified in the DCB exit list.



# AMERICAN NATIONAL STANDARD LABELS

This section describes the organization, format, and content of American National Standard labels as defined by the American National Standards Institute (ANSI) in the *American National Standard Magnetic Tape Labels for Information Interchange, ANS X3.27—1969*.

Support for the American National Standard Code for Information Interchange (ASCII) is currently limited to magnetic tape. Operating system support consists of processing labels and translating from ASCII to EBCDIC on input and from EBCDIC to ASCII on output.

## Label Definition and Organization

American National Standard labels are generally similar to IBM standard labels. The principal differences are:

- An unlimited number of ANSI user labels may be placed at the beginning and end of a file.
- The formats of the ANSI labels VOL1, HDR2, EOF2, and EO2 are slightly different from the formats of the corresponding IBM labels.
- The ANSI labels HDR2, EOF2, and EO2 are optional. These labels are required for IBM standard labels.

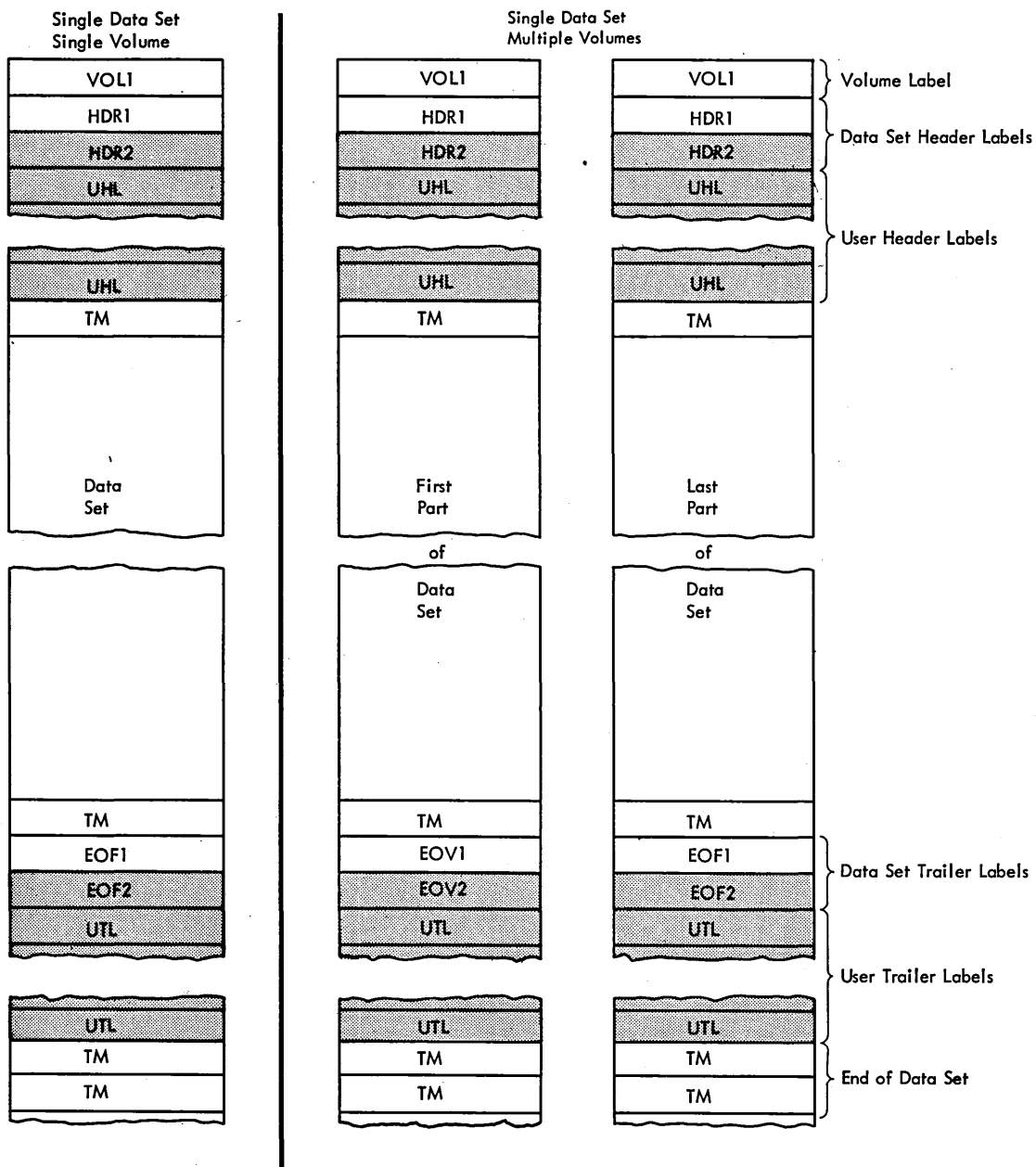
ANSI labels fully supported by the operating system are 80-character records that are recorded in ASCII on 9-track 800 BPI NRZI tape using odd parity. Because ASCII is designed for interchange of data among computers of various manufacturers, it cannot be assumed that these conditions will always be met. Therefore, the operating system also reads ANSI labels that are longer than 80 characters. Only the first 80 characters, however, are used for label checking and verification. The operating system also reads and writes ANSI labels in other densities. All ANSI labels must be recorded in ASCII; mixed-mode labels in ASCII and EBCDIC, or ASCII and some other code, are not supported. Unlabeled tapes recorded in ASCII can be processed, as explained in the "Unlabeled Tapes" section.

The first four characters of an ANSI tape label are always used to identify the label:

Label Identifier	Label Description
VOL1	Volume label
HDR1 and HDR2	Data set header labels
EOV1 and EO2	Data set trailer labels (end of volume)
EOF1 and EOF2	Data set trailer labels (end of data set)
UHLn <sup>1</sup>	User header labels (unlimited number permitted)
UTLn <sup>1</sup>	User trailer labels (unlimited number permitted)

1. The fourth character in ANSI user header and trailer labels is not checked by the operating system. It can be any ASCII character except the substitute character.

Figures 11 and 12 show the position of the labels with various tape volume organizations. A tape with ANSI labels must contain a volume label and data set labels. User labels (UHL, UTL) are optional.



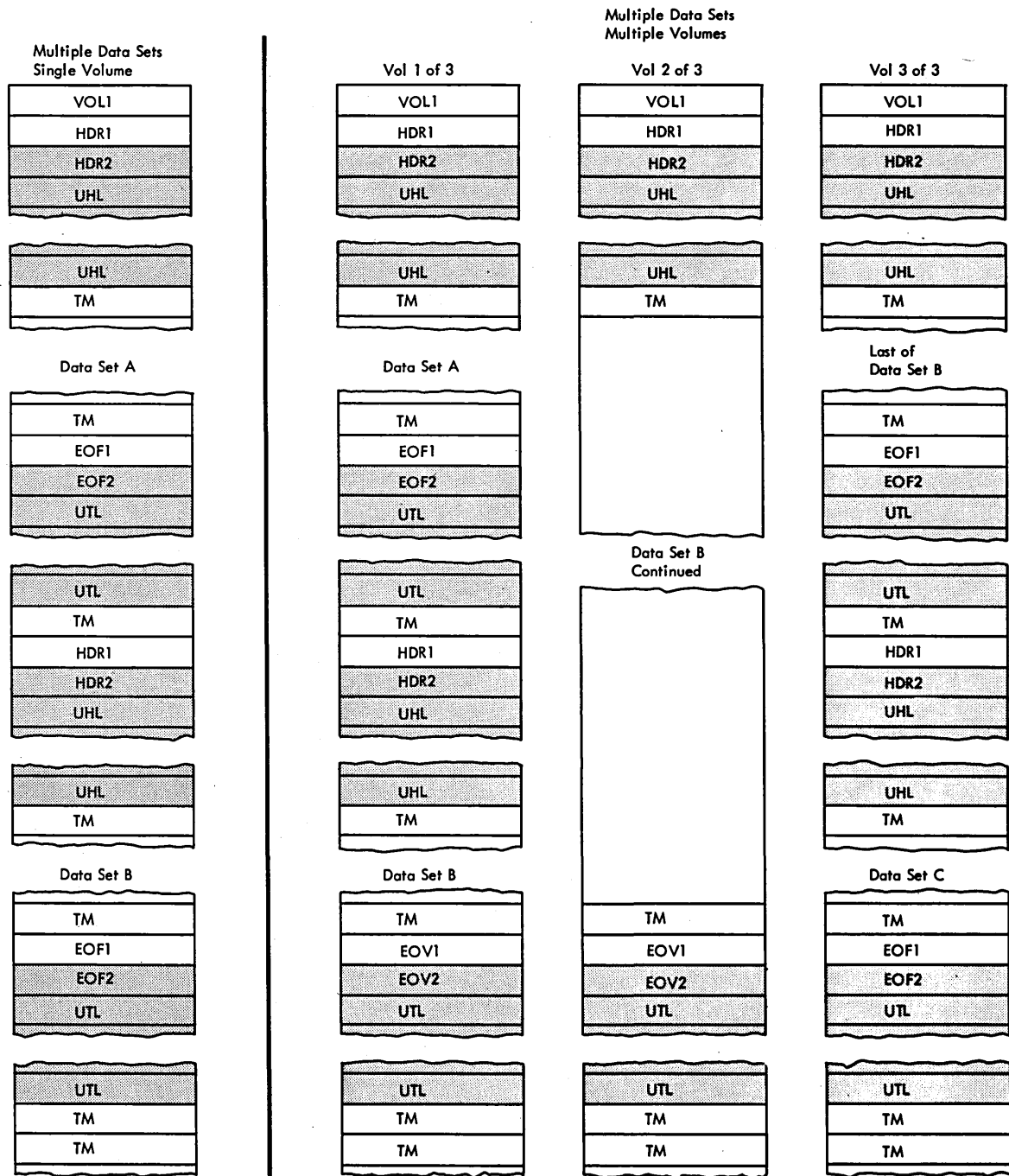
Single Data Set/Single Volume: The volume label is followed by the data set header labels and optional user header labels. The data set is preceded and followed by a tapemark. The data set trailer labels are identified as EOF and followed by optional user trailer labels. Two tapemarks follow the trailer label group to indicate that the data set is the last data set on the volume and is not continued on another volume.

Single Data Set/Multiple Volumes: More than one volume is needed to contain the data set. The last volume is organized the same as a single volume. On the other volumes, the data set trailer labels are identified as EOVS instead of EOF, and the trailer label group is followed by two tapemarks. The data set and user labels are repeated on each volume, and there is a separate volume label for each tape.

Note: Shading indicates optional labels for ANSI labeled tapes.

Figure 11. Volume Organization with ANSI Standard Labels (Single Data Set)





**Multiple Data Sets/Single Volume:** The tape begins with a volume label. Each data set is preceded by a header label group and a tapemark, and is followed by a tapemark and a trailer label group. The data set trailer labels are identified as EOF. Each trailer label group is followed by a tapemark; the trailer label group for the last data set on the volume is followed by two tapemarks.

**Multiple Data Sets/Multiple Volumes:** More than one volume is needed to contain the multiple data set aggregate. The last volume is organized the same as a multiple data set/single volume layout. On the other volumes, the last data set trailer labels are identified as EOF instead of EOF, and the last trailer label group is followed by two tapemarks. There is a separate volume label for each tape.

**Note:** Shading indicates optional labels for ANSI labeled tapes.

Figure 12. Volume Organization with ANSI Standard Labels (Multiple Data Sets)

## ***American National Standard Volume Label***

The American National Standard volume label (VOL1) is the first record on an ASCII tape. It is at least 80 characters long; any additional characters are ignored by data management for volume label processing.

The volume label identifies the volume and its owner. It is used by data management to verify that the correct volume is mounted.

When a tape is first received at an installation, the volume label is created by a utility program or a user program.

## ***Data Set Header Labels***

ANSI standard data set header labels consist of data set label 1 (HDR1) and, in some cases, data set label 2 (HDR2). The ANSI HDR1 label is created automatically by data management routines each time a data set is recorded on tape. A HDR1 label identifies and describes the data set and protects it from unauthorized use.

The ANSI HDR1 label is at least 80 characters long; any additional characters are not used by data management for label checking and verification.

A HDR2 label is optional for ANSI standard labeled tapes. The operating system produces an ANSI HDR2 label for output tapes; such a label on an input tape is treated similarly to an IBM HDR2 label. If the HDR2 label is produced by another system, only the fields that conform to ANSI standards are processed by data management.

## ***Data Set Trailer Labels***

The data set trailer label 1 (EOV1, EOF1) is required for ANSI standard labeled tapes. The ANSI standard trailer label 1 duplicates the ANSI standard header label 1 so that the tape can be read backward. A trailer label 1 is identical to a header label 1 except that:

- The identifier is EOV or EOF instead of HDR.
- A block count is recorded in the first trailer label (EOV1 or EOF1) and is used on input to verify that all blocks of the data set are processed. The block count field in the HDR1 label contains zeros.

ANSI standard EOV1 and EOF1 labels are created automatically by data management when the data set is recorded on tape. EOV1 and EOF1 labels are at least 80 characters long; any additional characters are not used for checking or verification.

The data set trailer label 2 (EOV2, EOF2) is optional for ANSI standard labeled tapes. ANSI EOV2 and EOF2 labels are produced by the operating system for ANSI standard labeled tapes. These labels are treated similarly to IBM EOV2 and EOF2 labels. If the ANSI EOV2 or EOF2 labels are produced by another system, only the fields that conform to ANSI standards are processed by data management.

## ***User Header and Trailer Labels***

User labels contain user-specified information about data sets. User labels (UHL, UTL) are optional for ANSI standard labeled tapes, and there is no limit to the number of ANSI user labels that can be associated with a data set.

## **Tapemarks**

The tapemark requirements for ASCII interchange tapes differ slightly from those of EBCDIC tapes. On ASCII interchange tapes, a multivolume file will have two tapemarks placed after the EOVS label group; EBCDIC tapes have only a single tapemark for the corresponding situation. Apart from this difference, the requirements for ASCII and EBCDIC tapemarks are the same:

- Each header label group must be followed by a tapemark which indicates the beginning of the data to be processed.
- Each data set must be followed by a tapemark which indicates the beginning of the trailer label group.
- If the data set is the last one on the volume, each trailer label group must be followed by two tapemarks. Otherwise, one tapemark is required.

When the operating system is used to create a data set with ANSI standard labels, data management routines write the necessary tapemarks.

## **ANSI Standard Label Processing**

Label processing is handled by the I/O support routines of data management (Open, EOVS, and Close). This processing consists of four basic functions.

- Translating input labels from ASCII to EBCDIC and translating output labels from EBCDIC to ASCII.
- Checking the labels on input tapes to ensure that the correct volume is mounted, and to identify, describe, and protect the data set being processed.
- Checking the existing labels on output tapes to ensure that the correct volume is mounted, and to prevent overwriting of vital data.
- Creating and writing new labels on output tapes.

These processing functions are summarized in Figure 13. The table shows the specific labels that are processed for each function and which routines perform the functions. The summary in Figure 13 is the general basis for the discussions of label processing that follow in this section.

Because ASCII is designed for interchange of data among computers of various manufacturers, it cannot be assumed that all tape labeling practices will be identical. The operating system, therefore, accepts labels longer than 80 characters, those with densities other than 800 BPI, and those that are not supported by the operating system on output (user volume header labels, HDR3-HDR9, EOVS3-EOVS9, and EOF3-EOF9).

When a data set is opened for input, the volume label and the header labels are processed. For an input end-of-data condition, the trailer labels are processed, unless deferred user input trailer label processing is specified in the DCB exit list. For an input end-of-volume condition, the trailer labels on the current volume are processed, and then the volume label and header labels on the next volume are processed. (When the FEOVS macro instruction is issued for an input tape, the trailer labels on the current volume are not processed, but the volume labels and header labels on the next volume are processed.) No label processing is performed when an input data set is closed, unless deferred user input trailer label processing is specified in the DCB. If deferred user input trailer label processing is specified, the processing otherwise performed for an input end-of-data-set condition is performed when an input data set is closed.

When a data set is opened for output, the existing volume label and HDR1 label are checked, and a new volume label and new header labels are written. For an output

Processing	Volume Label		Header Labels <sup>1</sup>				Trailer Labels <sup>1</sup>			
	VOL1	USER VOLUME LABELS	HDR1	HDR2	HDR3-9	UHL	EOF1 or EOVI	EOF2 or EOVI	EOF3-9 or EOVI3-9	UTL
First or only volume <sup>2</sup> : Checks labels on input tape.	Open	ignored	Open	Open	ignored	Open	EOV	bypassed	ignored	EOV
Checks existing labels on output tape before overwriting.	Open	ignored	Open	not read	not read	not read	not read	not read	not read	Open <sup>5</sup>
Writes new labels on output tape.	Open, or user <sup>4</sup>	not written	Open	Open	not written	Open	Close or EOVI	Close or EOVI	not written	Close or EOVI
Second or subsequent volumes <sup>3</sup> : Checks labels on input tape.	EOV	ignored	EOV	bypassed	ignored	EOV	EOV	bypassed	ignored	EOV
Checks existing labels on output tape before overwriting.	EOV	ignored	EOV	not read	not read	not read	not read	not read	not read	not read
Writes new labels on output tape.	EOV or user <sup>4</sup>	not written	EOV	EOV	not written	EOV	Close or EOVI	Close or EOVI	not written	Close or EOVI

Notes:

- For read backward operations, the action on header and trailer labels is reversed.
- Includes the first volume of concatenated data sets with unlike characteristics. (Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program. Any exception in processing makes the data sets unlike.)
- Includes the first volume of concatenated data sets with like characteristics.
- User creates the label with the IEHINIT utility program or a user program.
- If DISP-MOD is specified on the DD statement, the Open routine positions the tape at the end of the existing data set and allows an input user trailer label routine to process user trailer labels (before overwriting the existing labels).

Figure 13. ANSI Standard Label Processing by Data Management Routines

end-of-volume condition (including FEOV), trailer labels are written on the current volume, the existing volume and header labels on the next volume are checked, and then a new volume label and new header labels are written on the next volume. When an output data set is closed, trailer labels are written.

## Opening an Input Data Set

If ANSI standard labels are specified, the first record on the input tape must be an ANSI standard volume label (VOL1). At the time the data set is opened, data management checks the first record on the tape to determine whether the record is at least 80 bytes long and contains the identifier VOL1 in the first 4 bytes. If the record is over 80 bytes long, the excess characters are ignored.

## ***Volume Serial Number***

Data management uses the VOL1 label to ensure that the correct tape is mounted. The volume serial number in the label is compared to the volume serial number that you specify. You can specify the serial number either directly in the DD statement or indirectly through the catalog facility. Serial numbers are required when the processing method is INPUT, INOUT, or RDBACK.

If the volume serial number is correct, data management resets a mount switch in the unit control block (UCB) to indicate that volume mounting is verified (the switch is initially set when the mount message is issued to the operator). If the serial number is not correct, data management rejects the tape and issues another mount message.

## ***Positioning the Volume to the Data Set***

When the volume is mounted and verified, data management positions the tape in front of the header label group of the data set to be processed. Usually there is only one data set on the reel, and the header label group immediately follows the volume label.

To retrieve a data set when more than one data set is on a simple reel of tape, you specify a data set sequence number in the LABEL parameter of the DD statement, unless the data set is cataloged. You need not specify a data set sequence number for a cataloged data set, because the number can be obtained from the catalog along with the volume serial number.

- The sequence number can be from 1 to 9999, with 1 representing the first data set on the volume. If you specify a sequence number higher than the number of data sets on the volume, the tape is spaced through and removed from its reel.
- If the data set is not cataloged and you do not specify a sequence number or you specify 0, data management assumes that the data set is the first in sequence on the volume.

To position the tape, data management uses the requested data set sequence number shown in the JFCB and the data set sequence number shown in the first HDR1 label on the tape, and maintains a logical data set sequence number in the unit control block (UCB). The number in the UCB represents the current position of the tape, and is maintained as follows:

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened the Open routine sets the data set sequence number in the UCB to 1. The exceptions are:
  - If the tape is still positioned from previous processing, such as for a LEAVE request, the Open routine does not reset the number in the UCB.
  - If the data set sequence number in the JFCB and the data set sequence number in the first HDR1 label on the tape are both greater than 1, the Open routine sets the data set sequence number in the UCB to the value of the number in the first HDR1 label. (The data set sequence number in the first HDR1 label may be greater than 1 when the volume is part of a multiple-data-set/multiple-volume aggregate.)
3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the Open routine adjusts the data set sequence number in the UCB as the tape is spaced past each data set, until the number in the UCB equals the number in the JFCB.

4. When multiple tape units are used and a volume switch causes processing to be continued on a volume on a different unit, the EOVR routine copies the data set sequence number from the previous UCB to the current UCB.

Only one data set on a tape volume may be open at any given time. An attempt to begin processing of a second data set on the same volume results in abnormal termination.

When the tape is positioned to the data set header label group of the first data set or the requested data set, data management checks the label identification. If the identifier HDR1 is not found, processing is abnormally terminated.

### ***Data Set Name***

To ensure that the correct data set is being opened, data management compares the data set name shown in the HDR1 label of the requested data set to the data set name specified in the DD statement. This comparison is made on only the 17 least significant (rightmost) characters of the data set name (including 8 characters for the generation and version numbers if the data set is part of a generation data group).

If the comparison shows an incorrect data set name, processing is abnormally terminated.

### ***Expiration Date***

The expiration date shown in the HDR1 label is not verified for input data sets, unless the processing method is INOUT. For INOUT, if the expiration date has not been reached, data management notifies the operator and asks him to confirm use of the tape. If confirmation is not received, processing is abnormally terminated. (If you override the INOUT specification by coding LABEL=(,IN) on the DD statement, the expiration date is not verified.)

### ***Accessibility***

An indicator in the HDR1 label shows whether the data set is protected against unauthorized use. If accessibility is specified, data management asks the operator or TSO terminal user to key in the correct password. If an ASCII character other than a space, a 1, or a 3 is found, no further processing takes place. A space character indicates unlimited access. A 1 or a 3 indicates a password-protected data set created by the operating system.

The password is verified in a user-established password data set. This password data set contains the data set name, the password, and a protection mode indicator. The protection mode indicator is set to permit either read/write or read-only operations. Processing is terminated if:

- The operator or TSO terminal user, in two attempts, does not supply the correct password.
- The password record for the data set to be opened does not exist in the password data set.
- The read-only protection mode is specified for INOUT or OUTIN processing.

*OS Data Management for System Programmers* describes the protection feature in detail and contains the information you need to create and maintain the password data set.

## ***Block Count***

The block count shown in the HDR1 label is always an ASCII 0. This 0 is recorded in the data control block and incremented during processing for comparison to the block count shown in the trailer label (EOV1 or EOF1).

For reading backward, the block count shown in the trailer label (EOV1 or EOF1) is recorded in the data control block and decremented during processing for comparison to the 0 block count in the HDR1 label.

The block count is verified at end of data set or end of volume.

## ***Data Set Characteristics***

If present, the HDR2 label immediately follows the HDR1 label. Because the HDR2 label is optional with ASCII interchange tapes, and because the HDR2 format may vary if produced by systems of other manufacturers, the manner of processing HDR2 labels depends on whether the HDR1 label specifies "OS360" as the name of the system producing the tape. If "OS360" is specified, then HDR2 contains the same data as an IBM standard HDR2 label, and is processed as an IBM standard HDR2 label. If "OS360" is not specified, then the data in the 35 bytes reserved for use by the operating system is unknown and is ignored.

Unless user header labels are to be processed, data management spaces forward past the tapemark immediately after processing the HDR2 label. All labels that follow the HDR2 label are bypassed and the tape is positioned at the first data set record.

## ***User Header Labels***

User header labels (UHL) may follow the HDR2 label. To make the user header labels available to your program, AUL must be coded on the DD statement and the address of an input user header label routine must be specified in the DCB exit list. If you omit one of these parameters, data management forward spaces past the tapemark immediately after processing the HDR2 label, or the HDR1 label if the HDR2 label is not present.

## ***Read Backward***

For the read backward (RDBACK) processing method, data management uses the data set's trailer labels as header labels, and vice versa. Each label group is read in the normal sequence, that is, EOF1 before EOF2, etc. The data records, however, are read in reverse sequence.

Multivolume data sets can be read backward. Concatenated data sets and Format-D (variable-length) records cannot be read backward.

## **End of Data or End of Volume on Input**

Data management's EOV routine handles both end-of-data-set and end-of-volume conditions on input. These conditions occur when:

- A tapemark is read.
- An FEOV (force-end-of-volume) macro instruction is executed by the processing program.

After encountering a tapemark, data management checks the first 4 bytes of the first trailer label for the identifier EOV1 or EOF1. If neither identifier is found, processing is abnormally terminated. When the FEOV macro instruction is executed, the trailer labels are not checked.

## ***Block Count***

To verify that all records on the input data set on the current volume have been read, data management compares the block count shown in the first trailer label (EOV1 or EOF1) against the block count that was accumulated in the data control block. For reading backward, data management compares the 0 block count shown in the HDR1 label against the block count in the data control block.

If the block count in the label does not equal the block count in the data control block, the EOV routine gives control to the appropriate entry in the user's DCB exit list. This entry in the exit list is identified by the code 0B (hexadecimal). The EOV routine passes the following information to the exit routine:

Register 0      The block count shown in the label  
Register 1      The address of the data control block

After your exit routine analyzes the discrepancy (and possibly prints a message), your exit routine must return to the EOV routine with one of the following return codes in register 15:

0    Abnormally terminate with completion code 237 (hexadecimal)  
4    Continue processing

If you do not provide the appropriate user exit entry in the DCB exit list, a block count discrepancy causes processing to abnormally terminate with a completion code of 237 (hexadecimal).

When the FEOV macro instruction is executed, the block count is not verified.

## ***The EOV2/EOF2 Label***

Data management ignores the second trailer label (EOV2 or EOF2) of an input data set, except when it is used as a header label for a read backward operation.

## ***User Trailer Labels***

If user trailer labels (UTL) are present on input, data management can make them available to your program. To make them available, AUL must be coded on the DD statement and the address of an input user trailer label routine must be specified in the DCB exit list.

## ***Determining Volume Switch***

For a multivolume input data set, you must specify the serial numbers of all the volumes to be processed. The serial numbers are specified either directly in the DD statement or indirectly through the catalog procedure. You specify the serial numbers in forward sequence, regardless of whether the tapes are to be read forward or backward.

- For noncataloged data sets, you specify the volume serial numbers in the VOLUME parameter of the DD statement. Data management processes the group of volumes in whatever order you specify and processes only the volumes you specify.
- For cataloged data sets, the group of volumes must be processed in sequential order. However, you can begin processing at any volume of the group by specifying a volume sequence number in the VOLUME parameter of the DD statement.

For input, the label identifier of the trailer labels determines whether data management continues processing the data set. When data management finds an EOV label, it performs volume switching. When data management finds an EOF label, it passes control



to the user's end-of-data routine, with one exception: if the DD statement specifies OPTCD=B, data management performs volume switching.

To determine whether additional volumes are required, data management maintains a volume sequence number in the data extent block (DEB) in storage.

- For read forward operations, the volume sequence number in the DEB is incremented as each volume is processed. This count is compared to the total number of volumes requested, as shown in the JFCB.
- For read backward operations, the volume sequence number in the DEB is initialized to the total number of volumes requested, as shown in the JFCB. The DEB count is decremented as each volume is processed, until the count reaches 0.

If another volume is not required (end-of-data-set condition), control is given to the user's end-of-data-set routine that is specified in the data control block. Subsequently, the processing program or the operating system closes the data set.

- The user's end-of-data-set routine is not entered until the last specified volume or the last concatenated data set is processed.
- If an input data set is closed before the end of the data set is reached, the user's end-of-data-set routine is not entered.

If another volume is required (end-of-volume condition), data management obtains the next volume serial number from the JFCB and performs volume switching. If the new volume is not already mounted, the EOVRoutine issues a mount message to the operator.

When multiple tape units are being used, the EOVRoutine also checks to see if there is a next-plus-one volume specified, and if the volume just completed can be rewound and unloaded. If so, the EOVRoutine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid—the next-plus-one volume label is not verified at this time.

### ***Checking the Next Volume***

When volume switching is performed for multiple volume input, the EOVRoutine checks the volume and header labels on the new volume.

The VOL1 label is checked as if it were the first volume of the group that is, the volume serial number is verified to ensure that the correct volume is mounted.

The method of locating and checking the HDR1 label varies according to the situation. The processing depends on whether the data set is a continuation of a multivolume data set or is a concatenated data set with like characteristics. (Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program.)

- *Multivolume data set:* The data set sequence number is irrelevant for the second and subsequent volumes of a multivolume data set. The EOVRoutine assumes that the data set continues at the beginning of the new volume and, therefore, checks the first header label group on the tape. The HDR1 label is checked in the same manner as when the data set was opened on the first volume. If the data set name is not the same, or the password was not verified for each volume, processing is abnormally terminated.
- *Concatenated data sets:* The EOVRoutine handles concatenated data sets with like characteristics. Such data sets are not necessarily the first on the volume, so the EOVRoutine positions the tape according to the specified data set sequence number. This positioning is the same as for opening a data set. The HDR1 label is checked as it was

when the first data set was opened, including password verification if protection is indicated.

The HDR2 label on the new volume is not processed. The data set characteristics that were established when the data set was opened apply to all subsequent volumes handled by the EOVR routine.

The data set's block count is not accumulated from volume to volume. It is initialized and verified separately for each volume.

## Closing an Input Data Set

The Close routine does not process trailer labels on an input data set. Usually, the trailer labels are processed by the EOVR routine before the data set is closed unless deferred user input trailer label processing was specified in the data control block (DCB) exit list. If deferred user input trailer label processing was specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

If an input data set is closed before it reaches the end of data set or the end of volume, or if the FEOVR macro instruction is executed, processing of trailer labels is omitted.

## Creating a Volume Label

The VOL1 label is usually created by the IEHINITT utility program or a user's program when the reel of tape is first received at the installation. At that time, a permanent volume serial number is assigned to the reel, physically posted on the reel, and recorded in the VOL1 label.

The IBM-supplied utility program IEHINITT writes the following labels in ASCII code:

1. A volume label (VOL1) with the volume serial number and owner identification that you specify. You cannot specify any other fields of the VOL1 label.
2. A dummy header label (HDR1) with an ASCII blank (space) in Field 11 (accessibility) and ASCII zeros in the other fields.
3. A tapemark.

A tape initialized with these labels should not be confused with an ANSI standard interchange tape, which requires at least one file.

The IEHINITT utility program can write a volume label on a labeled, unlabeled, or blank tape—it makes no checks to see what data, if any, exists on the tape. Detailed procedures for using the program are contained in *OS Utilities*.

Methods other than the IEHINITT utility program can be used to write volume labels. You can use a card-to-tape program, or you can replace the IBM-supplied volume label editor routine (see "Volume Label Verification and Volume Label Editor Routines") with one that writes volume labels. If you use an editor routine to write the volume label, some data or a tapemark should already exist on the tape; otherwise, data management reads through the entire reel of blank tape looking for a label.

The VOL1 label is rewritten (that is, reproduced) by the Open or EOVR routines if all the following conditions are met:

- OUTPUT or OUTIN is specified in the OPEN macro instruction.
- The tape is positioned to the first data set on the volume. (The tape cannot be blank.)
- The data set is not security-protected.

All VOL1 labels that can be successfully read under these conditions are rewritten.

If you request an ANSI standard labeled (AL, AUL) output volume and the tape that you are allocated is recorded in the wrong density and cannot be read, the VOL1 label is rewritten in the density that you specify. This allows you to make nonspecific requests for output tapes (that is, you need not specify a volume serial number in your DD statement) and allows the operator to mount any available scratch tape to satisfy your request.

If you make a nonspecific output volume request for a standard labeled (AL, AUL) tape and the mounted volume is an NL or NSL labeled tape, the Open or EOVS routine creates a volume label (VOL1) and sends a message to the console operator requesting serial number and owner information (see Figure 14, Fields 3 and 7).

## **Opening an Output Data Set**

If ANSI standard labels are specified, the first existing record on the output tape must be an ANSI standard volume label (VOL1). At the time the data set is opened, data management checks the first existing record on the tape to ensure that the record is at least 80 bytes long and contains the identifier VOL1 in the first 4 bytes. The various error conditions that can occur during verification of the first record are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

### ***Volume Serial Number***

You need not specify a volume serial number for output tapes. If none is specified, the mount message directs the operator to mount a scratch tape. Data management gets the volume serial number from the VOL1 label and records it in the JFCB and UCB.

If you choose to specify the volume serial number, data management compares it with the volume serial number shown in the VOL1 label. If the number is correct, data management resets a mount switch in the unit control block to indicate that volume mounting is verified (the switch is initially set when the mount message is issued to the operator). If the volume serial number is incorrect, data management rejects the tape and issues another mount message.

### ***Positioning the Volume to the Data Set***

When the volume is mounted and verified, data management positions the tape to receive the new data set. Usually the new data set is the first and only data set on the tape, so the tape remains positioned immediately following the VOL1 label.

To create a data set that follows another data set already stored on the tape, you specify a data set sequence number in the LABEL parameter of the DD statement.

- The sequence number can be from 1 to 9999, with 1 representing the first data set on the volume. If you specify a sequence number that is two greater than the number of data sets existing on the volume, one of two things may happen: (1) the tape will be spaced through and removed from its reel or (2) the data set will be written but separated from the preceding data set by unusable (old) data.
- If you do not specify a sequence number, or if you specify 0, data management assumes that the data set is to be written as the first one on the volume.

To position the tape, data management maintains a logical data set sequence number in the unit control block (UCB). The method of positioning is the same as that previously explained for opening an input data set.

Only one data set on a tape volume can be open at any given time. If you attempt to open another data set on the same volume, processing is abnormally terminated. This restriction includes system output (SYSOUT) tapes.

When the tape is positioned to receive the new data set, data management expects to find either an existing HDR1 label or a tapemark. If neither one is present, data management assumes that other data is recorded where the HDR1 label should be and, therefore, processing is abnormally terminated. (If the last data set on a tape has EOV labels, another data set cannot be written to follow it.)

If a tapemark is found, it indicates that a HDR1 label does not exist at the position where the new data set is to be written. Data management bypasses all further label verification and accepts the tape for output. The conditions under which data management finds a tapemark instead of a HDR1 label are:

- When a tapemark immediately follows the VOL1 label. This may occur when the tape is initialized by means other than the IEHINITT utility program (IEHINITT writes a dummy HDR1 label following the VOL1 label. The tapemark is overwritten by the new HDR1 label.
- When, for multi-data-set organizations, the new data set is to be written after the last existing data set on the volume. In this case, data management encounters the second tapemark following the existing EOF trailer label group. The tapemark is overwritten by the new HDR1 label.

### ***Expiration Date on Existing Label***

The existing HDR1 label is inspected for the expiration date. If the expiration date has not been reached, the operator is asked to confirm use of the tape or to mount another tape.

If other data sets exist on the same volume, data management checks only the one expiration date and assumes that all following data sets expire on the same date.

### ***Protection and Data Set Name on Existing Label***

After checking the expiration date, data management inspects the accessibility indicator in the existing HDR1 label. This indicator shows whether the existing data set is protected against unauthorized use.

If protection is not indicated by an ASCII space character in the accessibility field, the tape is accepted for output. Data management neither requests a password nor checks the data set name.

If protection is indicated by an ASCII 1 or 3 in the accessibility field, data management compares the data set name shown in the existing HDR1 label to the name specified in the DD statement. If the names are not the same, processing is abnormally terminated unless the data set is the first one on the first or only volume. In this case, even if you specify a specific volume, the operator will be requested to demount the tape and mount a new scratch tape. If a security-protected data set is deleted, the data set security byte in the HDR1 label must be set to 0 before the volume can be written on again. This can be done by using either the IEHINITT utility or a user program to relabel the volume.

If the data set name is correct, data management requests the operator or TSO terminal user to key in the required password. The password is verified in a user-established password data set. This password data set contains the data set name, the password, and a protection mode indicator. The protection mode indicator is set to permit either read/write or read-only operations. The read/write mode is necessary for output data sets. Processing is terminated if:

- The operator or TSO terminal user, in two attempts, does not supply the correct password.
- The password record for the data set to be opened does not exist in the password data set.
- The read-only protection mode is specified.

*OS Data Management for System Programmers* describes the protection feature in detail and contains the information you need to create and maintain the password data set.

If the tape was created by the operating system and a character other than an ASCII space character, a 1, or a 3 is encountered, no further processing takes place.

**Note:** Verification of existing labels is considered complete after checking the HDR1 label. Any labels, data, data sets, or tapemarks following the HDR1 label are irrelevant and may be overlaid by the new output.

### ***Writing Data Set Header Labels***

When the tape is accepted by data management for output, data management creates the header labels (HDR1 and HDR2) for the new data set. These labels are created from information in the updated JFCB and other system control blocks.

The source of information for each field of each label is explained in the description of label formats. The process of updating the JFCB is explained in the introduction to this manual.

If no user header labels are to be written, data management writes a tapemark after the HDR2 label. The tape is then ready to receive the new data set.

### ***Writing ANSI User Header Labels***

When AUL is coded on the DD statement and the address of an output user header label routine is specified in the DCB exit list, data management can write user header labels (UHL).

### ***Permanent I/O Error***

If a permanent I/O error occurs during label processing, and the data set is the first one on the first or only volume, the operator will be requested to demount the tape and mount a scratch tape, even if you request a specific volume. If the data set is not the first one on the volume or this is not the first volume of a multivolume data set, the job will be abnormally terminated.

## **End of Volume on Output**

The data management EOV routine automatically switches volumes when an end-of-volume condition occurs, that is, when a reflective strip is encountered or a FEOV macro instruction is executed. This volume switching includes:

1. Writing trailer labels on the current volume
2. Checking existing labels on the new volume
3. Writing headers labels on the new volume

When multiple tape units are being used, the EOV routine also checks to see if a next-plus-one volume is needed, and if the volume just written can be rewound and unloaded. If so, the EOV routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid—the next-plus-one volume label is not verified at this time.

### ***Writing Data Set Trailer Labels***

Trailer labels are always written at an end-of-volume condition on output tapes and are identified as EOVS1 and EOVS2 (as opposed to EOF for end of data). These labels are created in the same manner and with the same content as the data set header labels, except for the label identifiers and the block count.

At end of volume, two tapemarks are written following the data set trailer labels. If user trailer labels are to be written, the tapemarks follow the user labels.

### ***Writing User Trailer Labels***

When AUL is coded on the DD statement and the address of an output user trailer label routine is specified in the DCB exit list, data management can write as many user trailer labels as desired.

### ***Labels on New Volume***

The EOVS routine handles label processing on the new volume (checking existing labels and writing new labels). The processing is the same as the Open routine's handling of the first volume.

When creating a multivolume data set, the data set sequence number is irrelevant for the second and subsequent volumes. The EOVS routine assumes that the data set continues at the beginning of the new volume.

### ***Special End-of-Volume Conditions***

When a reflective strip causes an end-of-volume condition during the writing of data, the EOVS routine writes the trailer labels as described above. If the reflective strip is encountered while writing the trailer labels, the EOVS or the Close routine continues to write the trailer labels. In both cases, the data set can be read or overwritten normally even though it crosses the reflective strip.

If you add another data set to a tape (multi-data-set organization) on which the last existing trailer label group crossed the reflective strip, or on which the new header label

group crosses the reflective strip, data management:

1. Writes the new header label group
2. Allows the user to write one record
3. Writes the new trailer label group
4. Performs volume switching

## **Closing an Output Data Set**

The Close routine handles end-of-data-set processing on output tapes. When a write operation is the last operation that occurs before closing a data set (for OUTPUT, OUTIN, or INOUT) or when no output is written before closing (for OUTPUT or OUTIN), the Close routine creates data set trailer labels.

### ***Writing Data Set Trailer Labels***

The Close routine writes the data set trailer labels with the identifiers EOF1 and EOF2. These labels are created in the same manner and with the same content as the data set header labels, except for the label identifiers and the block count.

The Close routine writes two tapemarks following the trailer labels. If user labels are to be written, the tapemarks follow the user trailer labels. If another data set is added to the tape (multi-data-set organization), its HDR1 label overlays the second tapemark.

### ***Writing User Trailer Labels***

When AUL is coded on the DD statement and the address of an output user trailer label routine is specified in the DCB exit list, the Close routine can write as many user trailer labels (UTL) as desired.

## **Restarting from a Checkpoint**

When a job step is restarted from a checkpoint, the Restart routine repositions tape volumes containing data sets that were open when the checkpoint was taken. Specifically, the Restart routine:

1. Restores applicable control blocks to the conditions that existed when the checkpoint was taken.
2. Ensures that the first existing record on the tape is a standard volume label (VOL1) and verifies the volume serial number shown in the label.
3. Uses the data set sequence number shown in the JFCB to position the tape at the interrecord gap preceding the first record of the required data set. The method of positioning is the same as previously explained for opening an input data set. The data set labels are not reprocessed.
4. Uses the block count shown in the DCB to reposition the tape at the proper record within the data set. This positioning is always performed in a forward direction. If the block count is 0 or a negative number, the tape remains positioned at the interrecord gap preceding the first record.

**Note:** ANSI labeled tape may not be assigned for the checkpoint data set.

## Format of the ANSI Standard Volume Label (VOL1)

The ANSI standard volume label (VOL1) appears at the beginning of each tape. The VOL1 label identifies the volume and its owner, and it is used to verify that the correct volume is mounted. Each tape has only one VOL1 label. When the tape is first received at an installation, the VOL1 label is created by a utility program or by a user's program.

Figure 14 shows the format of the ANSI volume label. The shaded areas represent fields that are recorded in the label, but are not used or verified during processing. The contents and processing of each field of the label are described, as are differences between the ANSI volume label and the IBM volume label.

Some systems may permit user volume labels on ASCII tapes. The operating system does not support user volume labels; if user volume labels are present on input, they are ignored. They will not appear on output tapes produced by the operating system.

### 1 — Label Identifier (3 bytes)

- *Contents:* The characters VOL identify this label as a volume label.
- *Processing:* This field is read to verify that a standard labeled tape is mounted, and that this label is a volume label.

The labels are created by the IEHINITT utility program or a user's program.

In the following situations, the Open or EOV routines create a standard volume label based on specifications in your DD statement:

When you request a standard label (AL, AUL) output volume, and an NL or NSL labeled volume is mounted to satisfy your request.

When you request a standard labeled (AL, AUL) output volume, and a volume that can be overwritten and that is recorded in the wrong density is mounted to satisfy your request.

### 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always a 1 for the ANSI standard volume label.
- *Processing:* Verified in conjunction with Field 1 to identify this label as VOL1.

### 3 — Volume Serial Number (6 bytes)

- *Contents:* A unique identification code that is assigned through IEHINITT to the volume when it enters the system or that is assigned by the operator when Open or EOV routines label the volume. This code may also appear on the external surface of the volume for visual identification. The code is normally numeric (000001-999999), but can be from 1 to 6 characters long. If the code is less than 6 characters long, it must be left-justified.

If the volume serial number is assigned in the JCL statements, all national characters, the hyphen and other special characters are accepted when enclosed in apostrophes. However, their use is not recommended because difficulties can arise in recognizing volume serial numbers when typewriter heads and print chains with nonalphanumeric characters are used.

If the volume serial number is assigned through the IEHINITT utility program, A through Z, 0 through 9, and the hyphen are the only valid characters that may be specified.

- *Processing:* The user-specified volume serial number is obtained from the JFCB and recorded in the UCB. Then the number in the UCB is compared to the number in this field of the label to ensure that the correct volume is mounted.



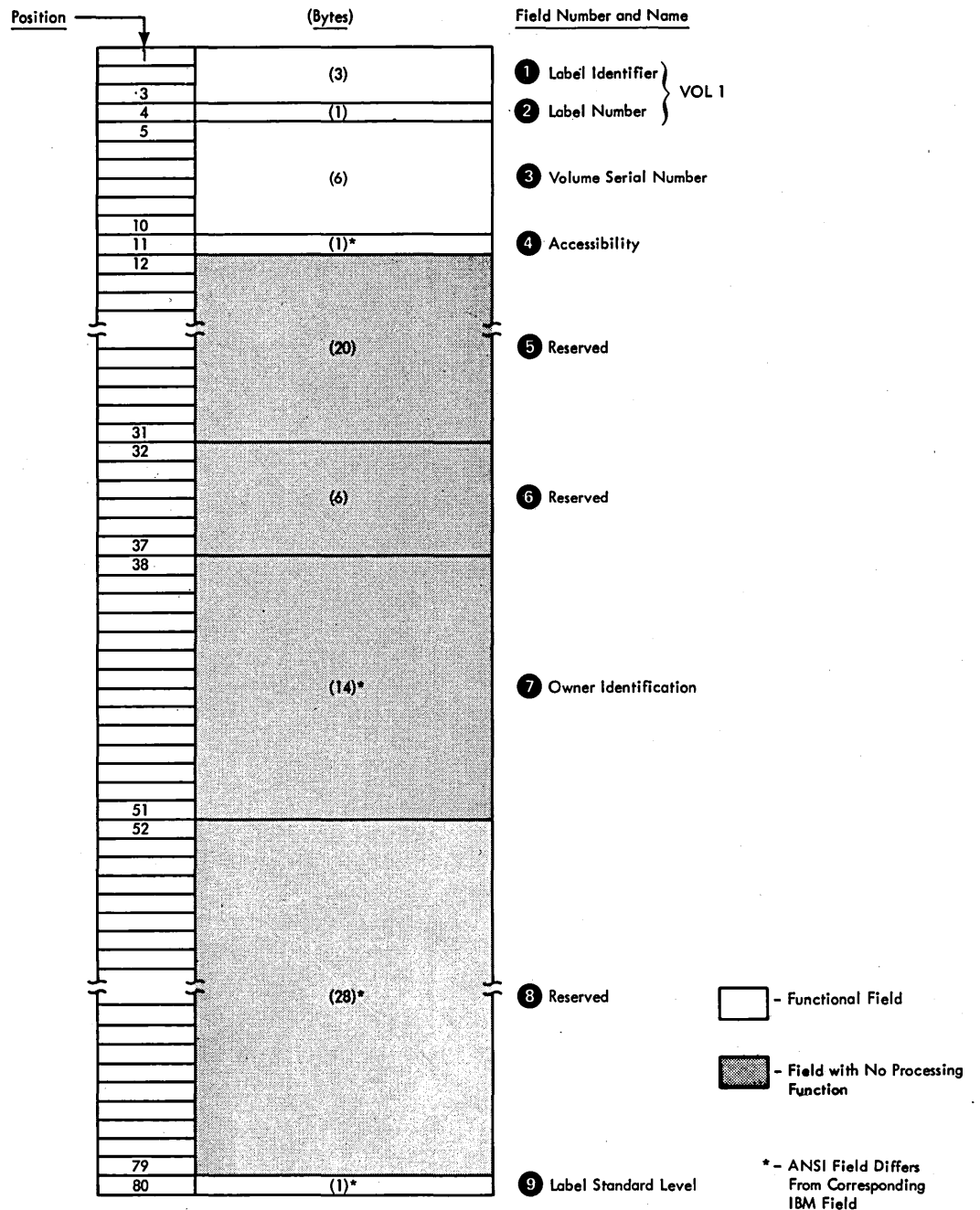


Figure 14. Format of ANSI Standard Volume Label

For scratch output tapes, the volume serial number is obtained from this field of the label and recorded in both the JFCB and the UCB.

The IEHINITT utility program can create this label with a volume serial number of up to six characters. The number is left-justified and the remainder of this field is padded with blanks. (Blanks are translated to ASCII space characters on output.)

#### 4 — Accessibility (1 byte)

- *Contents:* An ASCII space character indicates that the volume is not protected. Any character other than a space character indicates a protected volume.

- *Processing:* If any character other than an ASCII space character is present, the volume is not processed.
- *Difference from IBM Field:* The corresponding field in an IBM standard label is reserved and currently unused.

#### 5 — Reserved (20 bytes)

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field. (Blanks are translated to ASCII space characters on output.)

#### 6 — Reserved (6 bytes)

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field. (Blanks are translated to ASCII space characters on output.)

#### 7 — Owner Identification (14 bytes)

- *Contents:* Indicates a specific customer, person, installation, department, etc., to which the volume belongs. Any code or name is acceptable.
- *Processing:* Not used or verified. The IEHINITT utility program writes the text specified by the user, and the Open and EOVR routines write the text specified by the operator. If the code is less than 14 bytes long, it is left-justified and the remainder of the field is padded with blanks. (Blanks are translated to ASCII space characters on output.)
- *Differences from IBM Field:* The corresponding field on an IBM standard label is 10 bytes long.

#### 8 — Reserved (28 bytes)

- *Contents:* Reserved for future use—should be recorded as blanks.
- *Processing:* Not used or verified. The IEHINITT utility program writes blanks in this field. (Blanks are translated to ASCII space characters on output.)

#### 9 — Label Standard Level (1 byte)

- *Contents:* 1 signifies that the tape adheres to ANSI interchange standards. Any other character signifies violation of the standards.
- *Processing:* The operating system will always place a 1 in this field on output.
- *Difference from IBM Field:* This field is blank in IBM standard labels.

## Format of the ANSI Standard Data Set Label 1 (HDR1/EOV1/EOF1)

The ANSI standard data set header 1 and trailer 1 labels have the same format. These labels are used to identify and describe data sets and to protect them from unauthorized use. When AL or AUL is specified in the LABEL parameter of the DD statement, these labels are created automatically by data management routines each time a data set is recorded on tape.

Figure 15 shows the format of HDR1, EOV1, and EOF1. The shaded areas represent fields that the operating system writes in the label, but that are not used or verified during processing. The contents and processing of each field of the label are described, as are differences between the ANSI labels and the IBM labels.

Note that if generation data sets are used, not all 17 characters can be used for the file identifier, because generation version numbers are placed in the file identifier field (see

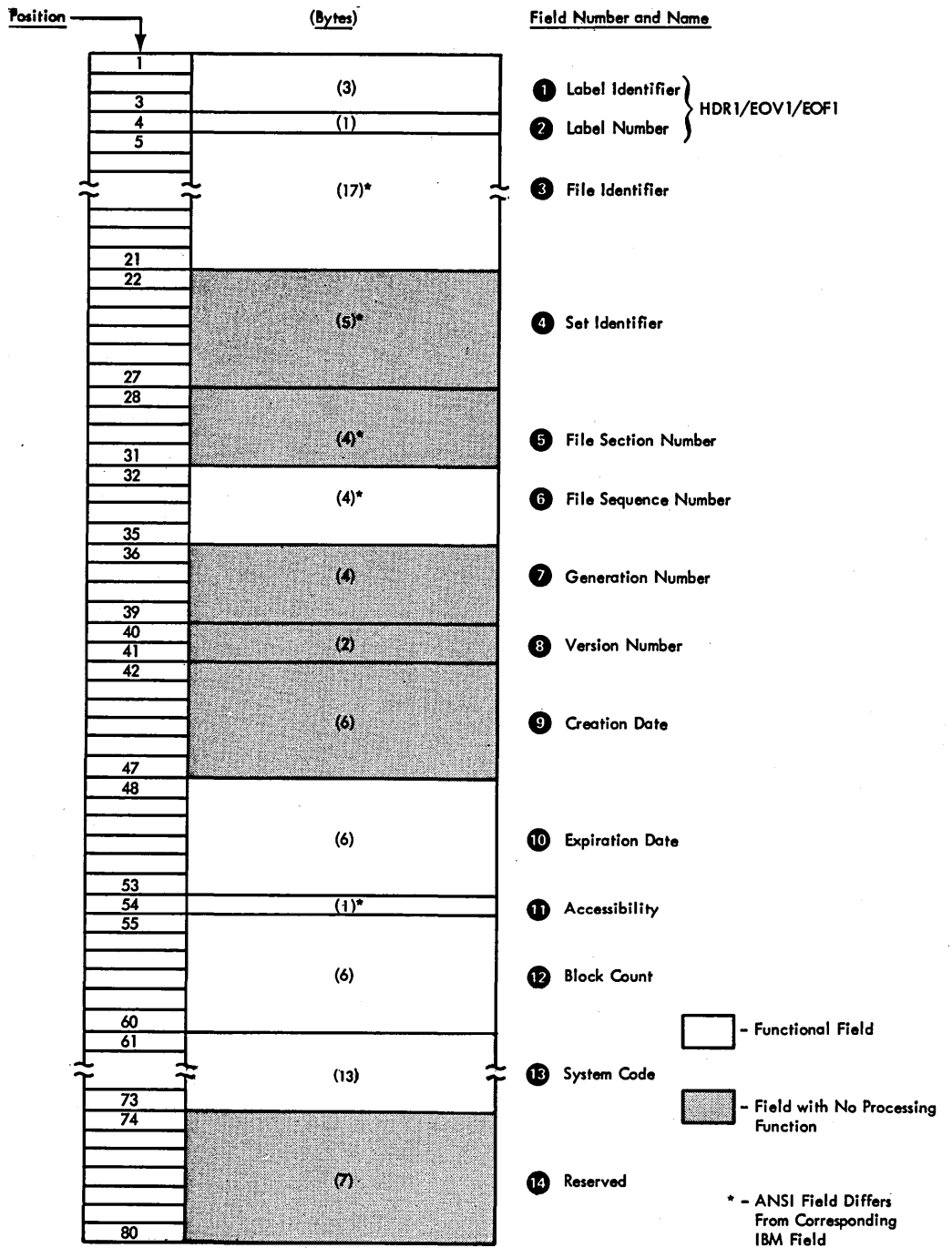


Figure 15. Format of ANSI Header 1 and Trailer 1 Labels

“File Identifier” description in this section). As a consequence, each new generation of a data set has a different file identifier. IBM does not support the optional generation and version numbers in the ANSI HDR1 label.

**1 — Label Identifier (3 bytes)**

- *Contents:* Three characters that identify the label are as follows:

HDR            Header label (at the beginning of a data set).

EOV Trailer label (at the end of a tape volume, when the data set continues on another volume).

EOF Trailer label (at the end of a data set) data set)

- *Processing:* Data management checks this field to verify that the record is an ANSI standard data set label.

For input data sets, data management checks the label identifier to determine whether data set processing is to be continued. When data management finds an EOVS label, it performs volume switching. When data management finds an EOF label, it passes control to the user's end-of-data routine.

If the DD statement specifies OPTCD=B for an input data set, the trailer label identifier (EOV or EOF) is not used to determine whether a volume switch is necessary. If more volumes are available, data management performs the switching. If no volumes are available, data management passes control to the user's end-of-data routine.

When creating trailer labels, the EOVS routine writes EOVS in this field, and the Close routine writes EOF.

## 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always a 1 for data set label 1.
- *Processing:* Verified and written in conjunction with Field 1 to identify this label as HDR1, EOVS1, or EOF1.

## 3 — File Identifier (17 bytes)

- *Contents:* The rightmost 17 bytes of the data set name (includes GnnnnVnn if the data set is part of a generation data group). If the data set name is less than 17 bytes, it is left-justified and the remainder of this field is padded with blanks. Blanks are translated to ASCII space characters on output. If the name contains embedded blanks or other special characters, you must enclose the name in apostrophes on the DD statement that requests this data set. The job control language publications list the restrictions that apply to enclosing a data set name in apostrophes. The apostrophes do not appear in the data set identifier field.
- *Processing:* For input, this name is compared to the user-specified data set name found in the JFCB. This ensures that the correct data set is being processed.

For output, the data set name in the existing label is verified in conjunction with security protection to determine whether the existing data set can be overwritten. If protection is not specified, the data set name is not checked.

When creating labels for a new data set, the user-specified data set name is obtained from the JFCB and recorded in this field.

- *Difference from IBM Field:* The corresponding field in an IBM standard label is called "Data Set Identifier."

## 4 — Set Identifier (6 bytes)

- *Contents:* The volume serial number of the tape volume containing the data set. For multivolume data sets, this field contains the serial number of the first volume of the aggregate created at the same time. The serial number can be alphanumeric, national, or special characters. If the code is less than 6 characters long, it must be left-justified. Although all national characters, the hyphen, and other special characters are accepted when enclosed in apostrophes, their use is not recommended. This is because difficulties can arise in recognizing volume serial

numbers when typewriter heads and print chains with nonalphanumeric characters are used.

- *Processing:* Not used or verified. When creating labels, the serial number is obtained from the UCB and recorded in this field.
- *Difference from IBM field:* The corresponding field on an IBM standard label is called "Data Set Serial Number."

#### **5 — File Section Number (4 bytes)**

- *Contents:* A number (0001-9999) that indicates the order of the volume within the multivolume group created at the same time. This number is always 0001 for a single volume data set.
- *Processing:* Not used or verified. When creating labels, the Open routine writes 0001 in this field; the EOVS and Close routines obtain the current volume sequence number from the DEB.
- *Difference from IBM Field:* The corresponding field on an IBM standard label is called "Volume Sequence Number."

#### **6 — File Sequence Number (4 bytes)**

- *Contents:* A number (0001-9999) that indicates the relative position of the data set within a multi-data-set group. This number is always 0001 for a single-data-set organization.
- *Processing:* This number in the first HDR1 label on the tape is referred to when the Open routine positions the tape. If this number in the first HDR1 label and the requested data set sequence number in the JFCB are both greater than 1, the logical data set sequence number in the UCB is set to the number in the label. Otherwise, the logical data set sequence number in the UCB is set to 1.

When creating labels, the Open and Close routines obtain the user-specified data set sequence number from the JFCB (a 0 is changed to 1). The EOVS routine obtains this number from the logical data set sequence number in the UCB.

#### **7 — Generation Number (4 bytes)**

- *Contents:* If the data set is part of a generation data group, this field contains a number from 0001 to 9999 indicating the absolute generation number (the first generation is recorded as 0001). If the data set is not part of a generation data group, this field contains blanks. (Blanks are translated to ASCII space characters on output.)
- *Processing:* Not used or verified. The generation number is available as part of the data set name in Field 3 of this label.

When creating labels, data management checks the JFCB to determine whether the data set is part of a generation data group. If so, the generation number is obtained from the last part of the data set name in the JFCB. Otherwise, this field is recorded as blanks.

#### **8 — Version Number of Generation (2 bytes)**

- *Contents:* If the data set is part of a generation data group, this field contains a number from 00 to 99 indicating the version number of the generation (the first version is recorded as 00). If the data set is not part of a generation data group, this field contains blanks. (Blanks are translated to ASCII space characters on output.)
- *Processing:* Not used or verified. The version number is available as part of the data set name in Field 3 of this label.

When creating labels, data management checks the JFCB to determine whether the data set is part of a generation data group. If so, the version number is obtained from the last part of the data set name in the JFCB. Otherwise, this field is recorded as blanks.

#### 9 — Creation Date (6 bytes)

- *Contents:* Year and day of the year when the data set was created. The date is shown in the format byydd where:

b = blank

yy = year (00-99)

ddd = day (001-366)

- *Processing:* Not used or verified. When data management creates labels, the date is obtained from the JFCB. This is the data when the job entered the system (set by the operator at IPL), and not necessarily the date when the label is created.

#### 10 — Expiration Date (6 bytes)

- *Contents:* Year and day of the year when the data set may be scratched or overwritten. The data is shown in the format byydd where:

b = blank

yy = year (00-99)

ddd = day (001-366)

- *Processing:* For input, not used or verified. For output, the expiration date in the existing label is compared to the current date shown in the CVT. If the date in the label is greater than the current date, a message informs the operator and gives him the option of using the tape or mounting another. If any other data sets follow on the same volume, they are considered to expire on the same day.
- When creating labels, data management obtains the expiration date from the JFCB. If you did not specify a retention period or expiration date, then the expiration date is recorded as zeros and the data set is considered expired.

#### 11 — Accessibility (1 byte)

- *Contents:* A code indicating the security status of the data set, as follows:

Space	No security protection.
1	Security protection. Additional identification of the data set is required before it can be read, written, or deleted.
3	Security protection. Additional identification of the data set is required before it can be written or deleted.
Other character	Protected volume. No access is possible under the operating system.

- *Processing:* For input, data management inspects this field on a single volume data set, on each concatenated data set, and on each volume of a multivolume data set. If protection is specified in this field, data management verifies the password furnished by the operator or TSO terminal user and sets a security indicator in the JFCB. If a character other than an ASCII space, a 1, or a 3 is encountered, the DCB will not be opened and no further processing will take place.

For output, data management inspects this field in the existing HDR1 label. If security is specified, the existing data set cannot be overwritten until data

management verifies the password and the data set name in Field 3 of this label. If you specify a data set name different from the one in Field 3, and the data set is the first one on the first or only volume, the operator is requested to demount the tape and mount a scratch tape, even though you requested a specific volume. If the data set is not the first one on the volume or this is not the first volume of a multivolume data set, the job is abnormally terminated.

When data management creates labels, the user's request for security is determined from the indicator in the JFCB.

#### **12 — Block Count (6 bytes)**

- *Contents:* This field in the trailer label shows the number of data blocks in the data set on the current volume. This field in the header label is always zero (000000).
- *Processing:* The DCB count is incremented as the data set is read. The final DCB count is compared with the count in the trailer label at end of data or end of volume. If the counts do not agree, a user exit entry in the DCB exit list determines whether processing will continue or abnormally terminate. If the appropriate user exit entry is not provided, a block count discrepancy causes processing to abnormally terminate.

For read backward, the verification process is reversed. The trailer label count is recorded in the DCB and decremented as the data set is read. The final DCB count should be 0, which is equal to the count in the header label.

When data management creates labels, the block count in the header label is set to zeros. The block count in the trailer label is obtained from the DCB.

#### **13 — System Code (13 bytes)**

- *Contents:* A unique code that identifies the system, for example, "OS360" followed by 8 zeros.
- *Processing:* On input, the field is checked to determine how succeeding labels are to be processed. On output, the operating system supplies the "OS360" code.

#### **14 — Reserved (7 bytes)**

- *Contents:* Reserved for future use—contains blanks.
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field. (Blanks are translated to ASCII space characters on output.)

## Format of the ANSI Standard Data Set Label 2 (HDR2/EOV2/EOF2)

For ASCII interchange tapes, this set of labels (HDR2, EOV2, and EOF2) is optional. This practice differs from IBM labeling conventions, which require header 2 and trailer 2 labels. The header 2 and trailer 2 labels have the same format. They are used to show information about the structure of records on the file. Because these labels are optional, and because their formats may vary depending on the system that produces them (non-IBM systems may use different formats), the processing of header 2 and trailer 2 labels will vary. If the labels are produced by the operating system, they are treated like IBM header 2 and trailer 2 labels. If the labels are produced by another system, only the fields that conform to ANSI interchange standards will be processed, as explained later in this section.

Figure 16 shows the format of HDR2, EOV2, and EOF2. The shaded areas represent fields that the operating system writes in the label, but that are not used or verified during processing. The contents and processing of each field of the label are described, as are differences between ANSI labels and IBM labels.

### 1 — Label Identifier (3 bytes)

- *Contents:* Three characters that identify the label are as follows:

HDR            Header label (at the beginning of a data set)

EOV            Trailer label (at the end of a tape volume, when the data set continues on another volume)

EOF            Trailer label (at the end of a data set)

- *Processing:* Data management checks this field to verify that the record is an ANSI standard data set label.

For input data sets, data management checks the label identifier to determine whether data set processing is to be continued. When data management finds an EOV label, it performs volume switching. When data management finds an EOF label, it passes control to the user's end-of-data routine.

If the DD statement specifies OPTCD=B for an input data set, data management accepts either EOV or EOF as the trailer label identifier, and the identifier is not used to determine whether a volume switch is necessary. If more volumes are available, data management performs the switching. available, data management passes control to the user's end-of-data routine.

When creating trailer labels, the EOV routine writes EOV in this field, and the Close routine writes EOF.

### 2 — Label Number (1 byte)

- *Contents:* The relative position of this label within a set of labels of the same type; it is always 2 for data set label 2.
- *Processing:* Verified and written in conjunction with Field 1 to identify this label as HDR2, EOV2 or EOF2.

### 3 — Record Format (1 byte)

- *Contents:* An alphabetic character that indicates the format of the records in the associated data set:

F Fixed length

D Variable length with a zoned decimal length indicator

U Undefined length



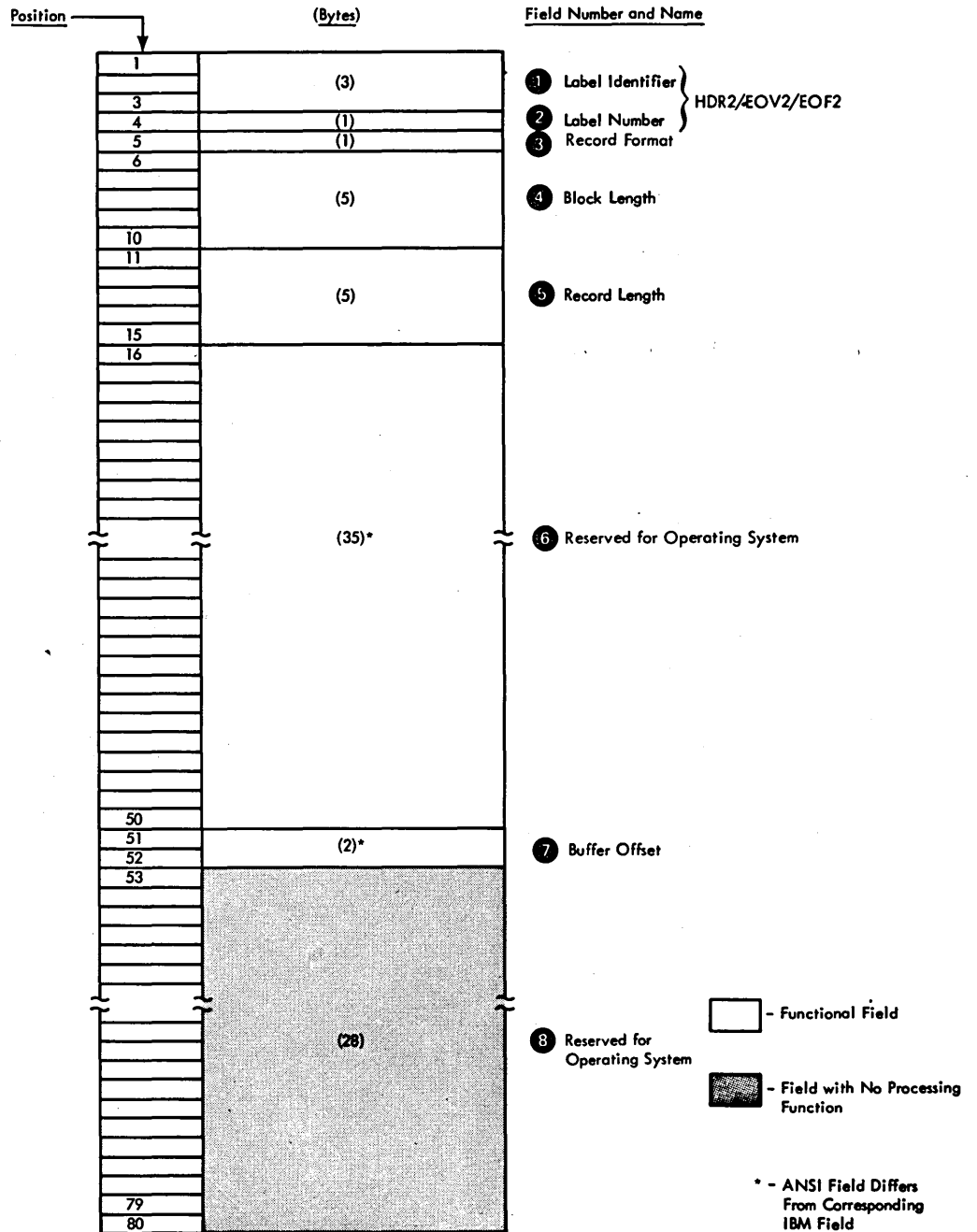


Figure 16. Format of ANSI Header 2 and Trailer 2 Labels

- Processing:** For input, the record format is obtained from this label and recorded in the JFCB (if the JFCB field is 0). Then the record format in the JFCB is recorded in the DCB (if the DCB field is 0). Note that this is a merging process in which existing specifications in the JFCB and DCB cannot be overridden.

When creating labels, a reverse merge follows the forward merge described above. The record format in the DCB overrides the record format in the JFCB, and the updated JFCB provides the information for the label.

This merging process is explained and illustrated in the introduction to this manual.

#### 4 — Block Length (5 bytes)

- *Contents:* A number from 18 to 2048 that indicates the block length (including buffer offset and padding) in bytes. *Note:* The 18-byte to 2048-byte limit on block length is an ASCII standard. Larger blocks may be specified with the agreement of the interchange parties. Interpretation of the number depends on the associated record format in Field 3, as follows:

Format F Block length

Format D Maximum block length (including the 4-byte length indicator in the records and the optional block prefix)

Format U Maximum block length

- *Processing:* The number in the label is converted to binary and merged with appropriate fields in the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

#### 5 — Record Length (5 bytes)

- *Contents:* A number that indicates the record length in bytes. Interpretation of the number depends on the associated record format in Field 3, as follows:

Format F Logical record length

Format D Maximum logical record length (including the 4-byte length indicator in the records and the optional block prefix)

Format U Zeros

- *Processing:* The number in the label is converted to binary and merged with the appropriate fields in the JFCB and DCB. The merging process is the same as for the record format code in Field 3 of this label.

#### 6 — Reserved (35 bytes)

- *Contents:* If the label is produced by the operating system, the content and format of this field are identical to the content and format of the seven fields following the record length field in an IBM standard label. If the label is produced by another system, the content and format of this field are optional, and the field is not processed by the operating system. The fields produced by the operating system are:

##### A. Tape Density (1 byte)

- *Contents:* A DEN value of 2 indicates a 9-track tape recorded in 800 CPI (characters per inch), NRZI (non-return-to-zero-inverted). If the tape is recorded in ASCII code but is not an interchange tape, other values (as described in the section on IBM standard labels) may be placed in this field.
- *Processing:* Not used or verified. When data management creates labels, the information for this field is obtained from the JFCB.

##### B. Data Set Position (1 byte)

- *Contents:* A code indicating a volume switch is as follows:
  - 0 No volume switch has occurred.
  - 1 A volume switch previously occurred.
- *Processing:* Not used or verified. When creating labels, the Open routine writes 0 in this field, and the EOVS routine writes a 1. The Close routine determines which code to write by comparing the volume serial number in the JFCB to the

number in the UCB. It writes 0 if the numbers are equal, and 1 if they are not equal.

**C. Job/Job Step Identification (17 bytes)**

- *Contents:* Identification of the job and job step that created the data set. The first 8 bytes contain the name of the job; the ninth byte is a slash (/), and the last 8 bytes contain the name of the job step.
- *Processing:* Not used or verified. When data management creates labels, the names of the job and job step are obtained from the TIOT.

**D. Tape Recording Technique (2 bytes)**

- *Contents:* Recorded as blanks for 9-track tape; 9-track tape can only be recorded in odd parity with no translation.
- *Processing:* Not used or verified.

**E. Control Characters (1 byte)**

- *Contents:* A code indicating whether a control character set was used to create the data set, and the type of control characters used:
  - A Contains ASCII control characters
  - b Contains no control characters
- *Processing:* The specification in the label is converted to a bit code and merged with the appropriate fields of the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

**F. Buffer Alignment Block (1 byte)**

- *Contents:* Reserved for future use—recorded as blanks.
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field.

**G. Block Attribute (1 byte)**

- *Contents:* A code indicating the block attribute used to create the data set:
  - B Blocked records
  - b Records not blocked
- *Processing:* The specification in the label is converted to a bit code and merged with the appropriate fields of the JFCB and DCB. The merging process is the same as for the record format code in Field 3 of this label.

**7 — Buffer Offset (2 bytes)**

- *Contents:* The length of the block prefix.
- *Processing:* Used to determine buffer-offset length.
- *Difference from IBM Field:* This field is not present in IBM standard labels.

**8 — Reserved (28 bytes)**

- *Contents:* Reserved for future use—recorded as blanks. (Blanks are translated to ASCII space characters on output.)
- *Processing:* Not used or verified. When creating labels, data management writes blanks in this field.

## Format of ANSI User Header and Trailer Labels (UHL/UTL)

ANSI user labels contain user-specified information about the associated data sets. There is no limit to the number of user labels that may be associated with a data set. User labels are optional.

User header and trailer labels have the same format. If you want the data management routines to write user header labels or to make user header labels available to your program, you must specify AUL on the DD statement and specify the address of a user header or trailer label routine in the DCB exit list. (The exit list can address several user routines simultaneously, for example, a routine that processes input user header labels and a routine that creates output user header labels.) The exit list (EXLST) is described in *OS Data Management Services Guide*.

Figure 17 shows the UHL and UTL labels.

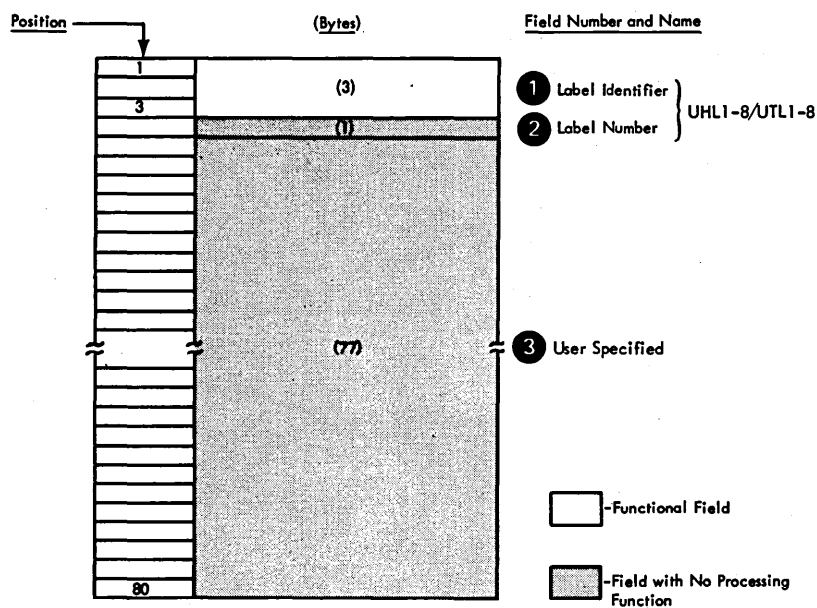


Figure 17. Format of ANSI User Labels

### 1 — Label Identifier (3 bytes)

- **Contents:** Three characters that identify the label are as follows:  
 UHL User header label (at the beginning of a data set)  
 UTL User trailer label (at the end of volume or end of data set)
- **Processing:** This field is read to verify that the record is a user label Data management accepts either UHL or UTL.

### 2 — Label Number (1 byte)

- **Contents:** Any valid ASCII character except the ASCII substitute character.
- **Processing:** This field is not checked by the operating system.
- **Difference from IBM Field:** This field can contain only numeric 1-8 for IBM standard user labels. A maximum of 8 user header or trailer labels is supported for

conventional IBM standard user labels.

### 3 — User Specified (76 bytes)

- *Contents:* Specified by the user.
- *Processing:* Specified in the DCB exit list.

## Other File Header Labels

Other file header labels (HDR3-HDR9, EOF3-EOF9, EOVS3-EOVS9) will not be created by the operating system, nor will user volume labels (UVL). Because these labels may appear on magnetic tapes created by other systems, the operating system will accept them as input. These labels are ignored during label processing and are not placed on output tapes. Figure 18 shows a hypothetical input tape from a non-IBM system and a corresponding output tape produced by the operating system. The user volume label, the additional header label, and the additional trailer label are not placed on the output tape.

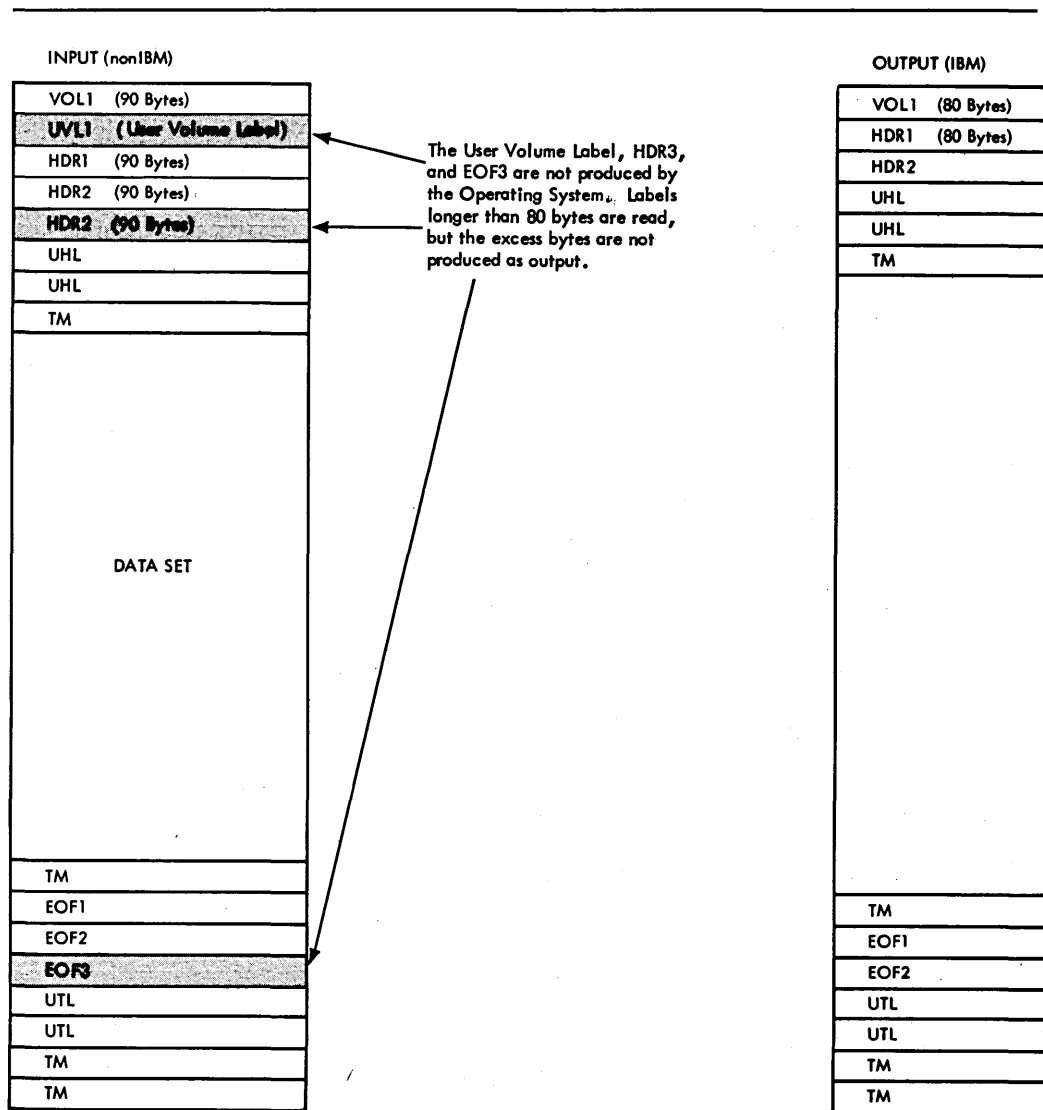


Figure 18. Hypothetical ASCII Tape From a Non-IBM System



## NONSTANDARD LABELS

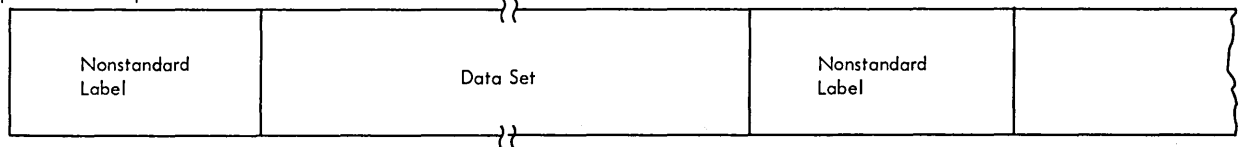
Nonstandard labels do not conform to the IBM or ANSI standard label formats. They are designed by your installation and are written and processed by routines provided by your installation. There are no requirements as to the length, format, contents, and number of nonstandard labels, except that the first record on a BCD, EBCDIC, or ASCII tape cannot be a standard volume label.

Figure 19 shows some examples of how you can organize tape volumes with nonstandard labels. Other variations are possible. Since your routines do the positioning, there are no special requirements for multivolume or multi-data-set organizations. All labels and tapemarks are written by your routines. If an operating system access method is used to retrieve the data, tapemarks should precede and follow the data set to indicate the end-of-data-set condition for forward and backward read operations.

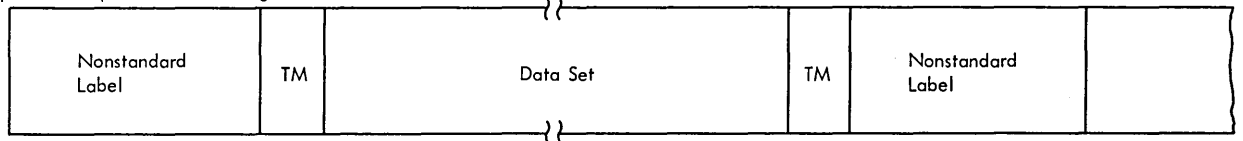
If you wish to use nonstandard labels for tape volumes at your installation, you must:

1. Create nonstandard label processing routines for input header labels, input trailer labels, output header labels, and output trailer labels.
2. Insert your routines into the operating system as part of the SVC library (SYS1.SVCLIB).

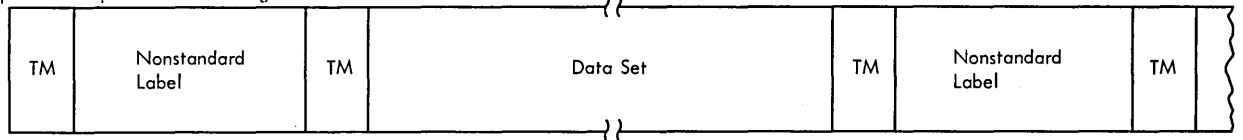
Example 1 -- No Tapemarks



Example 2 -- Tapemarks Delimiting the Data Set



Example 3 -- Tapemarks Delimiting the Labels and the Data Set



1. **No Tapemarks:** This type of organization can be created by your nonstandard label processing routines, and read with the EXCP technique. It should not be used with an operating system access method because there is no tapemark to signal end-of-data.
2. **Tapemarks Delimiting the Data Set:** This is the recommended organization. The tapemarks are written by your nonstandard label processing routines. When the tape is read by an operating system access method, the tapemark following the data set signals end-of-data for forward read operations, and the tapemark preceding the data set signals end-of-data for backward read operations.
3. **Tapemarks Delimiting the Labels and the Data Set:** This is an expansion of the preceding organization. The additional tapemarks that precede and follow the labels are not handled by the operating system. They are written and used by your nonstandard label processing routines.

Figure 19. Examples of Tape Organizations with Nonstandard Labels

3. Code NSL in the LABEL parameter of the DD statement at execution time.

This section of the manual explains how your nonstandard label processing routines work with the operating system control program, how to write your routines, and how to insert your routines into the operating system.

## Processing Tapes with Nonstandard Labels

Your appropriate nonstandard label processing routine is selected, brought into main storage, and executed when a data set is opened or closed, when an end-of-volume or end-of-data-set condition occurs, or for repositioning a volume when a job step is restarted from a checkpoint. When your routine has completed its processing, it must return control to data management's Open, Close, EOV, or Restart routine which then continues its normal processing. For input, the EOV routine handles both end-of-volume and end-of-data-set conditions. For output, the EOV routine handles the end-of-volume condition, and the Close routine handles the end-of-data-set condition.

Your routines must provide for reading labels, processing labels, writing labels, writing tapemarks, identifying volumes, and positioning volumes. The control program assumes that a tape with nonstandard labels is properly positioned upon completion of a nonstandard label processing routine.

If you want the control program to maintain a block count, your header label routines that receive control from Open or EOV must properly initialize the block count field of the DCB.

When processing is completed, the control program handles volume disposition in accordance with the parameters of the DD statement. Your nonstandard label processing routines are responsible for any positioning specified by the OPEN or CLOSE macro instructions. If you need to process a data set more than once in a job, or if you want to handle multi-data-set volumes, your routines must control the positioning. If you handle volume disposition in your nonstandard label processing routines, you must issue volume disposition messages to the console operator. Data management checks to see if your routine has handled disposition, and it bypasses disposition and message handling if volume disposition is verified.

The following paragraphs explain the flow of control between the control program and each type of nonstandard label processing routine. Information on tape positioning and volume identification is also provided.

### *Input Header Label Routines*

When nonstandard labels are specified, the control program checks the input tape to make sure that the first record is not an IBM or ANSI standard volume label. If the first record contains the identifier VOL1 in the first four bytes, it is recorded in EBCDIC and is 80 bytes long, or is recorded in ASCII and is 80 bytes or more in length. The tape is rejected by a message from the control program directing the operator to demount the current volume and mount the correct volume. The various error conditions that can occur during verification of the first record are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

When it is determined that the tape does not contain a standard volume label, the Open or EOV routine gives control to your routine for processing input header labels. Control comes from the Open routine for the first or only volume of a data set, or for a concatenated data set with unlike characteristics. (Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program. Any exception in processing makes the data sets unlike.) Control comes from the EOV routine for the second and subsequent volumes of a data



set, or for a concatenated data set with like characteristics. When your routine receives control, the tape has been rewound; the tape is positioned at the interrecord gap preceding the nonstandard label. Note: If the control program finds that the tape volume has been previously verified in the job, the tape has not been rewound.

If your routine determines that the wrong volume is mounted, you must place a 1 in the high-order bit position of the SRTEDMCT field of the unit control block (UCB), and return control to the control program. The control program then issues a message directing the operator to mount the correct volume. When the new volume is mounted, the control program again checks the initial label on the tape before giving control to your routine.

Before returning control to the control program, your input header label routine must position the tape at the appropriate data set:

- For forward read operations, position the tape at the interrecord gap that precedes the initial record of the data set.
- For backward read operations, position the tape after the last record of the data set.

### ***Input Trailer Label Routines***

When a tapemark is encountered on an input tape, data management's EOV routine gives control to your routine for processing input trailer labels, with two exceptions. The EOV routine does not give control to your input trailer label routine when:

- The FEOV macro instruction is used to force an end-of-volume condition.
- At the end of the data set, the data control block (DCB) exit list indicates deferred nonstandard input trailer label processing.

When your routine receives control, the tape is already positioned for label processing:

- For forward read operations, the tape has been positioned immediately after the tapemark at the end of the data set.
- For backward read operations, the tape has been positioned immediately before the tapemark at the beginning of the data set.

Your routine need not reposition the tape before returning control to the control program.

If additional volumes are specified in the job file control block (JFCB), the control program uses the next specified volume serial number and performs volume switching. (You specify the volume serial numbers in forward sequence, regardless of whether the tapes are to be read forward or backward.) If the new volume is not already mounted, the control program issues a mount message to the operator. The new volume is then processed by the EOV routine and your input header label processing routine.

If another volume is not specified in the JFCB, the control program gives control to your end-of-data-set routine that is specified in the data control block (DCB). Subsequently, the processing program or the operating system closes the data set. When an input data set is closed, your output trailer label routine is given control. This allows you to position the tape if necessary. When an end-of-data-set condition occurs and the data control block (DCB) exit list indicates deferred nonstandard input trailer label processing, the Close routine passes control to your input trailer label routine before passing control to your output trailer label routine. The DCB exit list is described in *OS Data Management Services Guide*, see "Exit List (EXLST)."

## ***Output Header Label Routines***

When nonstandard labels are specified for output, the control program checks the tape to make sure that the existing first record is not a standard volume label. If the first record is 80 bytes in length and contains the identifier VOL1 in the first four bytes, the tape is not accepted, as is. If an IBM standard label exists, it is overwritten with an IBM tapemark, if possible. If an ANSI standard label exists, the console operator must confirm that it can be destroyed. The various error conditions that can occur during verification of the first record are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

When the control program ensures that the first record on the tape is not a standard volume label, the Open or EOJ routine gives control to your routine for processing output header labels. Control comes from the Open routine for the first or only volume of a data set. Control comes from the EOJ routine for the second and subsequent volumes of a data set. When your routine receives control, the tape has been positioned at the interrecord gap preceding the nonstandard label (the tape has been rewound). However, the tape has not been rewound if the control program found that this volume has been previously verified during the job.

If your routine determines that the wrong volume is mounted, you must place a 1 in the high-order bit position of the SRTEDMCT field of the unit control block (UCB) and return control to the control program. The control program then issues a message directing the operator to mount the correct volume. When the new volume is mounted, the control program again checks the initial label on the tape before giving control to your routine.

Your routine need not reposition the tape before returning control to the control program.

When tapes are first received at your installation, they should be initialized with a tapemark or other record. If a blank tape is mounted for an output data set, it is spaced through and removed from its reel when the control program looks for an existing standard volume label.

## ***Restart Label Processing Routine***

If you restart at checkpoints and use tapes with nonstandard labels, you must provide a routine to process nonstandard labels at restart time. You need only a routine to check existing header labels. You do not need separate routines for input and output, because output tapes will contain the header labels that were written when the data sets were opened (before checkpoint).

At restart time, the control program checks the tape to make sure that the first record is not a standard volume label. If the first record is 80 bytes long and contains the identifier VOL1 in the first four bytes, the tape is rejected by a message from the control program directing the operator to mount the correct tape.

When it is determined that the tape does not contain a standard volume label, the control program's Restart routine gives control to your routine for processing nonstandard labels. When your routine receives control, the tape has been positioned at the interrecord gap preceding the nonstandard label (the tape has been rewound).

If your routine determines that the wrong volume is mounted, you must place a 1 in the high-order bit position of the SRTEDMCT field of the unit control block (UCB), and return control to the control program. The control program then issues a message directing the operator to mount the correct volume. When the new volume is mounted, the control program again checks the initial label on the tape before giving control to your routine.

Before returning control to the control program, your routine must position the tape at the interrecord gap that precedes the initial record of the appropriate data set. This applies to both forward and backward read operations. The control program then uses the block count shown in the DCB to reposition the tape at the appropriate record within the data set. This positioning is always performed in a forward direction. If the block count is zero or a negative number, the control program does no positioning. (If you want the control program to reposition the tape, your normal header label routines—Open and EOVS—must properly initialize the block count field of the DCB. The block count field of the DCB must not be altered at restart time.)

### ***Output Trailer Label Routines***

Your routine for writing output trailer labels receives control from data management's EOVS or Close routines. The EOVS routine handles end-of-volume conditions (reflective strip or FEOVS macro instruction). The Close routine handles end-of-data-set conditions (CLOSE macro instruction). When your routine receives control, the tape has been positioned at the interrecord gap following the last data set record that was written.

Your routine need not reposition the tape before returning control to the control program.

Your output trailer label routine is also given control when input data sets are closed. This allows you to position the tapes if necessary.

## **Writing Nonstandard Label Processing Routines**

The following paragraphs describe conventions, requirements, and techniques for writing your nonstandard label processing routines.

### ***Programming Conventions***

The programming conventions to be observed when writing your routines are:

- *Size of the routine:* Nonstandard label processing routines are not limited in size. However, if the routine exceeds 1024 bytes, you must divide it into load modules of 1024 bytes or less. To pass control between load modules, you must use the XCTL macro instruction (E-form).
- *Design of the routine:* Nonstandard label processing routines must be reenterable. You cannot use relocatable address constants in the routines or in any channel command words to be used by the routines. You cannot use any macro instructions whose expansions contain relocatable address constants.
- *Register usage:* When your routine receives control, it must save the contents of registers 2 through 14 (in your own work area). Before returning control, your routine must restore the contents of these registers.
- *Entry point of the routine:* The entry point of the routine must be the first byte of the load module and must be on a doubleword boundary.
- *Exit from the routine:* You must use the XCTL macro instruction (E-form) to exit from your routine and return control to a specific control program module. These modules differ depending on the control program routine from which control was received and the type of label processing being performed. Module names are shown below for each control program routine and for each type of label processing routine.

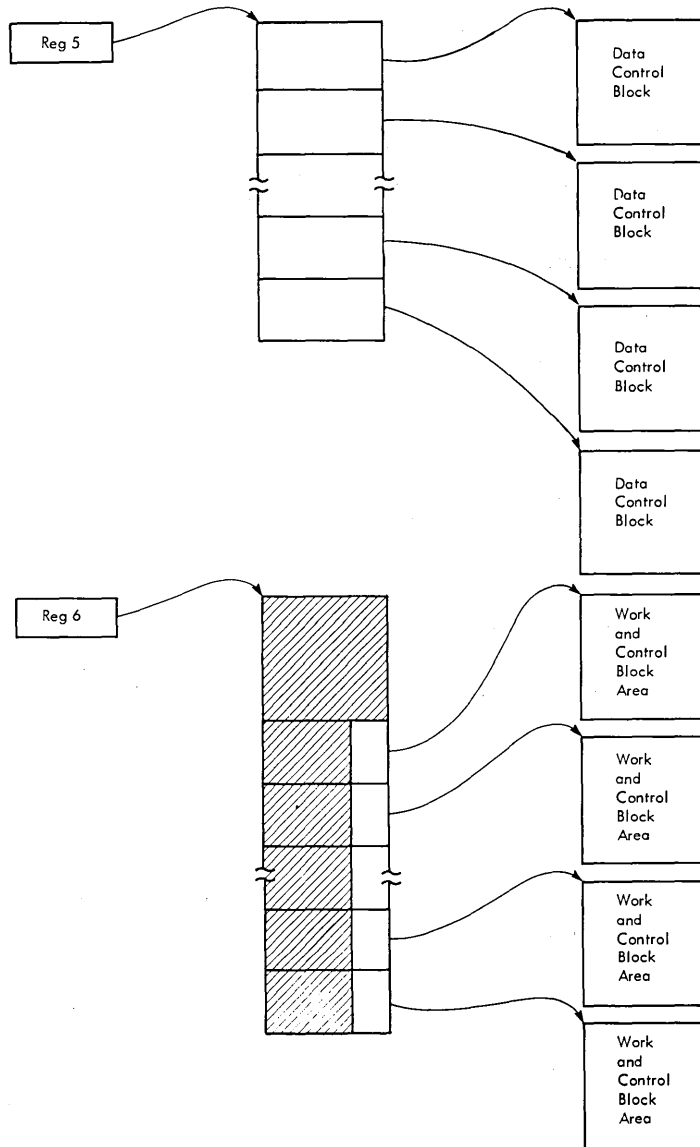
Label Processing Routine	Control Program Routine	Control Program Module Name
Input Header	Open	IGG0190B
	EOV	IGG0550D
Input Trailer	EOV	IGG0550B
Output Header	Open	IGG0190R
	EOV	IGG0550H
Output Trailer	EOV	IGG0550F
	Close	IGG0200B
Restart Header	Restart	IGC0K05B

- *Work areas:* You must use the GETMAIN macro instruction to obtain main storage for all of your work areas, including areas used to read in or create a label. You must use the FREEMAIN macro instruction to release this main storage.

## ***Program Functions***

In processing nonstandard labels, you must perform many of the functions that the control program performs in processing standard labels. All input/output operations, such as reading labels, writing labels, and positioning volumes, must be performed by using the EXCP (execute channel program) macro instruction. The use of EXCP normally requires that you build several control blocks in your work area. However, you can save coding effort and main storage space by using control blocks already established by the control program.

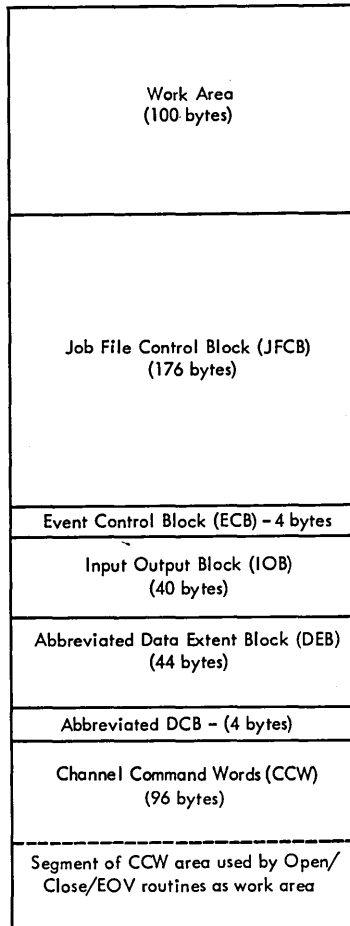
- When your routine receives control from the Open or Close routine, the status of control information and pointers is as shown in Figure 20.
- When your routine receives control from the EOV routine, register 2 contains the address of a DCB, and register 4 contains the address of a combined work and control block area. The format of this area is shown in Figure 21.
- When your routine receives control from the Restart routine, register 9 contains the address of a restart table entry. The table entry contains the address of a control block area. This status is as shown in Figure 22.



Register 5 contains the starting address of a list of DCB addresses. Each DCB specified in the OPEN or CLOSE macro instruction has a 4-byte entry in the list. The DCBs to which the entries point are in the problem program.

For each DCB specified in the OPEN or CLOSE macro instruction, a combined work and control block area is built. Register 6 contains the starting address of a table that contains an address for each work and control block area. The addresses of the areas are contained in the low-order 3 bytes of 8-byte entries. The list of 8-byte entries begins 32 bytes from the starting address of the table. The format of the combined work and control block area is shown in Figure 21.

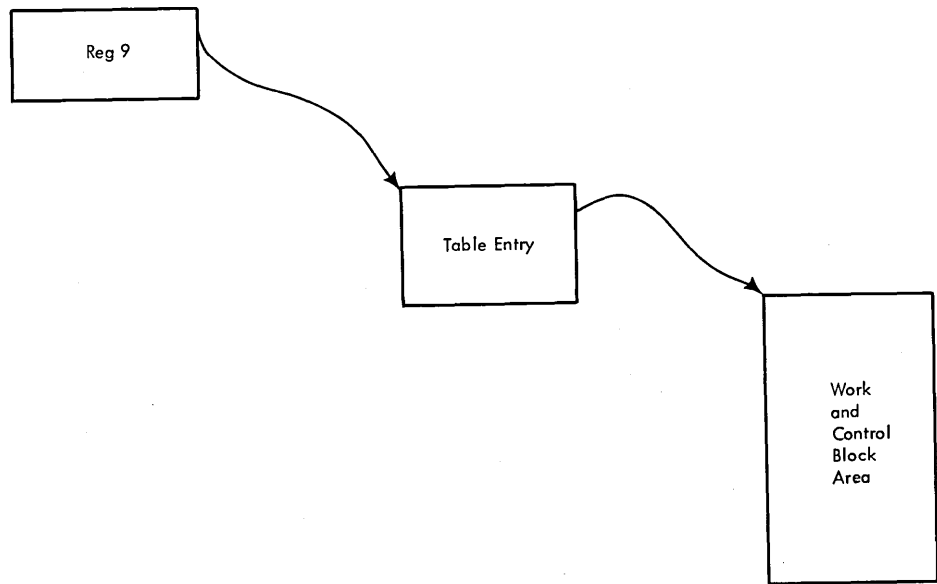
**Figure 20. Status of Control Information and Pointers**



Each of the fields within the work and control block area can be addressed by your nonstandard label processing routines. The IECDSECT macro instruction defines the symbolic names of all these fields. Code this macro instruction (with a null operand field and immediately preceded by a DSECT statement) in the list of constants for each of your nonstandard label processing routines. Using the starting address of the work area as a base, you are able to address any field symbolically. For more information on this macros see "OS Data Management for System Programmers."

When your nonstandard label processing routine receives control from the Close or EOV routine, some of the information shown in the work area DEB is not the same as contained in the actual DEB. If you need actual DEB information at these times, you can get the address of the DEB from the DCBDEBAD field in the DCB.

**Figure 21. Format of Combined Work and Control Block Area**



Register 9 contains the starting address of a 48-byte table entry. Five bytes from the starting address of this table entry, a 3-byte field contains the starting address of a work and control block area that is associated with the data set. Complete descriptions of the table entry and the work and control block area are provided in Appendix C.

**Figure 22. Status of Control Information and Pointers from the Control Program's Restart Routine**

General flowcharts of nonstandard label processing routines are shown in Figures 23, 24, 25, and 26. These flowcharts suggest the logic that you could use in your routines. The logic is shown separately for routines receiving control from the Open, Close, EOVS, or Restart routines of the control program. Each block in the flowcharts is numbered, and the number corresponds to an item in the list of explanations that follows.

#### Explanation of Logic Blocks—Figures 23, 24, and 25

- 1 — The entry is in the form of an XCTL macro instruction issued by the control program.
- 2 — Use the GETMAIN macro instruction to obtain main storage.
- 3 — Use the store multiple instruction (STM).
- 4 — To locate the address of the data control block (DCB), use the contents of register 5. To determine if the DCB is to be processed, test bits 6 and 7 of the DCBOFLGS field of the DCB; if these bits are 1, the DCB is to be processed. (The symbolic names of all fields in the DCB are defined by the DCBD macro instruction.)
- 5 — To determine if a tape data set is being processed, test the UCBTBYT3 (bit 0) field of the unit control block (UCB); this bit is 1 for a tape data set. The symbolic names of all fields in the UCB are defined by the IEFUCBOB macro instruction. The address of the UCB is contained in the DXDEBUCB field of the data extent block (DEB) as defined by the IECDSECT macro instruction. A method for finding the address of the UCB is described in the *OS MVT Guide*, or the *OS MFT Guide*. For more information on the IECDSECT and IEFUCBOB macros see *OS Data Management for System Programmers*.

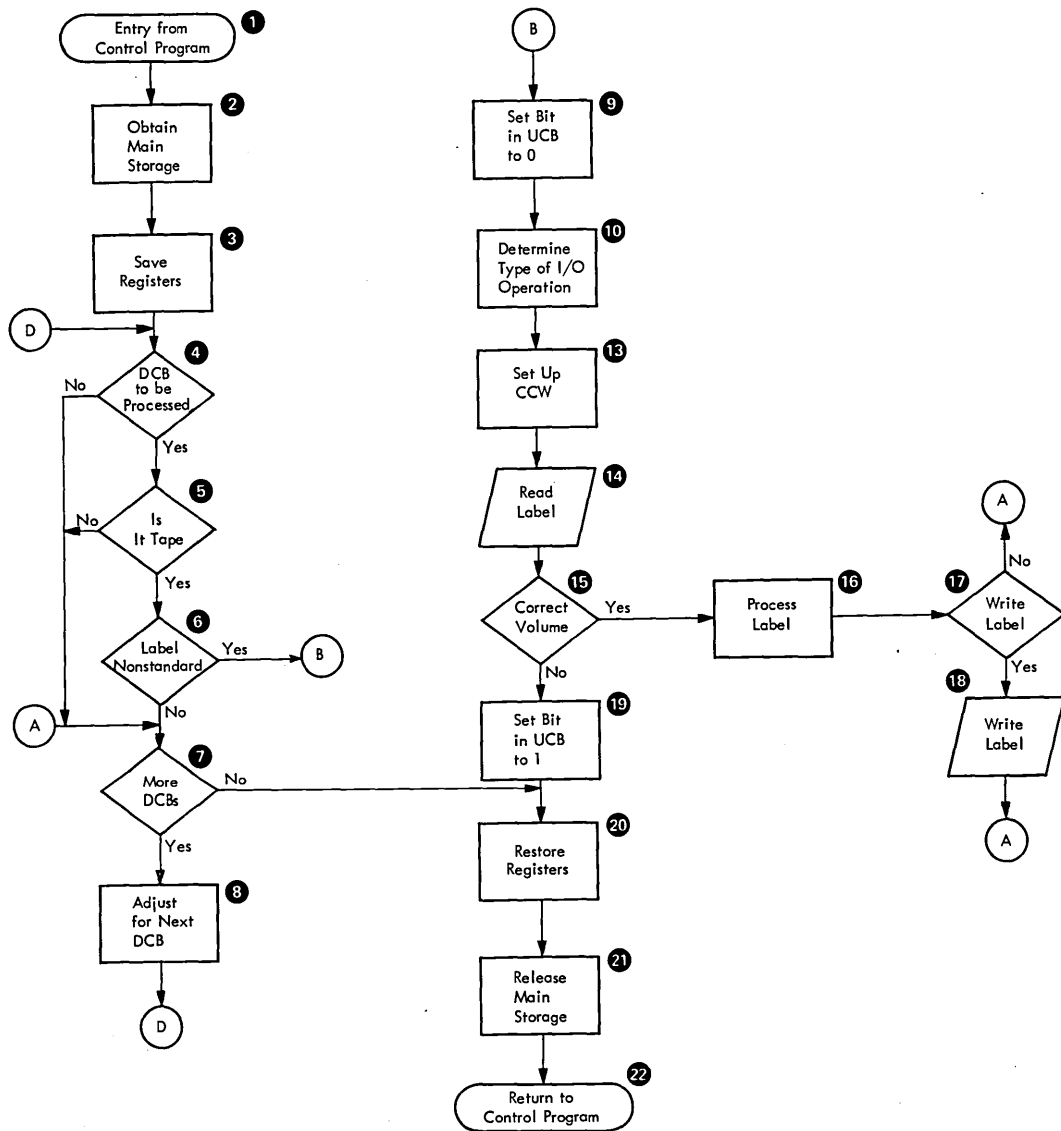


Figure 23. General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Open Routine

- 6 — To determine if nonstandard labels have been specified, test the JFCBLTYP field of the job file control block (JFCB); this field contains a hexadecimal 04 when nonstandard labels have been specified.
- 7 — The final DCB entry in the list of DCB addresses contains a 1 in its high-order bit position.
- 8 — Add 4 to the contents of register 5; add 8 to the contents of register 6.
- 9 — Set the high-order bit to 0 in the SRTEDMCT field of the UCB.
- 10 — To determine the type of I/O operation specified in the OPEN macro instruction,



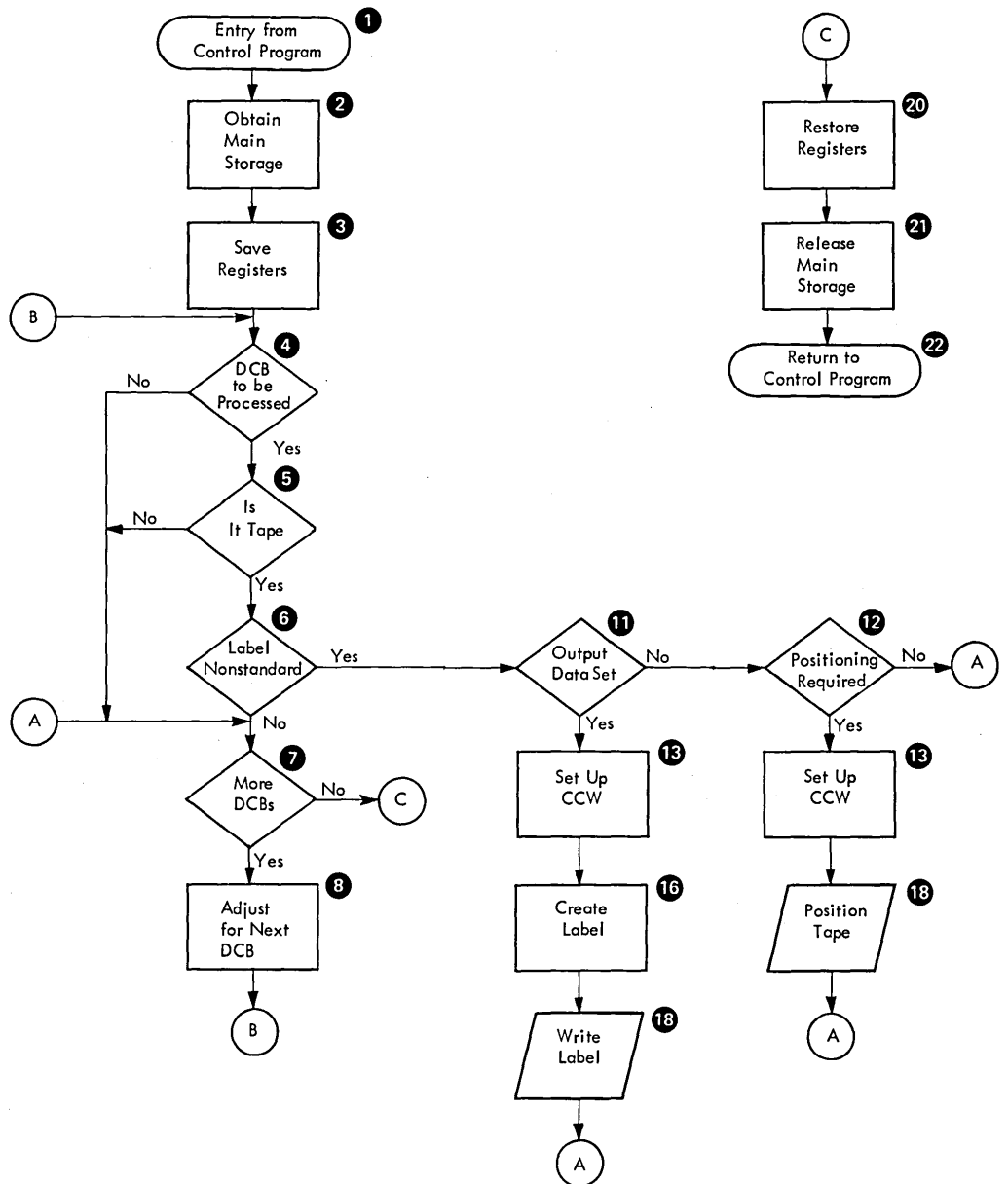


Figure 24. General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Close Routine

check the bit configuration of the high-order byte of the DCB entry in the list of DCB addresses. The bit configuration for each type of I/O operation is shown below. (The high-order four bits correspond to the disposition of the data set; the low-order four

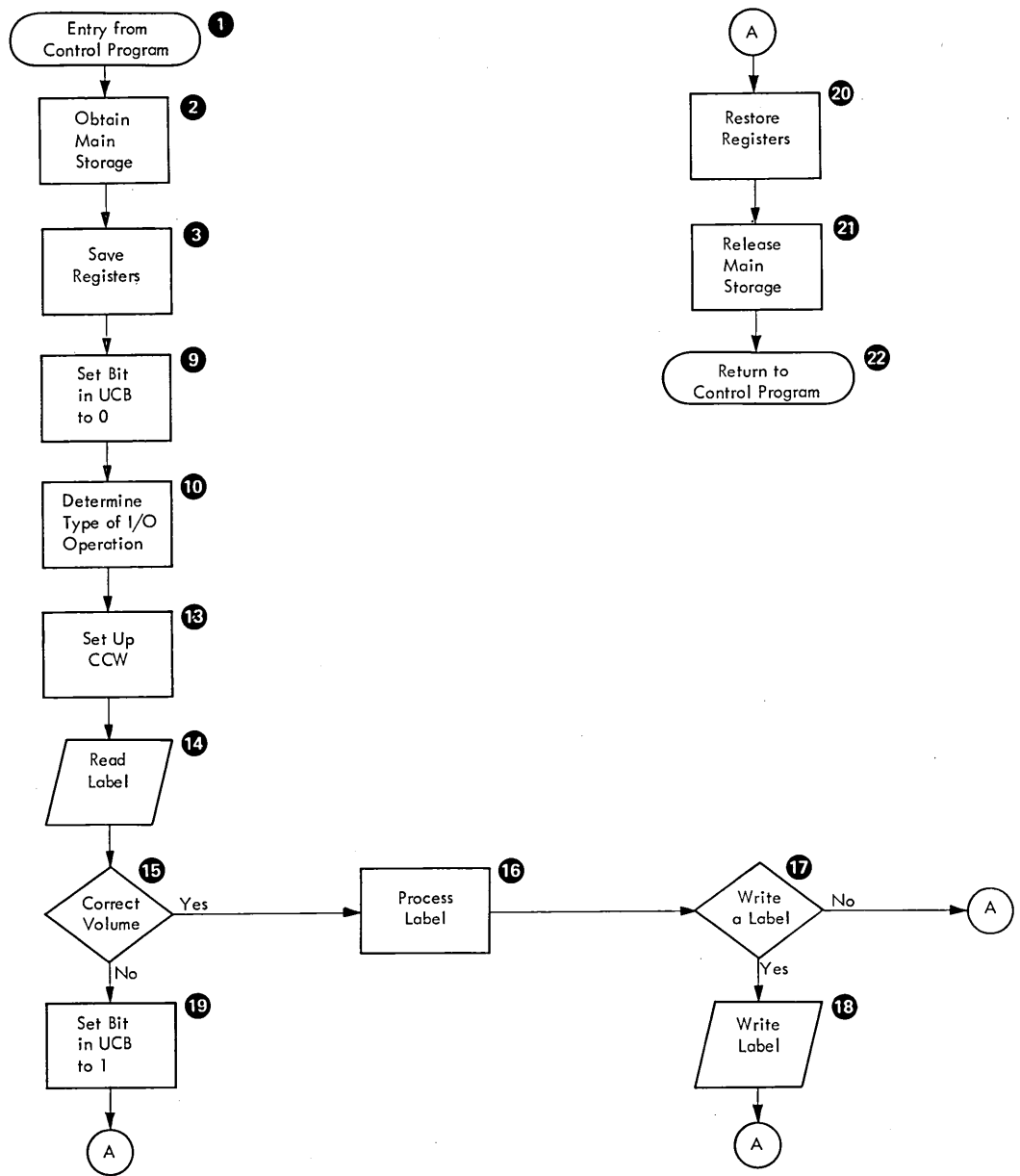


Figure 25. General Flow of a Nonstandard Label Processing Routine after Receiving Control from the EOVS Routine

bits correspond to the I/O operation itself. For example, the bit configuration x0110000 indicates a data set opened for input whose disposition is LEAVE.)

Bits	0	1	2	3	4	5	6	7	
	x	0	0	1	x	x	x	x	REREAD
	x	0	1	1	x	x	x	x	LEAVE
	x	0	0	0	x	x	x	x	Neither REREAD nor LEAVE
	x	x	x	x	0	0	0	0	INPUT
	x	x	x	x	1	1	1	1	OUTPUT
	x	x	x	x	0	0	1	1	INOUT
	x	x	x	x	0	1	1	1	OUTIN
	x	x	x	x	0	0	0	1	RDBACK
	x	x	x	x	0	1	0	0	UPDAT

**11** — To determine the mode of the data set, test the high-order bit of the DCBOFLGS field of the DCB. If this bit is 1, the data-set mode is output; if this bit is 0, the data-set mode is input. (The symbolic names of all fields in the DCB are defined by the DCBD macro instruction.)

**12** — You may want to position the tape if you have closed an input data set before all data has been read.

**13** — Move your CCW into the channel program area of the control program's work area. (The symbolic name of the first entry in the channel program area is DXCCW.) You can use the first six entries.

**14** — Issue an EXCP macro instruction specifying the address of the control program's IOB. (The symbolic name of the IOB is DXIOB.)

**15** — Techniques used to check for correct volume differ depending on the label formats used in the installation.

**16** — Label processing routines differ by label format.

**17** — If a write operation is required, this block can be used.

**18** — Issue an EXCP macro instruction specifying the address of the control program's IOB. (The symbolic name of the IOB is DXIOB.)

If the command is a rewind, set the rewind-issued bit in the UCB (UCBWGT field, bit 3) before issuing the EXCP.

If the command is a rewind-unload, set the unit-not-ready bit in the UCB (UCBFL1 field, bit 1) and zero out the UCB volume serial number field (SRTEVOLI) after the channel program is complete.

**19** — Set the high-order bit to 1 in the SRTEDMCT field of the UCB. of the UCB.

**20** — Use the load multiple (LM) instruction.

**21** — Use the FREEMAIN macro instruction to free the work area obtained in step 2.

**22** — Use the XCTL macro instruction, specifying the appropriate operand.

The following coding sequence illustrates an exit from your routine during Open or Close operations. Register 4 contains the address of the control program's Open/Close work area.

	LR	1,SAVEBASE	put work area pointer into reg 1 for FREEMAIN
	LM	2,14,REGSAVE	restore caller's registers
	FREEMAIN	R,LV=size,A=(1)	
	BALR	15,0	use 15 for temp base
	USING	*,15	
	MVC	0(8,6),MODNAME	module name to OPEN/CLOSE a
	LA	15,456(4)	use area in OPEN workarea
	XCTL	EPLOC=(6),SF=(E,(15))	
MODNAME	DC	C'IGGxxxxx'	

The following coding sequence illustrates an exit from your routine during end-of-volume operations. Register 4 contains the address of the control program's EOv work area.

	LR	1,SAVEBASE	put work area pointer into reg 1 for FREEMAIN
	LM	2,14,REGSAVE	restore caller's registers
	FREEMAIN	R,LV=size,A=(1)	
	BALR	15,0	use 15 for temp base
	USING	*,15	
	MVC	472(8,4),MODNAME	name of EOv area
	LA	15,464(4)	use area in EOv workarea
	LA	0,472(4)	address of module name
	XCTL	EPLOC=(0),SF=(E,(15))	
MODNAME	DC	C'IGGxxxxx'	

#### Explanation of Logic Blocks—Figure 26

- 1 — The entry is in the form of an XCTL macro instruction issued by the control program.
  - 2 — Use the GETMAIN macro instruction to obtain main storage.
  - 3 — Use the store multiple (STM) instruction.
  - 4 — Move your CCW into the channel program area of the control program's work area. (The symbolic name of the first entry in the channel program area is RSCCW1.)
- The device modifier byte RSDEBMOD, in the DEB portion of the restart work area, is provided by the control program and will contain the mode-set command for the data portion of the tape. If the non-standard labels at your installation are recorded in a mode different than the data, your NSL routine must set the device modifier byte (RSDEBMOD) to the density and recording technique of the labels. (See the discussion "Tape Characteristics" in "Introduction to Tape Processing.")
- 5 — Issue an EXCP macro instruction specifying the address of the control program's IOB. (The symbolic name of the IOB is RSIOB.)
  - 6 — Determine if an uncorrectable I/O error occurred. This can be any type of error that you do not want to accept.
  - 7 — Set the high-order bit to 1 in the TABTPLBL field of the table entry.
  - 8 — Techniques used to check for correct volume differ depending on the label formats used in the installation. The volume serial number for the mounted volume is shown in the UCB.
  - 9 — Perform any necessary label processing and tape positioning.
  - 10 — Set the high-order bit to 1 in the SRTEDMCT field of the UCB.
  - 11 — Use the load multiple (LM) instruction.

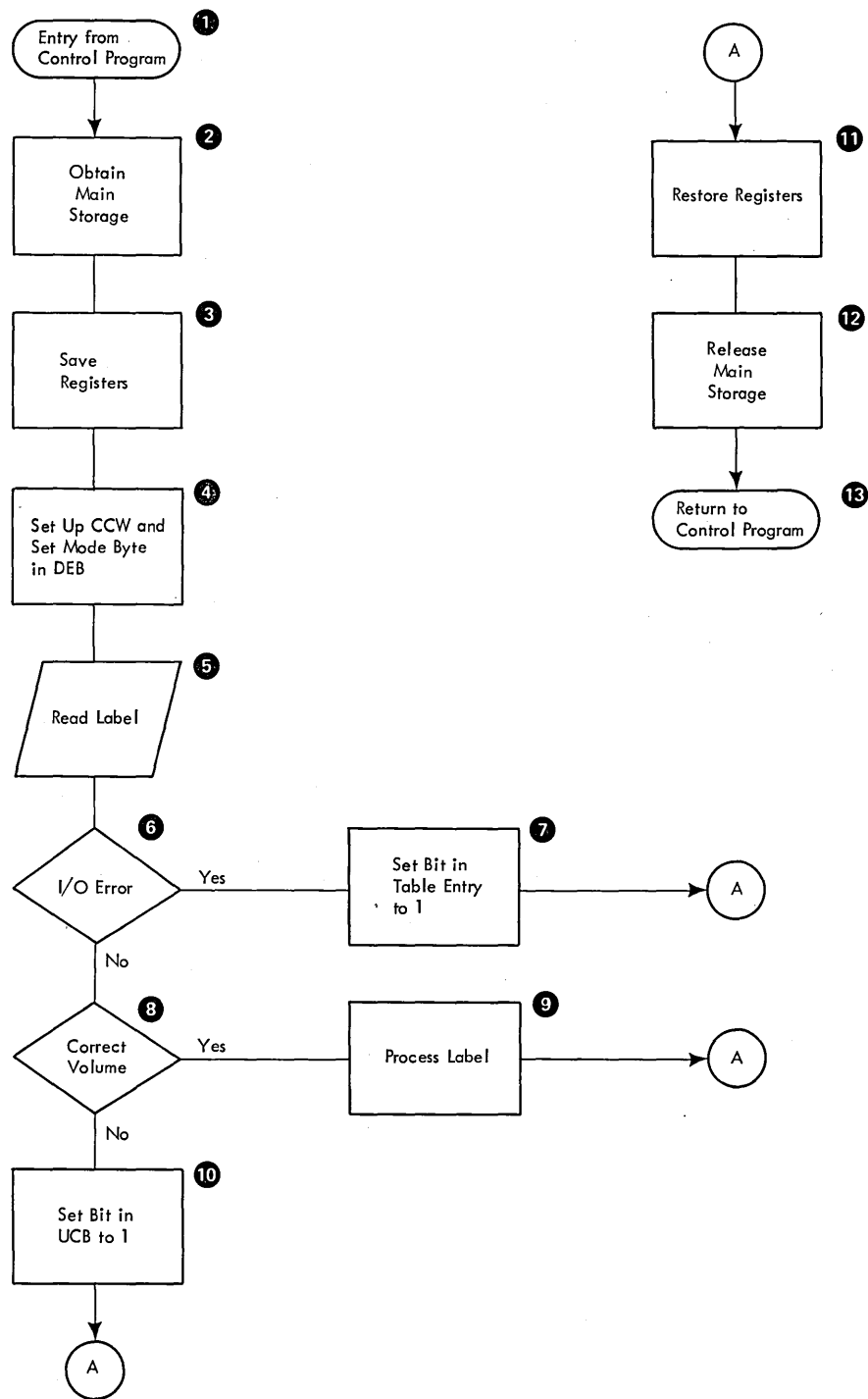


Figure 26. General Flow of a Nonstandard Label Processing Routine after Receiving Control from the Restart Routine

- 12 — Use the FREEMAIN macro instruction to free the work area obtained in step 2.
- 13 — Use the XCTL macro instruction. The following coding sequence illustrates an exit from your routine during restart operations:

LR	1,SAVEBASE	put work area pointer into reg 1 for FREEMAIN
LM	2,14,REGSAVE	restore caller's regs
FREEMAIN	R,LV=size,A=(1)	
BALR	15,0	use 15 for temp base
USING	*,15	
L	1,4(,9)	put pointer to restart work area into reg 1 (see Figure 12)
MVC	128(8,1),MODNAME	put module name into restart work area
LA	15,120(,1)	set up XCTL parameter pointers
LA	0,128(,1)	set up XCTL parameter pointers
XCTL	EPLOC=(0),SF=(E,(15))	
MODNAME	DC	C'IGCOK05B'

## Inserting Nonstandard Label Routines Into the Control Program

Nonstandard label processing routines must be included in the control program as part of the SVC library (SYS1.SVCLIB), since they are load modules of type 4 SVC routines. This is done during the system generation process. (The routines may also be inserted after system generation by link-editing them into the SVCLIB. This procedure is similar to replacing volume label editor routines, which is described in another section of this manual.)

Before your routines can be inserted into the control program, each load module must be a member of a cataloged, partitioned data set. You must name this data set with the SYS1 prefix (for example, SYS1.name).

To insert your load modules into the SVC library during system generation, you use the SVCLIB macro instruction. With this macro instruction, you must specify the name of the partitioned data set and the names of members to be included in the SVC library. Member names for the first load module of each type of label processing routine are listed below. Member names for additional load modules must begin with the letters NSL or IGC. The format and specifications of the SVCLIB macro instruction are contained in *OS System Generation*.

Nonstandard Label Processing Routine	Control Program Routine	Member Name
Input Header	Open	NSLOHDRI
	EOV	NSLEHDRI
Output Header	Open	NSLOHDRO
	EOV	NSLEHDRO
Input Trailer	EOV	NSLETRLI
Output Trailer	EOV	NSLETRLO
	Close	NSLCTRLO
Restart Reader	Restart	NSLRHDRI

## Automatic Volume Recognition (AVR) Nonstandard Label Processing Routine

To enable the AVR option to process nonstandard magnetic tape labels, you must write a routine to supply AVR with information concerning the nonstandard labels. This routine is inserted in the control program in place of an IBM-supplied routine that causes AVR to reject tape volumes that do not have standard labels. The information returned to AVR by your routine consists of a validity indication (for example, the label read is valid) and the location within the nonstandard label of the volume serial number field.

Specifically, your routine must:

1. Determine if the label under consideration is a valid, nonstandard label as defined by your installation.
2. Set general register 15 to zero if a valid label is detected, or to nonzero if the label is not recognizable. (A nonzero return causes AVR to unload the tape volume and issue an error message.)
3. When a valid label is detected, place the location of the volume serial number field within the label in an area provided by AVR. (The label or the first part of it, is read into an 80-byte work area by AVR before your routine receives control; the location is defined within this work area. Also before your routine receives control, AVR positions the tape at the interrecord gap after the nonstandard label.)
4. Return control to AVR. Register 14 contains the return address. (The SAVE and RETURN macro instructions may be used in your routine.)

Your label processing routine receives control when the AVR routine cannot identify the first record on a magnetic tape volume as a standard label. The various error conditions that can occur during verification of the first record are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

### *Entry Conditions*

When your routine receives control, the AVR routine has placed the nonstandard label in an 80-byte work area, and general register 1 contains the address of a 2-word area whose contents are as follows:

- |        |   |
|--------|---|
| Word 1 | The address of the beginning byte of the 80-byte work area  |
| Word 2 | The address of a 1-word area where your routine stores the beginning address of the volume serial number field within the nonstandard label |

### *Conventions*

The format of your installation's nonstandard label(s) must provide for a 6-byte volume serial number field within the first 80 bytes of the label. Otherwise the volume serial number will not be read into the 80-byte internal work area. This does not restrict the overall nonstandard label format from being more, or less, than 80 bytes in length.

The name of your routine must be IEFXVNSL.

## Inserting AVR Nonstandard Label Routines into the Control Program

You may replace the IBM-supplied routine IEFXVNSL with your routine by link-editing your assembled routine into the SYS1.CI505 data set prior to system generation, or you may replace the IBM-supplied routine after system generation by link-editing your

assembled routine into the control program module. The module and the object deck step are as follows:

**MFT—30K Scheduler: Module IEFXVAVR**

**object deck**

```
INCLUDE      SYSMOD( IEFXVAVR )
ALIAS        IEFXV001
ENTRY        IEFXV001
NAME         IEFXVAVR( R )
```

**MFT—44K Scheduler: Module IEFXVAVR**

**object deck**

```
INCLUDE      SYSMOD( IEFXVAVR )
ALIAS        IEFXV001
ENTRY        IEFXV001
NAME         IEFXVAVR( R )
```

**MVT: Module IEFXVAVR**

**object deck**

```
INCLUDE      SYSMOD( IEFXVAVR )
ALIAS        IEFXV001
ENTRY        IEFXV001
NAME         IEFXVAVR( R )
```

The following sequence of job control and linkage editor statements shows insertion of your AVR nonstandard label routine into the 18K scheduler of the MVT configuration.

```
//jobname      JOB      [parameters]
//stepname     EXEC     PGM=IEWL[,parameters]
//SYSPRINT     DD       SYSOUT=A
//SYSUT1       DD       UNIT=SYSDA,SPACE=(parameters)
//SYSMOD       DD       DSNAME=SYS1.LINKLIB,DISP=OLD
//SYSLIN       DD       *
```

**object deck**

```
INCLUDE      SYSMOD( IEFXVAVR )
ALIAS        IEFXV001
ENTRY        IEFXV001
NAME         IEFXVAVR( R )
/*
```



## Volume Verification and Dynamic Device Reconfiguration

If you use nonstandard tape labels and you wish to use the dynamic device reconfiguration (DDR) option, you must perform your own volume verification. Note that you must be able to perform your verification within the first 48 bytes of any record in your nonstandard label.

Before system generation time, code a routine named NSLREPOS and link-edit it into a cataloged data set. The either specify that data set, with NSLREPOS as a member name, in the SVCLIB system generation macro instruction, or link-edit NSLREPOS directly into the SYS1.SVCLIB after system generation.

At system generation time you must also specify DDRNSL as a task supervisor option of the SUPRVSOR macro instruction. Coding the SVCLIB and the SUPRVSOR macro instructions is discussed in *OS System Generation*.

When your NSLREPOS routine receives control from the DDR tape reposition routine, register 2 contains a pointer to an XCTL list (built by DDR) in IORMSCOM. This list contains the module name to which you transfer control when you return control to DDR. Register 5 points to a buffer (SVRBEXSA) containing the first 48 bytes of your label record. The serial number of the volume against which verification is made is in the STREVOL1 field of the UCB. Register 7 contains the UCB address.

Before returning control, your routine should put one of the following codes into register 0:

Code	Explanation
0	Volume verification is complete. Since a tapemark follows this label, the tape reposition routine must forward space to that tapemark and clear the block count it has accumulated before it begins repositioning.
4	The NSLREPOS routine needs more information for volume verification. When the tape reposition routine receives this code, it reads the first 48 characters of the next record into the buffer and returns control to NSLREPOS.
8	The wrong volume has been mounted. When the tape reposition routine receives this code, it sends a message to the operator explaining that the wrong volume has been mounted.
C	Volume verification is complete. Since no tapemark follows this label, the tape reposition routine repositions the volume, using the block count it has accumulated.
10	Volume verification is complete. Since the tapemark following the label has already been reached, the tape reposition routine clears the block count it has accumulated and repositions the volume.

If NSLREPOS uses any registers other than registers 0, 14 or 15, the routine must save the registers (using GETMAIN from subpool 255) and store them in its own area before returning control to the tape reposition routine. When your NSLREPOS routine returns control to DDR, the following sequence should be used:

Load register 0 with the return code

```
LR    15,2
XCTL SF=(E,(15))
```

Note that programs coded for earlier releases will still be operational.



## UNLABELED TAPES

To process or create a tape with no labels, you specify NL in the LABEL parameter of the DD statement. An unlabeled tape contains only data records and tapemarks. The organization of data sets on one or more volumes is shown in Figure 27. The data management routines of the operating system automatically write the tapemarks on output and expect to find a similar placement of tapemarks on input.

- A tapemark does not precede the first data set on any volume.
- A tapemark follows every data set.
- Two tapemarks follow a data set if it is the last or only data set on the volume.

An unlabeled tape can be read backward even though there is no tapemark preceding the first data set. In this case, the end-of-data-set condition is signaled by the reflective strip at the beginning of the tape.

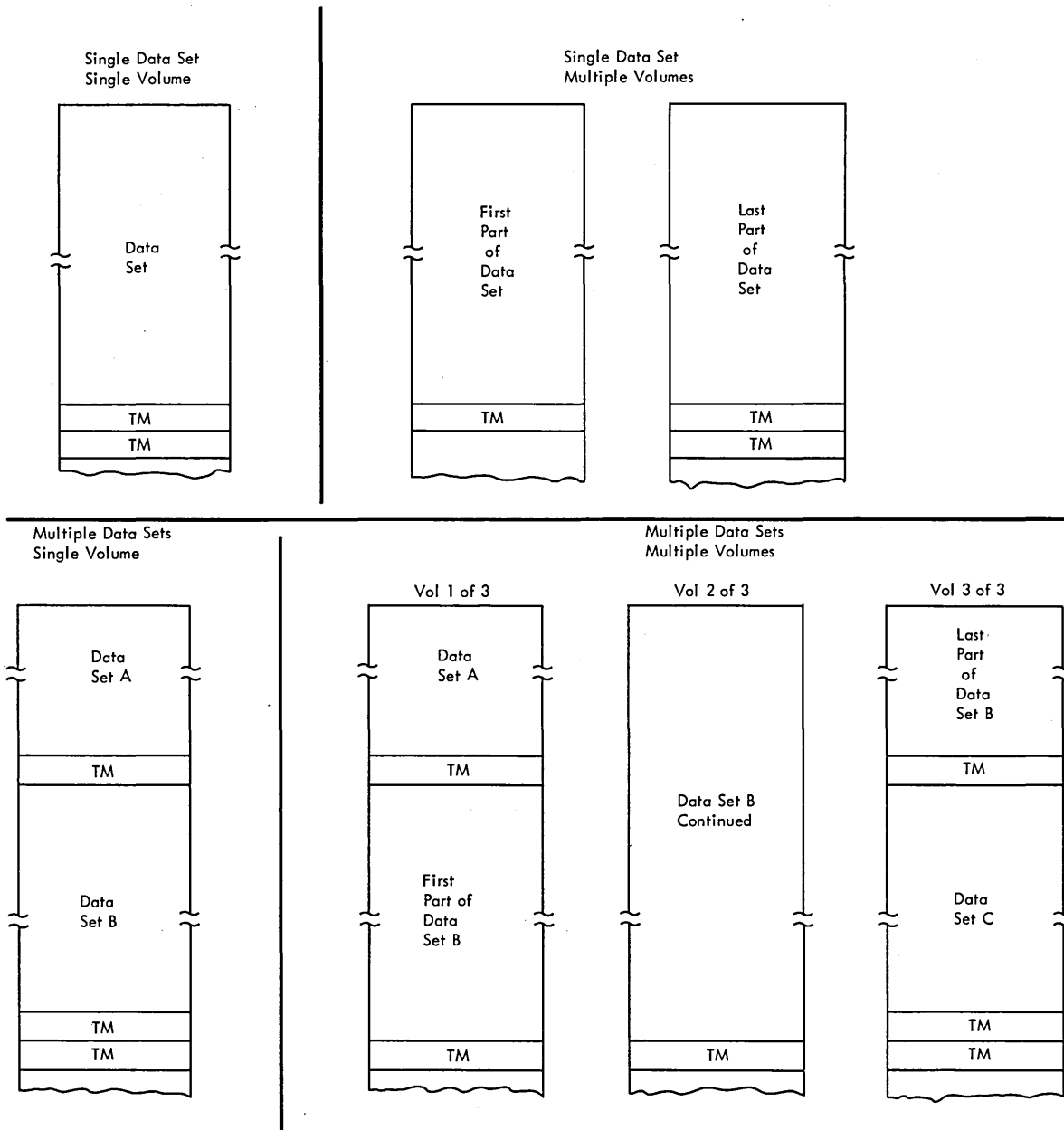


Figure 27. Organizations for Unlabeled Tapes

## Opening an Input Data Set

When you specify no labels, data management checks the input tape to ensure that the first record is not a standard volume label. If the first record is an IBM standard volume label or an ANSI standard volume label, the tape is rejected by a message from the control program directing the operator to mount the correct tape. The various error conditions that can occur during label verification are explained in "Volume Label Verification and Volume Label Editor Routines" in this publication.

The search for a standard label is the only mount verification performed by the operating system. Without labels, neither the volume nor the data set can be positively identified and data management assumes that they are correct. The operator is responsible for checking the reel's external identification to ensure compliance with the mount message.

### *Positioning the Volume to the Data Set*

When the tape is accepted for input, data management positions the tape at the first record of the data set to be processed. Usually there is only one data set on the volume and positioning is set to the first record on the tape.

To retrieve a data set when there are more than one on a single reel of tape, you specify a data set sequence number in the LABEL parameter of the DD statement, unless the data set is cataloged. You need not specify a data set sequence number for a cataloged data set, because the number can be obtained from the catalog along with the volume serial number:

- The sequence number can be from 1 to 9999 with 1 representing the first data set on the volume. If you specify a sequence number higher than the number of data sets on the volume, the tape will be spaced through and removed from its reel.
- If you do not specify a sequence number, or specify zero, and the data set is not cataloged, data management assumes that the data set is first in sequence on the volume.
- The first data set on an unlabeled tape is not preceded by a tapemark. If a tapemark precedes the first data set, the sequence number of that data set is two (the effect is as if a data set containing no data preceded the tapemark).

To position the tape, data management uses the requested data set sequence number shown in the JFCB and maintains a logical data set sequence number in the unit control block (UCB). The number in the UCB represents the current position of the tape and is maintained as follows:

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened, the Open routine sets the data set sequence number in the UCB to 1. (If the tape is still positioned from previous processing, such as for a LEAVE request, the Open routine does not reset the number in the UCB.)
3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the Open routine adjusts the data set sequence number in the UCB as the tape is spaced past each data set until the number in the UCB equals the number in the JFCB.
4. When multiple tape units are used and a volume switch causes processing to be continued on a volume on a different unit, the EOVR routine copies the data set sequence number from the previous UCB to the current UCB.

No more than one data set on a tape volume may be open at any given time. If you attempt to begin processing a second data set on the same volume, processing is abnormally terminated.

### ***Read Backward***

For the read backward (RDBACK) operation, the data records are retrieved in reverse sequence. Multivolume data sets can be read backward. Concatenated data sets cannot be read backward. Format V (variable length) records cannot be read backward. Seven-track tape with data conversion cannot be read backward.

### **End of Data or End of Volume on Input**

For input, data management's EOVS routine handles both end-of-data-set and end-of-volume conditions. These conditions occur when:

- A tapemark is read or
- A force-end-of-volume (FEOV) macro instruction is executed by the processing program.

### ***Determining Volume Switch***

The serial numbers of all volumes of the data set to be processed must be specified by the user at execution time. The serial numbers are specified either directly in the DD statement or indirectly through the catalog procedure. You specify the serial numbers in forward sequence, regardless of whether the tapes are to be read forward or backward.

- For noncataloged data sets, you specify the volume serial numbers in the VOLUME parameter of the DD statement. Data management processes the group of volumes in whatever order you specify and processes only the volumes you specify.
- For cataloged data sets, the group of volumes must be processed in Sequential order. However, you can begin processing at any volume of the group by specifying a sequence number in the VOLUME parameter of the DD statement.

For input, the volume serial numbers specified by the user are the bases for determining whether a volume switch is required. Data management does not consider whether the data set on the current volume is followed by one or two tape marks. To determine whether additional volumes are required, data management maintains a volume sequence number in the data extent block (DEB) in storage.

- For read forward operations, the volume sequence number in the DEB is incremented as each volume is processed. This count is compared to the total number of volumes requested, as shown in the JFCB.
- For read backward operations, the volume sequence number in the DEB is set to the number of volumes requested, as shown in the JFCB. This count in the DEB is decremented as each volume is processed until the count equals 0.

If another volume is required, data management obtains the next volume serial number from the JFCB and switches volumes. Data management checks the initial record of the new volume to ensure that it is not a standard volume label and positions the tape to the data set. For a multivolume data set, the tape is positioned to the first record on the new volume. For a concatenated data set, the tape is positioned according to the specified data set sequence number.

If another volume is not required, control is given to the user's end-of-data-set routine that is specified in the data control block. Subsequently, the processing program or the operating system closes the data set.

- The user's end-of-data-set routine is not entered until the last specified volume or the last concatenated data set is processed.
- If an input data set is closed before it reaches the end of the data set, the user's end-of-data-set routine is not entered.

## Opening an Output Data Set

When you specify unlabeled tape, data management checks the output tape to ensure that the existing first record is not a standard volume label. If the first record is 80 bytes in length and contains the identifier VOL1 in the first 4 bytes, data management checks for the following conditions, in the order presented: (1) specific volume request, (2) security protection, and (3) expiration date.

If data management determines that a specific volume is requested or that the volume is security-protected, a message to demount the tape is issued to the operator. Otherwise, data management continues processing by checking the expiration date. If the expiration date is low, a message is issued, and the operator can either refuse or allow the use of the tape. If the expiration date is high, another message is issued, and again the operator can refuse or allow the use of the tape. If in either situation the operator refuses the use of the tape, data management requests that another volume be mounted. If the operator accepts the tape, data management then destroys the standard label by writing a tapemark over it, thus providing you with the unlabeled tape that you requested. If you do not want data management to perform this checking for you, you should insert your own label editor routine, as discussed in "Volume Label Verification and Volume Label Editor Routines." The various error conditions that can occur during verification of the first record are also explained in that section of this publication.

### ***BLP (Bypass Label Processing) Option***

If you do not want data management to check an output tape for an existing standard label, you must specify BLP (instead of NL) in the LABEL parameter of the DD statement. In all other respects, tape processing under BLP is the same as if NL were specified. The BLP feature must also be specified in the cataloged procedure, which controls the system reader/interpreter. The BLP feature (which is an option in the PARM field of the EXEC statement of the procedure) is not specified in the IBM-supplied cataloged procedures for the reader. The "System Reader, Initiator, and Writer Cataloged Procedures" section of either the *OS MVT Guide* or the *OS MFT Guide* shows how to override parameters in IBM-supplied procedures or write new procedures. Note that if the BLP feature is not specified in the reader cataloged procedure, specifying BLP in the LABEL parameter of a DD statement has the same effect as specifying NL.

The BLP feature is designed mainly to process blank (unused) tapes. You may wish to write a tapemark, data, or a label on the blank tape. If BLP is not coded, data management will read through an entire blank tape looking for the first record.

There are other reasons for using the BLP feature. For example, you may want to overwrite a 7-track tape that differs from your current parity or density specifications. If such a tape is mounted, data management makes 4 attempts to read the initial record (to determine that it is not an IBM standard label) before accepting the tape. You can eliminate the 4 read operations by specifying BLP instead of NL.

## ***Volume Serial Number***

You are not required to specify volume serial numbers for unlabeled output tapes. If none is specified, the mount message directs the operator to mount a scratch tape.

If you request a specific volume, the operating system uses the specified volume serial number for mounting messages, for cataloging, and for passing the volumes to other job steps.

If you do not request a specific volume, the system cannot obtain the actual serial number of the volume that is mounted. In this case, the system generates a volume serial number and assigns it to the volume. Unless you specified `DISP=PASS`, these volume serial numbers are generated in the form `Lxxxxyy` where:

- `xxx` is a number the Open routine increments (by one) each time an output data set is opened on a nonspecified unlabeled volume. If more than one data set is created on the same volume, this number is incremented only when the first data set is opened.
- `yy` is set to 00 by the Open routine. The EOV routine increments this number (by one) each time an end-of-volume condition occurs. In this way, each volume of a multivolume data set is assigned a different volume serial number.

If you specified `DISP=PASS`, the system generates a 6-digit serial number beginning with an L.

If a data set is to be cataloged, you should specify the volume serial numbers for all the volumes required. This prevents different data sets residing on different volumes from being cataloged with identical volume serial numbers, which could result in the mounting of wrong volumes.

## ***Positioning the Volume to the Data Set***

When the tape is accepted for output, it is positioned to receive the new data set. Usually the new data set is the first or only data set on the volume, so the tape is positioned at load point.

To create a data set that follows another data set already stored on the volume, you specify a data set sequence number in the `LABEL` parameter of the `DD` statement.

- The sequence number can be from 1 to 9999 with 1 representing the first data set on the volume. If you specify a sequence number that is 2 greater than the number of data sets existing on the volume, one of two things may happen: (1) the tape will be spaced through and removed from its reel or (2) the data set will be written but separated from the preceding data set by unusable (old) data.
- If you do not specify a sequence number, or if you specify 0, data management assumes that the data set is to be written as the first one on the volume.

To position the tape, data management maintains a logical data set sequence number in the unit control block (UCB). The method of positioning is the same as that previously explained for opening an input data set.

No more than one data set on a tape volume may be open at any given time. If you attempt to open a second data set on the same volume, processing is abnormally terminated.



## End-of-Volume on Output

Data management's EOVS routine automatically switches volumes when an end-of-volume condition occurs on output; that is, when the reflective strip is encountered at the end of a tape or when an FEOV macro instruction is executed.

The EOVS routine writes one tapemark after the data set on the current volume and checks the new volume to ensure that it does not contain a standard volume label. The output is then continued on the new volume.

## Closing an Output Data Set

The Close routine handles end-of-data-set processing on output tapes. When a write operation is the last operation that occurs before closing a data set (for OUTPUT, OUTIN, or INOUT) or when no output is written before closing (for OUTPUT or OUTIN), the Close routine creates data set trailer labels.

## Restarting from a Checkpoint

When a job step is restarted from a checkpoint, the Restart routine repositions any tape volumes containing data sets that were open when the checkpoint was taken. Specifically, the Restart routine:

1. Restores applicable control blocks to the conditions that existed when the checkpoint was taken.
2. Ensures that the first existing record on the tape is not a standard volume label (VOL1).
3. Uses the data set sequence number shown in the JFCB to position the tape at the required data set. The method of positioning is the same as previously explained for opening an input data set.
4. Uses the block count shown in the DCB to reposition the tape at the proper record within the data set. For forward read operations, this positioning is performed in a forward direction (if the block count is zero or negative, the tape remains positioned at the interrecord gap preceding the first record). For backward read operations, this positioning is performed in a backward direction (if the block count is 0 or a positive number, the tape is positioned at the interrecord gap following the last record of the data set).



## VOLUME LABEL VERIFICATION AND VOLUME LABEL EDITOR ROUTINES

If you specify that an input or output tape has an IBM or ANSI standard label, the operating system checks for the corresponding standard volume label at the beginning of the tape. If you specify that the tape has nonstandard labels or no labels, the system attempts to verify that the first record is not a standard volume label. Due to conflicting label types or conflicting tape characteristics, various error conditions can occur during this verification of the first record. Under some error conditions, the tape is accepted for use. Under some error conditions, the tape is not accepted and the system issues another mount message. For certain other error conditions, the system gives control to a volume label editor routine—your installation can use routines supplied by IBM or it can supply its own routines. The IBM-supplied volume label editor routines determine the discrepancies between the requested tape and the mounted tape and, if necessary, pass control to the appropriate data management routine to create or destroy labels, as required. Installation-supplied routines can perform other functions.

### Verification of First Record

The system reads the first record on the tape in accordance with the following criteria:

- If a single-density 9-track tape unit is used, the record is read in the density (800 BPI or 1600 BPI) of the unit. If the record is not in the density of the unit, or if it is a 7-track record, unit check occurs.
- If a dual-density 9-track tape unit is used, the record is read in its existing density. If the record is a 7-track record, unit check occurs. If no corrective action is taken on the unit check, the sensing facility of the dual-density feature will assume that a 7-track tape is an 800 BPI 9-track tape—because the sensing facility will not receive the signal that indicates 1600 BPI.
- If a 7-track tape unit is used, the first record is read in the density specified by the user and in the translate on, even parity mode. If the record is in another density or mode, or is a 9-track record, unit check occurs. Seven-track tape is not supported for ANSI information interchange. If ASCII tape processing is specified for a 7-track tape unit, unit check occurs.

As previously explained, various error conditions can occur during the system's verification of the initial record on a tape. The system actions resulting from these error conditions are shown by flowcharts in Figures 28, 29, and 30. Figure 28 shows the actions when standard labels are specified. Figure 29 shows the actions when nonstandard labels are specified; Figure 30 shows the actions when no labels are specified.

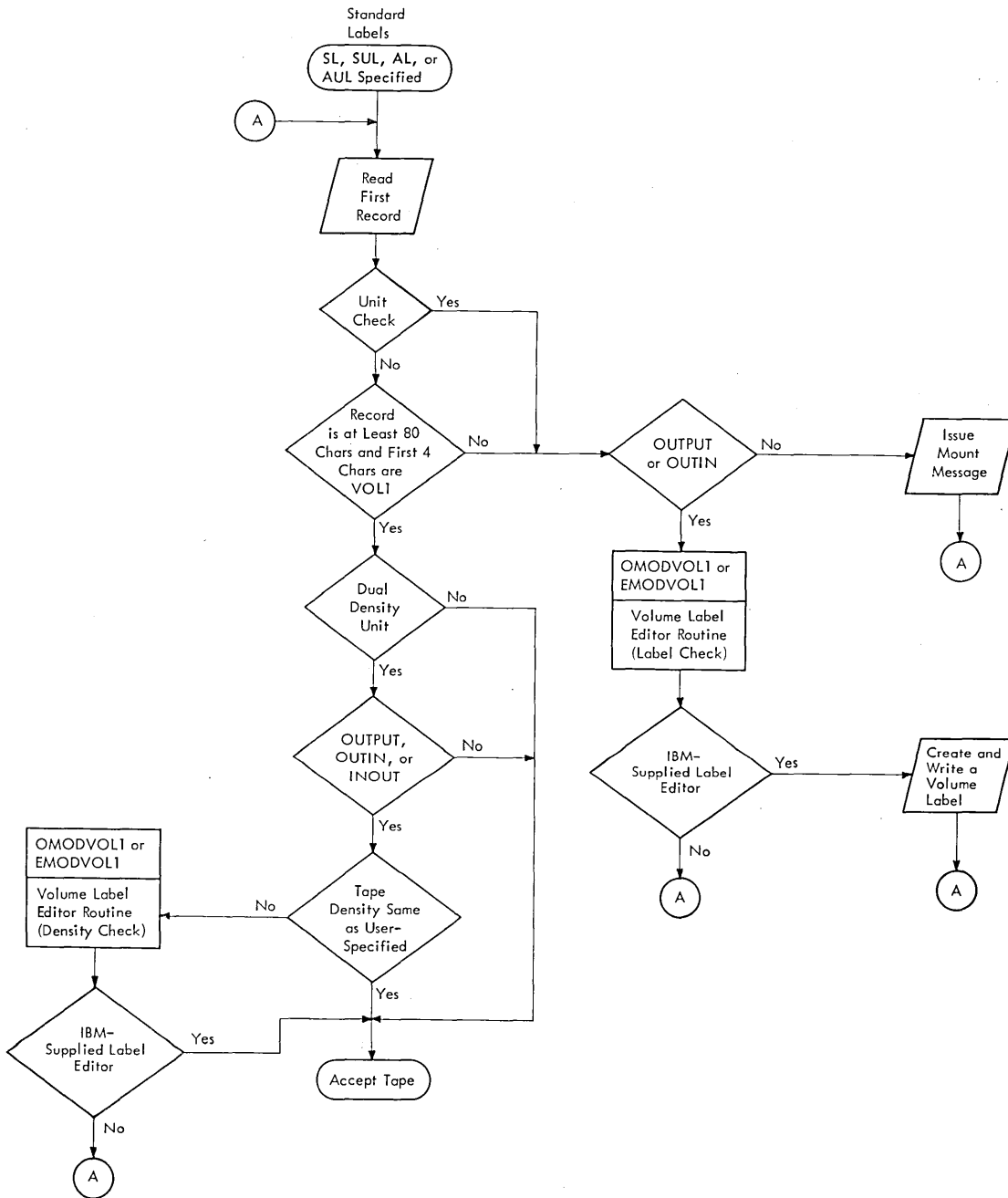


Figure 28. Verification of First Record When Standard Labels are Specified

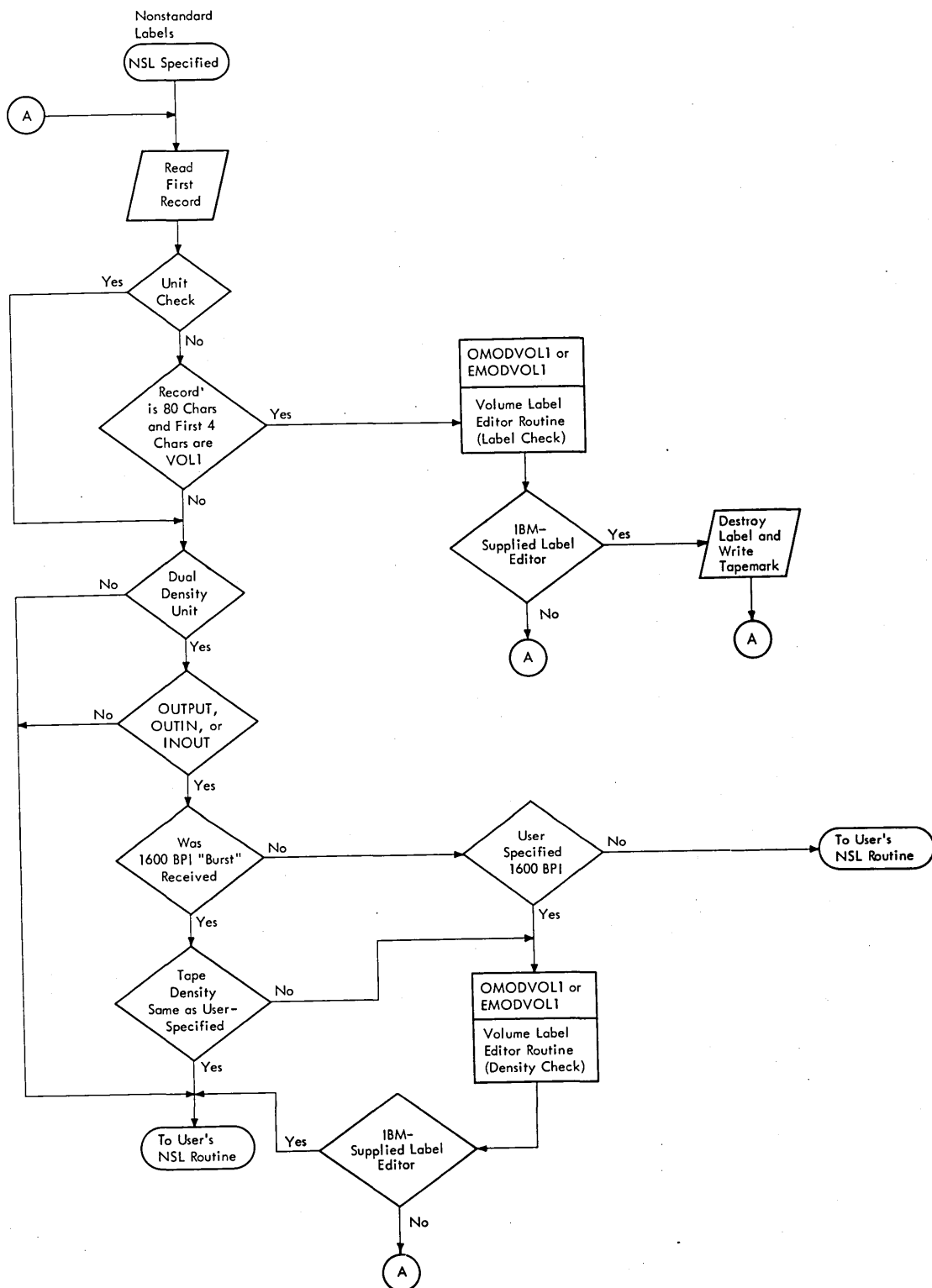


Figure 29. Verification of First Record When Nonstandard Labels are Specified

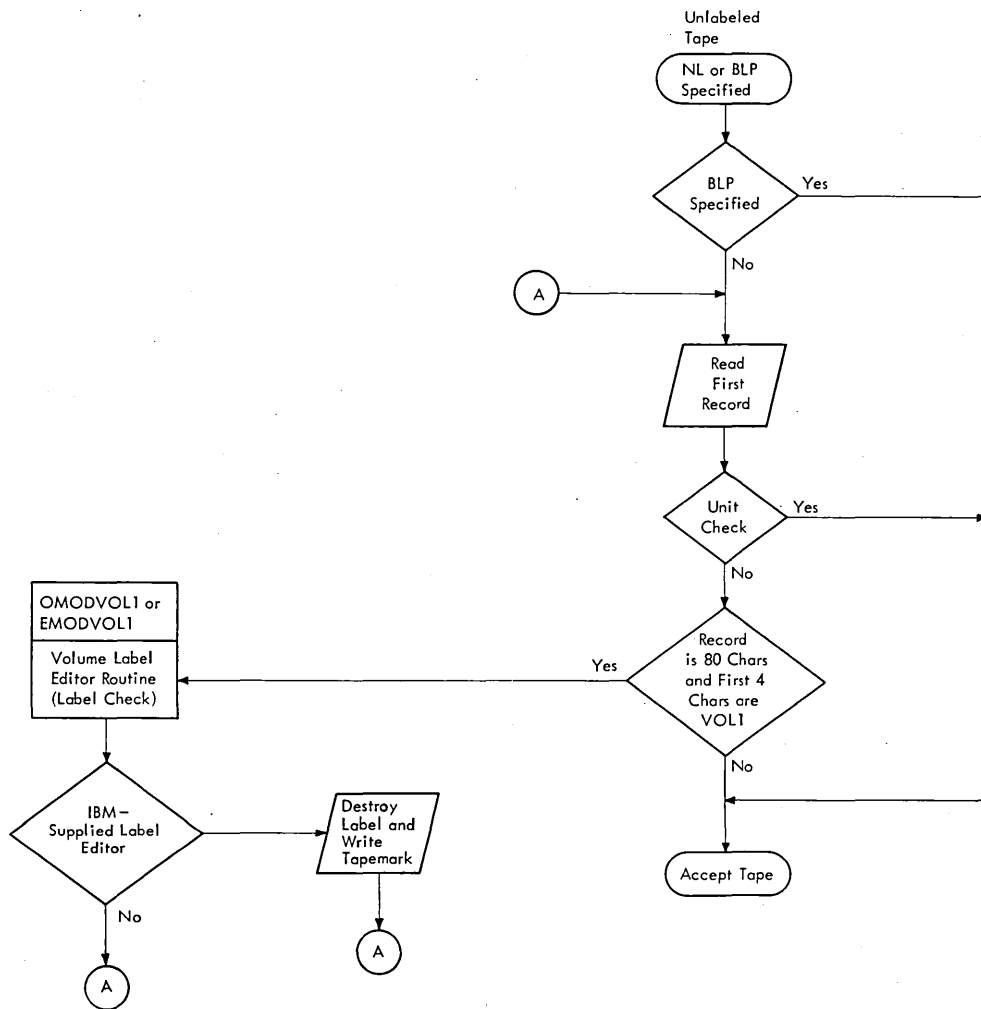


Figure 30. Verification of First Record When Unlabeled Tape Is Specified

## Volume Label Editor Routines

When data sets are written on tape, data management's Open or EOVS routine may detect conflicts between:

- The label type specified by the user and the actual label type on the mounted output volume (OUTPUT or OUTIN).
- The recording density specified by the user and the actual density of the output volume (OUTPUT, OUTIN, or INOUT) mounted on a dual-density tape unit.

When such conflicts occur, control is given to the volume label editor routines. The IBM-supplied editor routines determine whether the data management routines can resolve the conflict.

If a nonspecific volume request is made for an IBM or ANSI standard labeled tape, but the mounted volume does not have a standard label, data management issues a message to the operator requesting that he supply the volume serial number and owner information, or, optionally, that he refuse the use of this tape volume.

**Note:** If a specific volume request is made and the label format of the mounted volume does not match the format specified in the processing program, data management will reject the tape and issue a message to mount another volume.

If a nonspecific volume request is made for a nonstandard labeled or unlabeled tape, but the mounted volume has an IBM or an ANSI standard label, data management gives the operator the option to allow or refuse the use of the tape under the following conditions:

- The file sequence number is not greater than 1.
- The expiration date has passed, or the operator has allowed the use of the tape.
- The volume is not security-protected.

If the preceding conditions are not met, data management rejects the tape and issues a mount message. Data management follows the same procedure if the conditions are met, but the operator refuses the use of the tape.

If the operator accepts the tape, data management destroys the volume label by overlaying it with a tapemark.

**Note:** Even if the password is known, a security-protected tape is not converted to NL or NSL.

For dual-density standard labeled tapes, data management rewrites the labels in the density specified, unless the data set is not the first one on the volume, that is, unless the file sequence number is greater than 1.

You can replace the IBM-supplied editor routines with installation routines that resolve the conflict to your own specifications. Your editor routines can resolve label and density conflicts by writing labels, by overwriting labels with a tapemark, and by performing write operations to set the correct density on a dual-density tape device. Or, your editor routines can reset the appropriate system control blocks (in effect, change the program specifications) to agree with the label type and/or density of the currently mounted volume. Or, your installation may desire a combination of these actions, including demounting of the volume under certain conditions. You may include all of these possible actions in the design of your editor routines.

There are two IBM-supplied editor routines. One gets control from the Open routine for handling the first or only volume of a data set. The other gets control from the EOVS routine for handling the second and subsequent volumes of a multivolume data set. You can replace either or both of these routines.

The balance of this section provides you with the information necessary to write editor routines and insert them into the control program.

### ***Programming Conventions***

Your editor routines must conform to the same general programming conventions as the nonstandard label processing routines discussed in "Nonstandard Labels" in this manual (see "Programming Conventions") in so far as size, design, register usage, entry points, and work areas are concerned. As discussed in "Nonstandard Labels," you must use the EXCP technique to perform needed input/output operations.

You must name the first (or only) module of your routines as follows:

OMODVOL1 The editor routine associated with Open

EMODVOL1 The editor routine associated with EOVS

If your editor routines consist of more than one load module, names for the additional modules must begin with the prefix OMODVOL for the Open routine, or EMODVOL for the EOVS routine. Transfer between the modules must be by name.

## ***Program Functions***

Figure 31 presents the four conditions under which the Open or EOJ routines transfer control to your editor routines. Each condition suggests a general action that your routine could take to permit processing of the current volume to continue. The first two conditions (density checks) arise only when the tape volume is mounted on a dual-density tape device.

General flowcharts of editor routines are shown in Figures 32 and 33. These flowcharts suggest the logic that you could use in your routines. The logic is shown separately for routines that receive control from the Open or EOJ routines of the control program. Each block in the flowcharts is numbered, and the number corresponds to an item in the list of explanations that follows. Other items to note are:

- The logic in the flowcharts is oriented towards resolving the label and density conflicts by altering the characteristics of the mounted volume.
- Figure 33 (the EOJ editor routine) does not contain logic blocks corresponding to blocks 5, 18, and 19 in Figure 32 (the Open editor routine). These blocks represent functions that you must program only when receiving control from the Open routine. You must test all the DCBs defined by the OPEN macro instruction before returning control to the Open routine. When you receive control from the EOJ routine, there is only one DCB to process.
- If your installation does not support expiration date and protection checking on nonstandard label volumes, and does not desire to maintain such checking on standard label volumes, you need not implement the functions of logic blocks 6 through 14 in the flowcharts.



Program Specification	Mounted Volume Characteristics	Transfer Occurs On A	Possible Editor Routine Action
AL or SL-800 BPI or 1600 BPI density	AL <sup>1</sup> or SL—CD	Density Check <sup>2</sup>	Overwrite the standard label with a standard label. The first write from load point sets the recording density on a dual-density device. (See Figure 32 or 33—blocks 15B, 16, and explanation).
NSL <sup>3</sup> -800 BPI or 1600 BPI density	NSL or NL—CD	Density Check <sup>2</sup>	Write a tapemark to set density. The program specification NSL will cause control to be given to your nonstandard label routines after return to Open or EOVS. (See Figure 32—blocks 15, 15B, and 16. If your installation supports protection and retention date checking on NSL volumes, see block 6.)
AL or SL	NSL or NL <sup>4</sup>	Label Check <sup>5</sup>	Write a standard volume label. (See Figure 32—blocks 15, 15A, and 16. If your installation supports protection and retention date checking on NSL volumes, see block 6.)
NL or NSL	SL <sup>4</sup> or AL <sup>1</sup>	Label Check	Overwrite standard label with tapemark, for example, cancel. (See Figure 32—blocks 15, 15A, and 16.) Depending on whether NL or NSL is specified by the program. Open or EOVS will either position tape (NL) or transfer control to your nonstandard label routines (NSL) when control is returned to them.

**Legend:**

- AL ANSI standard volume label.
- SL IBM standard volume label.
- NSL Nonstandard volume label.
- NL No volume label.
- CD Conflicting density. The volume has been previously written in a recording density other than that specified by the program.

<sup>1</sup> ANSI standard labeled tape cannot be overwritten without the permission of the console operator.

<sup>2</sup> Dual-density devices only.

<sup>3</sup> If NL is specified, no density check is performed. For NL volumes, tape is positioned at load point and recording density is set by the first write command.

<sup>4</sup> If the volume is mounted on a dual-density device, a density condition may also exist. It will be corrected by the write operation.

<sup>5</sup> When SL is specified, a label check may also indicate that the system could not recognize the first record because of a unit check condition.

Note: The Open and EOVS routines position the tape at load point before transferring control to the editor routines.

**Figure 31. Editor Routine Entry Conditions from the EOVS Routine**

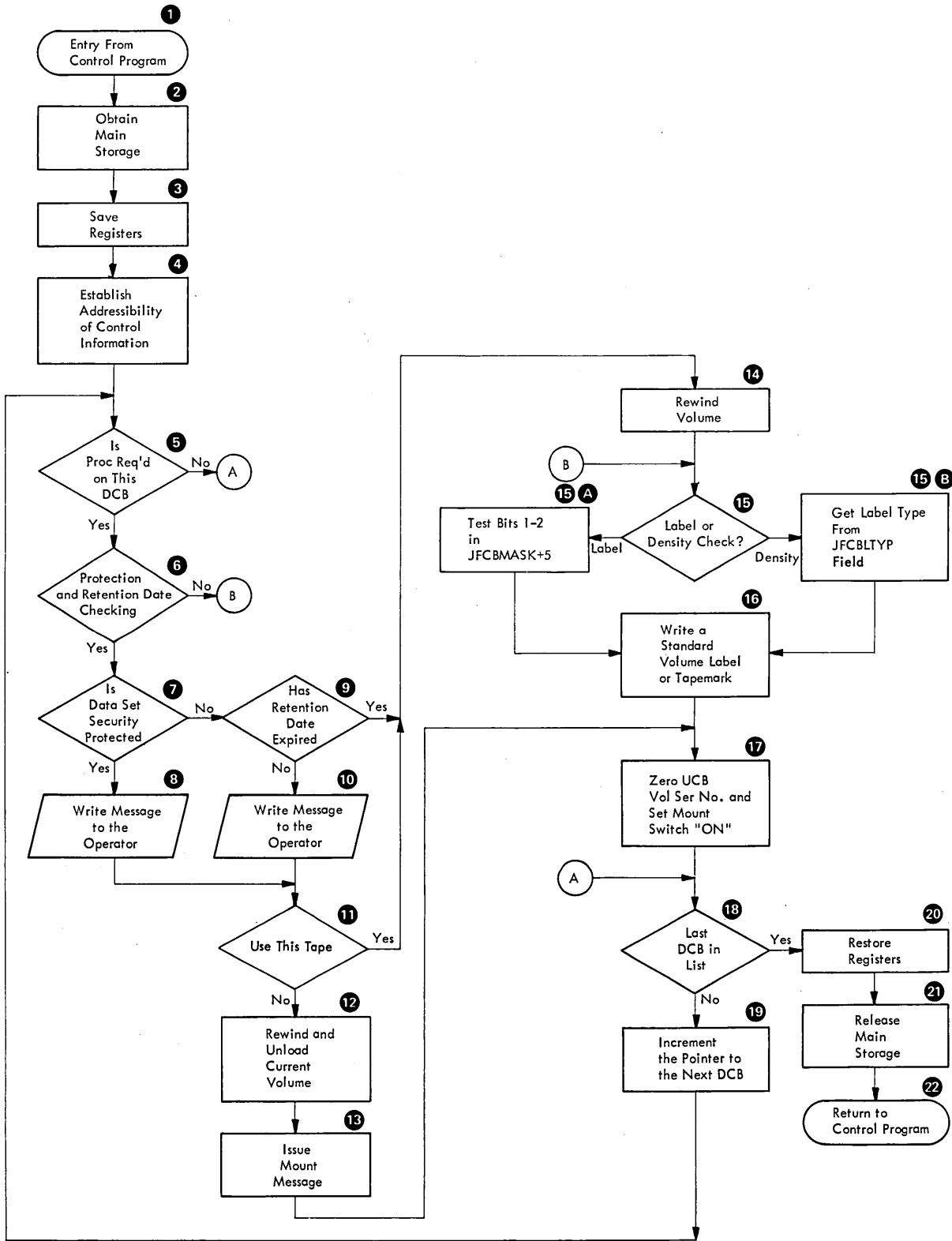


Figure 32. General Flow of an Editor Routine after Receiving Control from the Open Routine

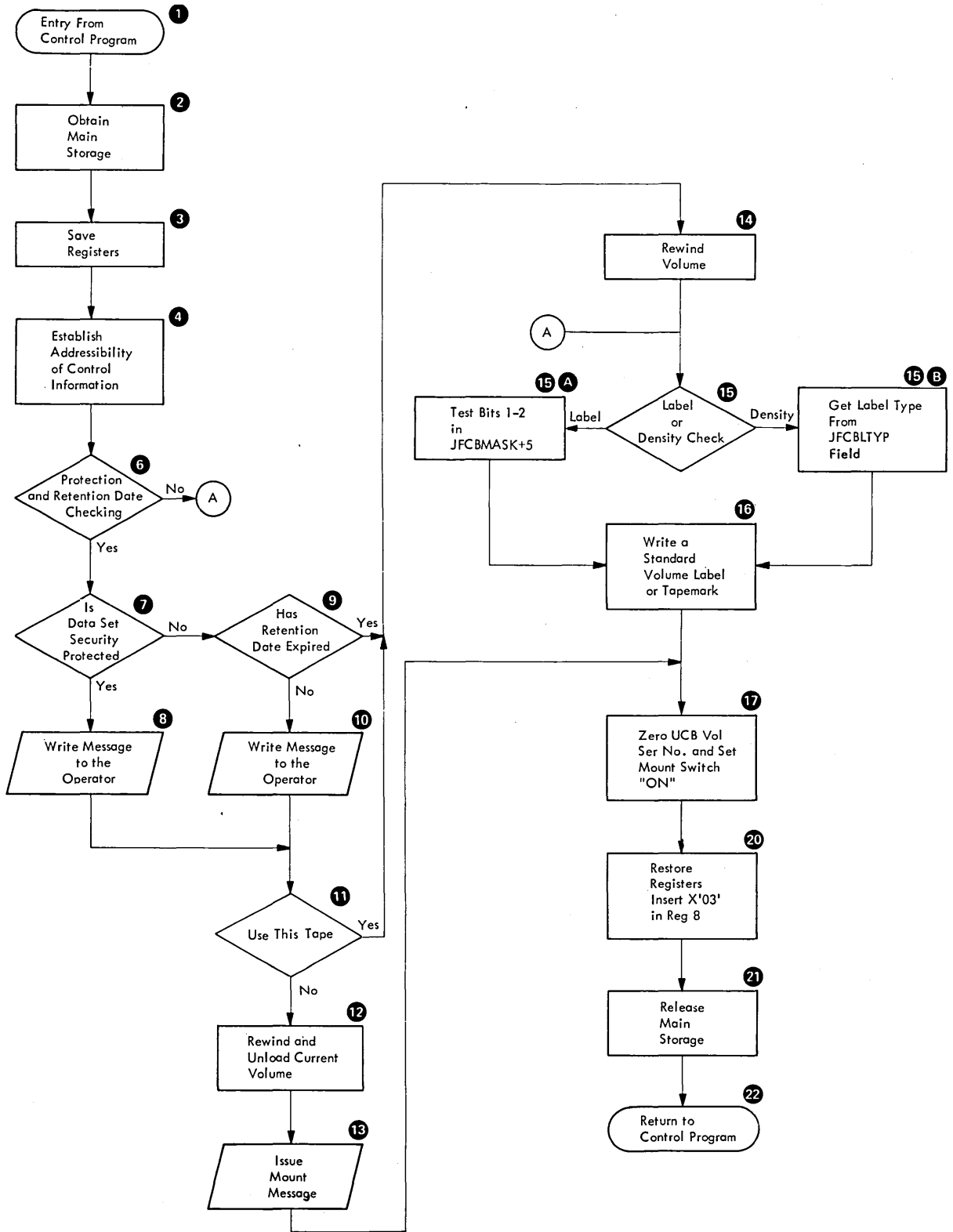


Figure 33. General Flow of an Editor Routine after Receiving Control from the EOVRoutine

## Logic Block Explanations

- 1 — Your exception routine receives control via an XCTL macro instruction issued by the Open or EOVR routines of the control program.
- 2 — Use the GETMAIN macro instruction. The main storage you obtain must contain all your work areas, including those used to read in a label or write a label.
- 3 — Use the store multiple (STM) instruction.
- 4 — Figure 20 in “Nonstandard Labels” in this manual provides the information you need to establish addressability of the DCB address list and work and control block area for each DCB defined by the OPEN macro instruction.

When you receive control from the EOVR routine, general register 2 contains the address of the DCB for the data set, and general register 4 contains the address of the work and control block area associated with the DCB.

The IECDSECT macro instruction symbolically defines the fields of the work and control block area (see Figure 21 in “Nonstandard Labels”).

You will also need to address the unit control block (UCB) for the device on which the tape volume is mounted. The address of the UCB may be obtained from the DXDEBUCB field of the data extent block defined by the IECDSECT macro instruction. The IEFUCBOB macro instruction defines the fields of the unit control block.

- 5 — Bit configurations in the byte addressed by JFCBMASK +5 indicate whether label checks or density checks have occurred, and, in the case of a label check, the condition that caused the check. At this point, you test bits 0 and 3. If either bit is set to 1, processing is required. However, if bits 6 and 7 of DCBOFLGS are set to 0, you should discontinue processing. When bit 6 (lock bit) is 0, the control program cannot open the DCB. When bit 7 (busy bit) is 0, the DCB is already being processed or is already open.

The field JFCBMASK is defined by the IECDSECT macro instruction. Bit settings in the byte at JFCBMASK+5 are defined as:

Bits	Setting	Meaning
0	1	Label check has occurred.
1	1	Standard label (SL or AL) specified; no label/nonstandard label on mounted volume.
2	1	No label (NL) or nonstandard label (NSL) specified; standard label (AL or SL) on mounted volume.
3	1	Density check has occurred.
4-7	-	Reserved for future use.

- 6 — If your installation supports a protection and retention date scheme involving nonstandard labels, and/or you want to maintain retention date and protection checking on standard labels, you must incorporate code in your editor routines to check for protection and retention date expiration.

If checking is desired, you must, at this point, read the first record and determine the label type.

To perform the I/O operation, move your CCWs into the channel program field of the work and control block area. The symbolic name for the first entry in this field is DXCCW. Then issue an EXCP macro instruction specifying the address of the control program's input/output block (IOB). The symbolic name for the IOB is DXIOB. These fields (DXCCW, DXIOB) are defined by the IECDSECT macro

instruction. *Note:* There are twelve CCW locations in the DXCCW field. You can only use the first six locations.

- 7 — To check the retention date and/or protection fields in a standard label, you must read the data set header 1 record into a work area. The format of the nonstandard label defined by your installation determines how you access the protection and retention date fields in the nonstandard label. Step 6 provides directions for handling the I/O operation.
- 8 — Write a message to the operator to inform him that the volume is protected and to determine if it is to be used.
- 9 — See step 7 above.
- 10 — Write a message to the operator to inform him that the expiration date for the mounted volume has not elapsed and to determine if it is to be used.
- 11 — If the volume is to be used, continue processing to resolve label or density conditions.
- 12 — Rewind and unload the currently mounted volume. Step 6 provides directions for handling the I/O operation. When you issue the rewind and unload command, you must turn on the UCB not-ready bit (UCBFL2) after the ECB has been posted.
- 13 — Write a message to the operator requesting demounting of the current volume and mounting of a new volume. The device name (in EBCDIC) may be obtained from the UCBNAME field of the unit control block.
- 14 — Step 6 provides directions for handling the I/O operation.
- 15 — Test bit 3 of the byte at JFCBMASK+5. If set to 1, control was received as a result of a density check.  
  
Test bit 0 of the byte at JFCBMASK+5. If set to 1, control was received as the result of a label check.
  - a—If control was received as the result of a label check, test bits 1 and 2 of the byte at JFCBMASK+5. See step 5.
  - b—If control is received as the result of a density check, use the JFCBLTYP field in the job file control block (JFCB) to ascertain the type of label specified in the program. A hexadecimal 04 indicates a nonstandard label (NSL) has been specified; a hexadecimal 02 indicates that a standard label has been specified.
- 16 — When correcting a density check or label check condition, and a nonstandard label (NSL) or no label (NL) is specified by the program, you must write some kind of record on the tape that will be interpreted by the Open or EOVR routines as a nonstandard label or no label; for example, it does not contain VOL1 in the first four bytes of the record. The easiest way to do this is to write a tapemark. Upon return to Open or EOVR and reverification of the label, the specification for label type and density will have been met. Open or EOVR will transfer control to your nonstandard label routines if NSL is specified, or position the tape for writing if NL has been specified.

The "IBM Standard Labels" section in this publication contains the format and field descriptions for IBM standard tape labels. "American National Standard Labels" contains the volume and field descriptions for ANSI standard tape labels. You must supply information for the label identifier, the label number, and the volume serial number fields, and record the balance of the label as blanks.

You enter VOL in the label identifier field, a 1 in the label number field, and a 6-character serial number in the volume serial number field. *Note:* To ensure that two

or more tape volumes carrying the same serial number are not produced, write to the operator at this point for assignment of a serial number.

Data set header labels 1 and 2 are constructed by the Open or EOV routines after control is returned to them.

*Note:* If you desire, at this point, you may change the control block settings to conform to the characteristics of the tape volume mounted (that is, reset the label type field in the JFCB to conform with the type of label on the volume mounted and change the density field in the DCB to the density of the tape mounted).

- 17 — The symbolic name for the volume serial number field in the unit control block is SRTEVOLI. The mount switch is the high-order bit of the field named SRTEDMCT in the unit control block. These fields are defined by the IEFUCBOB macro instruction. Perform an Exclusive OR (XC) operation on the SRTEVOLI field with itself and perform an OR (OI) operation on the SRTEDMCT field with X'80'.
- 18 — When receiving control from the Open routine, you must process the entire DCB list. The last entry in the list can be recognized by a 1 in bit 0 of the first byte in the entry.
- 19 — You increment the pointer to the DCB address list by 4 bytes. You must also increment the pointer to the work and control block area for each DCB. You increment this pointer by 8 bytes.
- 20 — Use the load multiple (LM) instruction. *Note:* When preparing to return to EOV, you must set register 8 to hexadecimal 03.
- 21 — Use the FREEMAIN macro instruction.
- 22 — Return control to the Open or EOV routines via an XCTL macro instruction, specifying the module to be given control as follows:

Return From	To Module
OMODVOL1	IGG0190A (Open)
EMODVOL1	IGG0550P (EOV)

*Note:* Open and EOV will rewind the volume upon receiving control from OMODVOL1 or EMODVOL1.

Return is via the XCTL macro instruction (E-form). See item 13 under "Explanation of Logic Blocks—Figure 26" under "Nonstandard Labels."

## Inserting Your Label Editor Routines Into the Control Program

You insert your editor routines into the control program after system generation by making a linkage-editor run against the system library named SYS1.SVCLIB. You will be replacing the IBM-supplied editor routines OMODVOL1 and/or EMODVOL1 with your routines.

The setup for making the linkage editor run is shown below.

```

      .
      .
      .
//jobname      JOB           [parameters]
//stepname     EXEC         PGM=IEWL[,parameters]
//SYSPRINT    DD           SYSOUT=A
//SYSUT1      DD           UNIT=SYSDA,SPACE=(parameters)
//SYSLMOD     DD           DSNAME=SYS1.SVCLIB,DISP=OLD
//SYSLIN      DD           *
      .
      .
      .

```

**object deck for Open**

```

      .
      .
      .
ENTRY          OMODVOL1
NAME           OMODVOL1(R)

```

**object deck for EOVS**

```

      .
      .
      .
ENTRY          EMODVOL1
NAME           EMODVOL1(R)
/*

```

**Note:** You should not attempt to insert routines into the SVC library when you are running in a multitasking environment. Also, if your editor routines (for Open or EOVS) consist of more than one module, you must have requested space for the SVC library directory entries for the additional modules at the time the system was generated.





## USING TAPE VOLUMES CREATED BY OTHER SYSTEMS

Occasionally, it may be necessary to process a tape volume that was created by another system. There is no exact procedure: many of the factors vary according to the situation and the user's options at the time the volume was created. The volume may be slightly or extremely different in its organization, its label formats, or its label contents. With the aid of this publication, a careful analysis of these factors will enable you to determine if the volume can be processed by your operating system. In some cases, certain modifications may be needed or restrictions observed. If tape volumes are to be transferred permanently to the operating system, it is recommended that you use the operating system to create new labels and volume organizations.

### IBM System/360 Standard Labels

Other IBM System/360 programming systems create tape volumes with the same standard label formats as the operating system. However, the actual contents of each label field may vary from system to system. Figures 7, 8, and 9 in "IBM Standard Labels" in this manual show which fields of each label are functional for the operating system. Check the processing of these functional fields against the actual contents of the labels you want to use. This comparison should indicate whether the volumes are compatible or what modifications must be made. Special attention should be given to the following fields of data set label 1 (HDR1, EOVI, EOF1):

- *Block Count*—The HDR1 label should contain zeros, and the trailer label should contain the actual count of data blocks in the data set on the current volume. Tapes from some other systems may contain blanks instead of zeros in the block count field of the HDR1 label. These blanks will cause a block count discrepancy, because the operating system uses the zeros in the HDR1 label to verify the count after processing (for backward read). The HDR1 label is not used to initialize the block count field in the data control block (for forward read).

If the HDR1 label of the data set you need contains blanks in the block count field, you can use the DCB exit routine to initialize the block count field in the data control block to zeros. This will permit system verification of the block count for forward read operations. Another method of processing the data set, for forward or backward read operations, is to provide a user exit entry (in the DCB exit list) that allows processing to continue when a block count discrepancy occurs.

- *Data Set Identifier*—The data set name in the label created by another system may contain embedded blanks or special characters. This name is compared to the data set name that you specify in the DD statement; therefore, you must enclose the name in apostrophes on the DD statement that requests this data set. *OS Job Control Language Reference*, lists the restrictions that apply to enclosing a data set name in apostrophes. The apostrophes do not appear in the data set identifier field.

To match the name in the label, you may have to modify the job file control block after the DD statement is recorded there.

The operating system (and the System/360 Time Sharing System) can obtain certain data set characteristics from the standard data set label 2 (HDR2/EOV2/EOF2). Other System/360 programming systems do not use or create data set label 2. The absence of data set label 2 does not interfere with normal processing by the operating system, as long as the label information is specified by some other means. The functional information in data set label 2 (record format, block length, record length, tape recording

technique, and printer control characters) can be furnished to the operating system either in the DCB macro instruction or the DD statement.

Labels created by systems other than System/360 or System/370 programming systems should be treated as nonstandard labels, provided the first record on the tape is not identified as VOL1 and the data sets are followed by recognizable tapemarks.

## Nonstandard Labels

Nonstandard labels are labels that do not conform to the formats described in this manual. If you want to retrieve the data set and process the nonstandard labels, you must write nonstandard label processing routines and insert them into the operating system. The procedure is described in the "Nonstandard Labels" section of this manual.

If you wish to ignore the nonstandard labels, you can retrieve the data set by treating the volume as an unlabeled tape. You use the data set sequence number in the DD statement to bypass the labels and position the tape to the data set.

## Unlabeled Tapes

The operating system can process unlabeled tape volumes created by other systems provided the data sets are followed by recognizable tapemarks.

To position a tape at the desired data set, you must specify the correct data set sequence number in the DD statement. If a tapemark might precede the first data set and the LABEL subparameter LTM is specified, the system will test for and bypass, if present, a leading tapemark. If a tapemark should precede the first data set and you do not specify LTM in the LABEL parameter field, you must add 1 to the data set sequence number.

If a multivolume data set from another system has a leading tapemark on one or more of the volumes, the operating system can process it as an unlabeled multivolume data set if the LABEL subparameter LTM is specified. Otherwise, the operating system cannot process it as an unlabeled multivolume data set.

The presence of a leading tapemark makes each data set the second in sequence on the tape. However, the operating system always assumes that continued data sets are first in sequence on the tape. By specifying LTM in the LABEL parameter field, the first data set on a tape can be accessed whether or not it is preceded by a leading tapemark.

The specification of LTM in the LABEL parameter field does not make allowances for any other excess tapemarks. You must make any such adjustments in the data set sequence number.

## APPENDIX A: COMPONENT CONSIDERATIONS

Job control statements make the label processing facilities of data management available to users of the operating system's assembler, linkage-editor, sort/merge feature, utility programs, and higher-level languages. Figure 34 shows the component support for each type of label processing.

Item	Assem- bler	Linkage Editor	Sort/ Merge	Utilities	ALGOL	COBOL E	ANS V2	ANS V3	FORTTRAN	PL/I	RPG
Uses Data Management Facilities for Label Processing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard Labels (SL,AL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard User Labels (SUL, AUL)	No	No	Yes	Yes	No	No	SUL-Yes AUL-No	SUL-Yes AUL-Yes	No	No	No
Supports Nonstandard Labels (NSL) <sup>1</sup>	Yes	Yes	Yes	Yes	Yes	No <sup>3</sup>	Yes	Yes	Yes	Yes	Yes
Supports Unlabeled Tape (NL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Bypass Label Processing Option (BLP) <sup>2</sup>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Concatenated Data Sets with Unlike Attributes	No	Yes	No	No	No	No	No	No	No	No	No

<sup>1</sup>NSL can be specified only when installation-written routines that write and process the nonstandard labels have been incorporated into the operating system.

<sup>2</sup>If the BLP option is not specified at system generation, its use defaults to NL.

<sup>3</sup>Nonstandard labels may be processed in the Installation SVC routine, but may not be processed in the COBOL program.

Figure 34. Component Support of Label Processing Features



## APPENDIX B: EXTERNAL LABELS

External labels are affixed to the tape reels to provide visual identification of the volume and its contents. Normal tape volume control requires two types of external tape labels. One is a permanent label that identifies the reel; the other is a temporary label that identifies the contents.

To write on external labels, you should use an implement, like a pen or felt-tip marker, which does not produce loose residue. Do not use a lead pencil. Do not use an eraser.

### Reel Label

The reel label should be applied with a permanent-type adhesive, so that it cannot be easily removed. It is affixed when the tape is first received by your installation. The label should contain the sequential volume serial number assigned by your installation; it may also identify your installation. The volume serial numbers are used to identify the tape reel by a unique number and to file the tapes in the tape rack. An example of a reel label is shown in Figure 35.

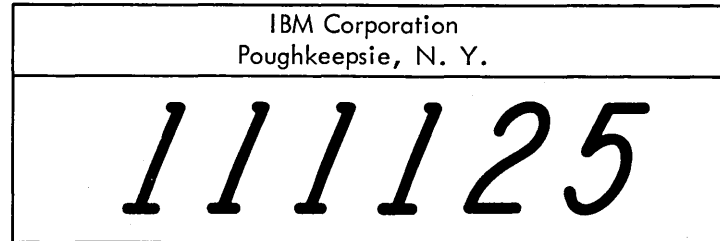


Figure 35. External Label for Reel Identification

---

### Contents Label

The contents label is used to identify the current contents of a particular volume. Since this is a temporary label, it should be applied with adhesive that is strong enough to hold the label securely and yet allow easy removal of the label when the contents of the volume change. The size and placement of the label should not interfere with the handling of the tape.

This label is applied when data is written on the volume and contains identifying information to ensure that the contents of the volume can be easily distinguishable. Your installation determines the format of the label. The information entered in the label is usually furnished partly by the programmer and partly by the operator. Examples of contents labels are shown in Figure 36.

REEL NUMBER			PROGRAMMER'S NAME, DEPT., BLDG.		
			DATE		
			SYSTEM		
SCRATCH DATE					
DENSITY	PARITY	TRACK	TAPE DESCRIPTION		

DESC.		JOB #	
REEL NO.	CREATN DATE	RET I	OP I
		NO	FILE IDENTIFIER

Figure 36. External Labels for Contents Identification

## APPENDIX C: RESTART WORK AREAS

This appendix describes the restart table entry and the restart work and control block area. When your nonstandard label processing routine receives control from the control program's Restart routine, register 9 contains the starting address of the table entry associated with the data set. The TABSEGAD field of the table entry points to the starting address of the work and control block area associated with the same data set.

### Table Entry

Figure 37 shows the format of the restart table entry. A description of each field follows.

Offset	Field Name	Bytes	Field Description																					
0(0)	TABDSORG	1	This field describes the data set organization used: <table border="1"> <thead> <tr> <th>Bits</th> <th>Settings</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Indexed Sequential organization.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Physical sequential organization.</td> </tr> <tr> <td>2</td> <td>1</td> <td>Direct organization.</td> </tr> <tr> <td>3-5</td> <td></td> <td>Reserved for future use.</td> </tr> <tr> <td>6</td> <td>1</td> <td>Partitioned organization.</td> </tr> <tr> <td>7</td> <td>1</td> <td>Unmovable—the data set contains location-dependent information.</td> </tr> </tbody> </table>	Bits	Settings	Meaning	0	1	Indexed Sequential organization.	1	1	Physical sequential organization.	2	1	Direct organization.	3-5		Reserved for future use.	6	1	Partitioned organization.	7	1	Unmovable—the data set contains location-dependent information.
Bits	Settings	Meaning																						
0	1	Indexed Sequential organization.																						
1	1	Physical sequential organization.																						
2	1	Direct organization.																						
3-5		Reserved for future use.																						
6	1	Partitioned organization.																						
7	1	Unmovable—the data set contains location-dependent information.																						
1(1)	TABDCBAD	3	Address of the DCB																					
4(4)	TABFLG1	1	This field contains the following information: <table border="1"> <thead> <tr> <th>Bits</th> <th>Settings</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Data set was specified in DD statement as NULLFILE or SYSCHECK.</td> </tr> </tbody> </table>	Bits	Settings	Meaning	0	1	Data set was specified in DD statement as NULLFILE or SYSCHECK.															
Bits	Settings	Meaning																						
0	1	Data set was specified in DD statement as NULLFILE or SYSCHECK.																						

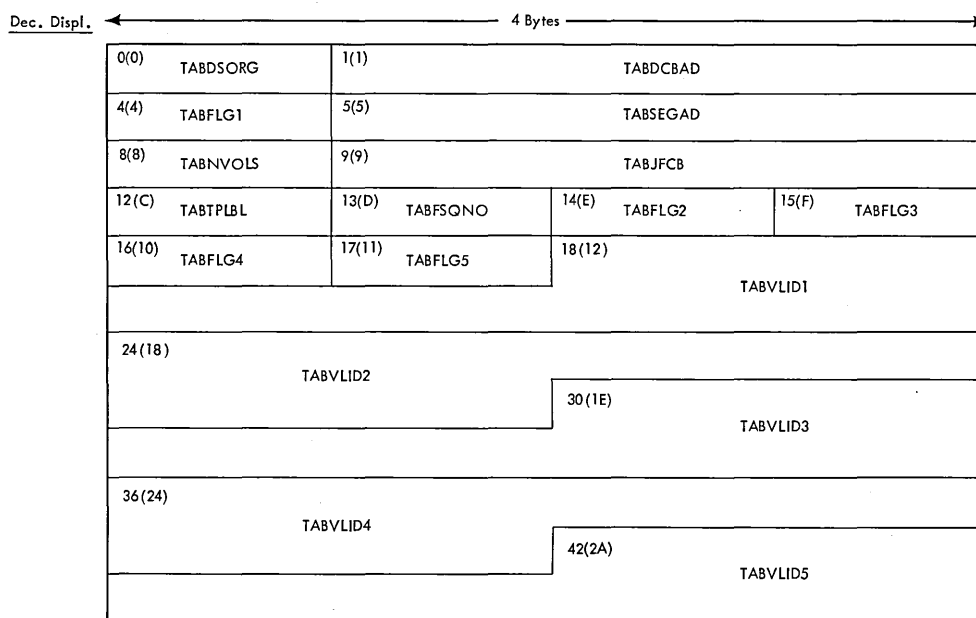


Figure 37. Restart Table Entry

Offset	Field Name	Bytes	Field Description																											
			1 1 Data set was specified in DD statement was SYSIN or SYSOUT.																											
			2 1 Device Type = direct access.																											
			3 1 Device type = tape.																											
			4 1 This is the last table entry in the Restart table.																											
			5 1 Device type = unit record.																											
			6 1 Reserved for future use.																											
			7 1 This is a DOS tape with an optional leading tapemark and/or contains embedded DOS checkpoint records.																											
5(5)	TABSEGAD	3	Address of the restart work and control block area for this data set.																											
8(8)	TABNVOLS	1	The total number of volumes for this data set, as specified in the DD statement.																											
9(9)	TABJFCB	3	The relative track address (TTR) of the JFCB.																											
12(C)	TABTPLBL	1	This field contains the following tape label information: <table border="1"> <thead> <tr> <th>Bits</th> <th>Settings</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>I/O error in NSL processing.</td> </tr> <tr> <td>1</td> <td></td> <td>Reserved.</td> </tr> <tr> <td>2</td> <td>1</td> <td>Bypass a leading tapemark if present, on an unlabeled tape. Bit 7 is also set.</td> </tr> <tr> <td>3</td> <td>1</td> <td>Bypass label processing.</td> </tr> <tr> <td>4</td> <td>1</td> <td>ANSI standard labels.</td> </tr> <tr> <td>5</td> <td>1</td> <td>Nonstandard labels.</td> </tr> <tr> <td>6</td> <td>1</td> <td>IBM standard labels.</td> </tr> <tr> <td>7</td> <td>1</td> <td>No labels.</td> </tr> </tbody> </table>	Bits	Settings	Meaning	0	1	I/O error in NSL processing.	1		Reserved.	2	1	Bypass a leading tapemark if present, on an unlabeled tape. Bit 7 is also set.	3	1	Bypass label processing.	4	1	ANSI standard labels.	5	1	Nonstandard labels.	6	1	IBM standard labels.	7	1	No labels.
Bits	Settings	Meaning																												
0	1	I/O error in NSL processing.																												
1		Reserved.																												
2	1	Bypass a leading tapemark if present, on an unlabeled tape. Bit 7 is also set.																												
3	1	Bypass label processing.																												
4	1	ANSI standard labels.																												
5	1	Nonstandard labels.																												
6	1	IBM standard labels.																												
7	1	No labels.																												
13(D)	TABFSQNO	1	Data set sequence number.																											
14(E)	TABFLG2	1	This field contains the following information: <table border="1"> <thead> <tr> <th>Bits</th> <th>Settings</th> <th>Meanings</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>More than five volumes associated with this data set.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Partitioned organization concatenation.</td> </tr> <tr> <td>2-7</td> <td></td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Settings	Meanings	0	1	More than five volumes associated with this data set.	1	1	Partitioned organization concatenation.	2-7		Reserved.															
Bits	Settings	Meanings																												
0	1	More than five volumes associated with this data set.																												
1	1	Partitioned organization concatenation.																												
2-7		Reserved.																												
15(F)	TABFLG3	1	Reserved for future use.																											
16(10)	TABFLG4	1	Reserved for future use.																											
17(11)	TABFLG5	1	Reserved for future use.																											
18(12)	TABVLID1	6	The volume serial number of the first volume to be mounted for this data set.																											
24(18)	TABVLID2	6	The volume serial number of the second volume.																											
30(1E)	TABVLID3	6	The volume serial number of the third volume.																											
36(24)	TABVLID4	6	The volume serial number of the fourth volume.																											
42(2A)	TABVLID5	6	The volume serial number of the fifth volume.																											



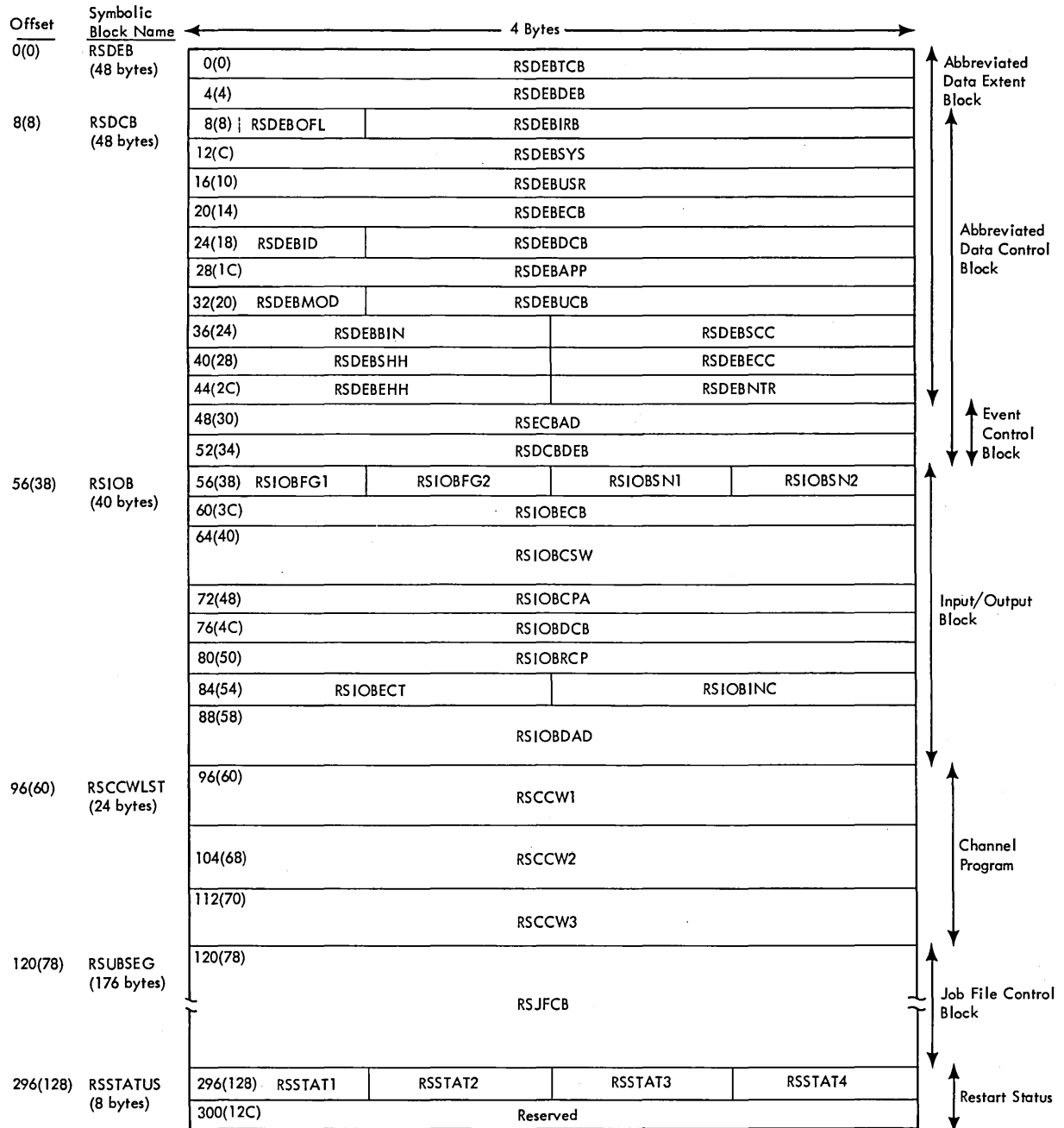


Figure 38. Restart Work and Control Block Area

## Work and Control Block Area

The following shows the format of the restart work and control block area. A description of the control block fields follows.

Offset	Field Name	Bytes	Field Description
0(0)	RSDEBTCB	4	Address of TCB for this DEB.
4(4)	RSDEBDEB	4	Address of the next DEB in the same task.
8(8)	RSDEBOFL	1	Data set status flags.
9(9)	RSDEBIRB	3	IRB address used for appendage asynchronous exits.
12(C)	RSDEBSYS	4	Address of first IOB in the system purge chain.
16(10)	RSDEBUSR	4	Address of first IOB in the user purge chain.
20(14)	RSDEBECB	4	Address of a parameter list used to locate the purge ECB for an SVC purge request.
24(18)	RSDEBID	1	A hex '0F' to identify this block as a DEB.
25(19)	RSDEBDCB	3	Address of DEB associated with this DEB (RSDCB).
28(1C)	RSDEBAPP	4	Address of the I/O appendage vector table.
32(20)	RSDEBMOD	1	Device modified.
33(21)	RSDEBUCB	3	Address of UCB.
36(24)	RSDEBBIN	2	Bin number of direct-access volume (data cell drive).
38(26)	RSDEBSCC	2	Cylinder address for start of an extent limit.
40(28)	RSDEBSHH	2	Track address for the start of an extent limit.
42(2A)	RSDEBECC	2	Cylinder address for the end of an extent limit.
44(2C)	RSDEBEHH	2	Track address for the end of an extent limit.
46(2E)	RSDEBNTR	2	Number of tracks allocated to a given extent.
48(30)	RSECBAD	4	Event control block (ECB).
52(34)	RSDCBDEB	4	Address of DEB associated with this DCB (RSDEB).
56(38)	RSIOBFG1	1	Flag byte 1, as follows:
			<i>Bits    Settings    Meaning</i>
		0-1	00    No chaining.
			01    Command chaining.
			10    Data chaining.
			11    Both command and data chaining
		2	1    Error routine in control.
		3	1    Device is to be repositioned.
		4	1    Cyclic redundancy check (CRC) needed—tape only.
		5	1    Exceptional condition—if this bit is on after the error routine returns, the error is considered permanent.
			<i>Bits    Settings    Meaning</i>
		6	1    IOB unrelated flag (that is, nonsequential).

Offset	Field Name	Bytes	Field Description
			7      0      Error recovery procedure uses channel program address at IOB START (IOB + 17).
			1      RESTART error recovery procedure uses channel program address at IOBRESTR (IOB + 24).
57(39)	RSIOBFG2	1	Flag byte 2, as follows:
			<i>Bits</i> <i>Settings</i> <i>Meaning</i>
			0          1          Halt I/O has been issued
			1          1          Sense will not be performed until the device is free
			2          1          IOB has been purged
			3          1          Home address (R0) record is to be read
			4-6      (variable)Internal I/O supervisor error correction flags
			7          1          QSAM—error recovery in control for a 2540 Punch with three buffers
58(3A)	RSIOBSN1	1	First sense byte (device-dependent).
59(3B)	RSIOBSN2	1	Second sense byte (device-dependent).
60(3C)	RSIOBECB	4	Address of the ECB to be posted (RSECBAD).
64(40)	RSIOBCSW	8	CSW.
72(48)	RSIOBCPA	4	Address of the channel program to be executed (RSCCW1).
76(4C)	RSIOBDCB	4	Address of the DCB associated with this IOB (RSDCB).
80(50)	RSIOBRCP	4	Restart address used by I/O supervisor error routines during error correction.
84(54)	RSIOBECT	2	Value used to increment block count field in DCB for magnetic tape.
86(56)	RSIOBINC	2	Used by I/O supervisor error routines to count temporary errors during retry.
88(58)	RSIOBDAD	8	This field is used for direct access only.
96(60)	RSCCW1	8	Channel program area.
104(68)	RSCCW2	8	Channel program area.
112(70)	RSCCW3	8	Channel program area.
120(78)	RSJFCB	176	Work area for job file control block.
296(128)	RSSTAT1	1	Status byte 1.
297(129)	RSSTAT2	1	Reserved for future use.
298(12A)	RSSTAT3	1	Reserved for future use.
299(12B)	RSSTAT4	1	Reserved for future use.
300(12C)		4	Reserved for future use.



# INDEX

Indexes to Systems Reference Library publications are consolidated in *OS Master Index to Reference 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

- accessibility
  - in ANSI data set 1 label 66-67
  - in ANSI VOL1 label 50,61
- American National Standard labels (see ANSI standard labels)
- ANSI standard labels
  - component support 119
  - definition 43
  - operating system support 43
  - overview of processing 47-49
  - specification of 1-2
  - types of 43
  - volume layouts for 44-45
- ASCII control character 39,71
- ASCII interchange tape
  - conversion to EBCDIC 43
  - operating system support 43
- assembler support 119
- assigning volume serial numbers
  - in JCL statements 30,60
  - system assignments 60,100
  - using IEHINITT utility program 30,60
- automatic volume recognition (AVR)
  - explained 7
  - with nonstandard labels 91
  - with seven-track tapes 10
- AVR (see automatic volume recognition)

## B

- backward read (see read backward)
- basic tape formats 2
- bits per inch (densities) 8-9
- blank tape
  - bypass label processing 99
  - IEHINITT utility
    - with ANSI standard labels 55
    - with IBM standard labels 23
  - nonstandard labels, output 78
- block count
  - with ANSI standard labels 51,52
  - with IBM standard labels 19,20,35
  - with nonstandard labels 76,79
  - in other systems 117
- block length 19,38
- BLP subparameter 2,99
- buffer alignment block 71
- bypass label processing 99
  - component support of 119
  - specification of 2

## C

- cataloged data sets
  - explained 4
  - with no labels 97
  - with nonstandard labels 90
  - with standard labels 21
- CCW locations
  - for nonstandard labels 87
  - for volume label editor 113
- channel program area
  - for nonstandard labels 87
  - for volume label editor 113
- channel program field
  - for nonstandard labels 87
  - for volume label editor 113
- channel status word (CSW) 10
- character code, EBCDIC 8
- checking volume labels 103-115
- checkpoint/restart (see Restart routine)
- CLOSE macro instruction
  - for ANSI standard labels 59
  - function of 6
  - for IBM standard labels 27
  - for no labels 100
- Close routine
  - for ANSI standard labels 54,59
  - function of 6
  - for IBM standard labels 22,27
  - for no labels 100
  - for nonstandard labels 76,79
- closing an input data set
  - with ANSI standard labels 54
  - with IBM standard labels 22
- closing an output data set
  - with ANSI standard labels 59
  - with IBM standard labels 27
  - with no labels 100
- COBOL 119
- completing the data control block 3
- component support 119
- concatenated data sets
  - with ANSI standard labels 48,53
  - explained 5
  - with IBM standard labels 16,22
  - with no labels 98
  - with nonstandard labels 77
- connected data sets (see concatenated data sets)
- control characters
  - for ANSI standard label 50,71
  - for IBM standard labels 19,39
- conversion, data
  - ASCII to/from EBCDIC 1,43
  - BCD to/from EBCDIC 9-10,40
- creating a volume label
  - ANSI standard volume labels 54
  - IBM standard volume labels 23
- creation date
  - for ANSI standard labels 66
  - for IBM standard labels 34

## D

- data control block (DCB)
  - completion of 3
  - end-of-data routine
    - with ANSI standard labels 51-53
    - with IBM standard labels 21
    - with no labels 98
    - with nonstandard labels 76
  - user label (ANSI) exits 51-53,58
  - user label (IBM) exits 19,21,26-27
- data conversion 9-10,39,43
- data group index 4-5
- data management routines 6
- data recording characteristics 8-10
- data set header labels
  - ANSI standard labels 46
  - IBM standard labels 12,14
- data set identifier 33,117
- data set label 1 (HDR1, EOVI, EOF1)
  - ANSI standard
    - contents of 62-67
    - definition of 43-44
    - format of 62-67
    - processing of 47-58,62-67
  - IBM standard
    - contents of 31-41
    - definition of 11-14
    - format of 31-35
    - in other systems 117-118
    - processing of 15-27,31-41
- data set label 2 (HDR2, EOVI, EOF2)
  - ANSI standard
    - contents of 36-40
    - definition of 43-46
    - format of 68-71
    - processing of 47-58,68-71
  - IBM standard
    - contents of 36-40
    - definition of 11-14
    - format of 36-40
    - in other systems 117-118
    - processing of 15-27,31-41
- data set labels (*see* labels)
- data set name
  - in ANSI standard labels
    - checking before input processing 50
    - checking before output processing 56
    - format in HDR1 label 64
  - in IBM standard labels
    - checking before input processing 18
    - checking before output processing 25
    - format in HDR1 label 33
  - in other systems 117
- data set position 3
  - ANSI standard labeled tapes 70
  - IBM standard labeled tapes 39
- data set protection
  - of multiple volumes
    - with ANSI standard labels 53
    - with IBM standard labels 22
  - processing of
    - with ANSI standard labels 50,66-67
    - with IBM standard labeled tapes 18,35
  - specification of 3
  - with volume label editor 112-113
- data set sequence number
  - with ANSI standard labels 49,55
  - for concatenated data sets
    - with ANSI standard labels 53
    - with IBM standard labels 22,34
  - with IBM standard labels 17-18
  - for multiple volumes
    - with ANSI standard labels 53
    - with IBM standard labels 22,34
  - with no labels 97-98,100
  - in other systems 118
  - specification of 3
  - for SYSOUT data sets 28
- data set serial number
  - for ANSI standard labels 64
  - for IBM standard labels 33
- data sets
  - attributes of 3-6
  - cataloged 4
  - concatenated 5
    - with ANSI standard labels 53
    - with IBM standard labels 22
  - generation 4
  - passed 5
  - processing methods 6-7
  - used in multiple job steps 5
- DCB exit routine 4
- DCB macro instruction
  - completing the data control block 3
  - DEN parameter for 8-9
  - TRTCH parameter for 10
- DD statements
  - for concatenated data sets 5
  - DCB parameter for 3-4
  - DEN parameter for 8-9
  - DISP parameter for 7
  - LABEL parameter for 2
  - for multiple data sets 5-6
  - for multiple volumes 5-6
  - for passed data sets 5
  - TRTCH parameter for 10
  - VOLUME parameter for 5-6
- DDR option 93
- defaults
  - data set sequence number 3
  - label type 2
- deferred user trailer label processing
  - with ANSI standard labels 47,58
  - with IBM standard labels 15,22
  - with nonstandard labels 77
- DEN parameter 8-10
- density
  - ANSI standard labels 70
  - conflicts, how resolved 8-9
  - DEN parameter codes 8-9
  - IBM standard labels 38-39
  - volume label editor 103,106-108
- describing data sets
  - LABEL parameter 2
  - physical attributes 2-3
  - volume and unit information 2-3
- describing labels 1-2
- device name 113
- device type 4
- DISP parameter 7
- disposition of volumes

- positioning parameters explained 7
- dual density 8-9
- dummy header label
  - for ANSI standard labels 54
  - for IBM standard labels 23
- dynamic device reconfiguration (DDR)
  - option 93

## E

- editor, volume label
  - entry conditions 109
  - explained 103-115
  - flowcharts of 110-111
  - module names 107
- EMODVOL1 107, 114-115
- end-of-data routine
  - for ANSI standard labels 51-53
  - explained 6
  - for IBM standard labels 20-21
  - for no labels 98
  - for nonstandard labels 76-79
- end of data set
  - with ANSI standard labels 51-53
  - explained 6-7
  - with IBM standard labels 20,21
  - with no labels 98
  - with nonstandard labels 76-79
- end of reel 10
- end of volume
  - with ANSI standard labels 51-53,58
  - explained 6-7
  - with IBM standard labels 20-22
  - with no labels 98
  - with nonstandard labels 76-79
  - special conditions 26-27,58
- EOV editor routine 106-115
- EOV macro instruction 6
- EOV routine
  - for ANSI standard labels 51-53,58
  - for IBM standard labels 20-21,26-27
  - for no labels 98
  - for nonstandard labels 76-79
  - OPTCD=B 31,64
  - volume label editor routine 106-107
- EOV 1 label
  - (see data set label 1)
- EOV2 label
  - (see data set label 2)
- error conditions 106-107
- expiration date
  - with ANSI standard labels 50,56
  - with IBM standard labels 18,35
  - in LABEL parameter 3
  - with volume label editor 112-113
- external labels 121-122

## F

- FEOV macro instruction
  - with ANSI standard labels
    - input tape 47,51
    - output tape 58
  - with IBM standard labels
    - input tape 20-22
    - output tape 26-27
    - (see also EOV routine)
  - with no labels 98,101
  - with nonstandard labels 77,79
- file identifier 64
- file section number 65
- file sequence number 65
- first record, verification of
  - for ANSI standard labels 48,53
  - bypass label processing 99
  - for IBM standard labels 16,22
  - for no labels 97,98
  - for nonstandard labels 76-78,90-91
  - volume label editor 103-115
- format of
  - data set label 1
    - ANSI standard labels 62-67
    - IBM standard labels 31-36
  - data set label 2
    - ANSI standard labels 68-71
    - IBM standard labels 36-40
  - user labels
    - ANSI standard labels 72-73
    - IBM standard labels 40-41
- FORTRAN 119
- forward merge of the DCB 3

## G

- generation data groups
  - with ANSI standard labels 65
  - explained 4
  - with IBM standard labels 34
- generation numbers
  - in ANSI standard labels 65
  - explained 4
  - in IBM standard labels 34

## H

- HDR1 label
  - (see data set label 1)
- HDR2 label
  - (see data set label 2)
- header labels
  - ANSI identifiers for 43
  - definition of 1-2
  - IBM identifiers for 11

## I

- IBM standard labels
  - component support 119
  - definition 11
  - operating system support 11
  - other systems 117-118
  - overview of processing 15-16
  - specification of 1-2
  - types of 11
  - volume layouts for 12-13
- identifier field
  - in ANSI standard labels 43
  - in IBM standard labels 11
  - in user labels 72
  - in volume labels 28,60
  - identifying tape reels 121-122
- IECDSECT macro instruction
  - for nonstandard labels 83
  - for volume label editor 112
- IEFUCBOB macro instruction
  - for nonstandard labels 83
  - for volume label editor 112
- IEFXVAVR 91-92
- IEFXVNSL routine 91-92
- IEHINITT utility program
  - assigning volume serial numbers 28
  - creating ANSI standard labels 54
  - creating IBM standard labels 23
  - dummy ANSI HDR1 label 56
  - dummy IBM HDR1 label 23
- IGC0K05B 80
- IGC0408E 93
- IGG0190B 80
- IGG0190R 80
- IGG0200B 80
- IGG0550B 80
- IGG0550D 80
- IGG0550F 80
- IGG0550H 80
- IGG0550P 114
- input data sets
  - with ANSI standard labels 47-54
  - with IBM standard labels 16-18
  - with no labels 97-98
  - with nonstandard labels 76-77
  - specifying 6
- input header label routine 76-77
- input/output support routines 6-7
  - (*see also* Open routine; Close routine; EOV routine)
- input trailer label routine 77
- introduction to tape processing 1-10

## J

- JFCB (*see* job file control block)
- job control statements 2-3
- job file control block (JFCB)
  - data set sequence number (*see* positioning tapes)
  - header label information 19
  - merging control block information 19
- job and job step name
  - in ANSI data set label 71
  - in IBM data set label 39

## L

- label editor routines 103-115
  - label identifier
    - data set labels
      - ANSI 62,72
      - IBM 31,36
    - user labels
      - ANSI 72
      - IBM 40
    - volume label
      - ANSI 60
      - IBM 29
  - label number
    - data set labels
      - ANSI 64
      - IBM 33,38
    - user labels
      - ANSI 72
      - IBM 40
    - volume label
      - ANSI 60
      - IBM 30
  - LABEL parameter 2
  - label processing
    - ANSI standard labels 47
    - IBM standard labels 15-16
    - as used by other system components 119
  - label standard level 62
  - labels, external 121
  - labels, tape
    - ANSI standard 43-73
      - location on volume 44-45
    - definition of 1-2
    - external 121
    - IBM standard 11-41
      - location on volume 12-13
    - model 5
    - nonstandard 75-93
    - types of 1-2
  - LEAVE parameter
    - for ANSI standard labels 49
    - explained 7
    - for IBM standard labels 17
    - for no labels 101
  - linkage editor 119
  - load point 10
- ## M
- magnetic tape characteristics 8-10
  - magnetic tape labels
    - (*see* labels, tape)
  - merging control block data 3
  - model data set label 5
  - module names
    - for nonstandard label routines 80,90
    - for volume label editor 107,114
  - mount switch (SRTEDMCT)
    - ANSI standard labels 49,55
    - IBM standard labels 17,24
    - nonstandard labels
      - bit value for incorrect volume 77-78
      - use in label processing routines 84,88
    - volume label editor 114
  - mounting volumes in advance 7



- multiple data sets
  - with ANSI standard labels
    - end-of-volume conditions 58
    - for input data sets 47-50
    - for output data sets 55-56
    - restart procedure 59
    - volume organizations 45
  - DD statements for 5-6
  - with IBM standard labels
    - end-of-volume conditions 27
    - for input data sets 17-18
    - for output data sets 24-25
    - restart procedure 28
    - volume organizations 13
  - with no labels 98-100
  - with nonstandard labels 75,76
- multiple volumes
  - with ANSI standard labels
    - checking volume labels 53
    - creating a volume label 54
    - switching volumes 52-53
    - volume organization 44,45
  - DD statements for 5-6
  - with IBM standard labels
    - checking volume labels 22
    - creating a volume label 23
    - from other systems 118
    - switching volumes 21
    - volume organization 12-13
    - with no labels 98-100
    - with nonstandard labels 75

## N

- named generation group 4
- new volume labels
  - ANSI standard volume labels 55
  - IBM standard volume labels 22-23
- nine-track tapes 8-9
- NL subparameter 2,95
- nonstandard label processing routines
  - for AVR91
  - in control program 90
  - flowcharts of 83-90
  - types of 76-79
  - writing of 79-80
- nonstandard labels
  - component support 119
  - in other systems 118
  - processing of 76-79
  - specification of 2
- NRZI mode 8-9,39
- NSL subparameter 2,76
- NSLREPOS routine 93

## O

- OMODVOL1 107,114
- OPEN macro instruction
  - overriding INOUT and OUTIN parameters 3
  - processing for a tape data set 6-7
- Open routine
  - for ANSI standard labels 48-49,55-56
  - function of 6
  - for IBM standard labels 16-18,23
  - for no labels 97-101

- for nonstandard labels 76-78
- volume label editor routine 106-108
- opening an input data set
  - with ANSI standard labels 48-49
  - with IBM standard labels 16
  - with no labels 97
- opening an output data set
  - ANSI standard labels specified 55
  - IBM standard labels specified 23
  - no labels specified 99
- other systems 43,117-118
- output data sets
  - with ANSI standard labels 55
  - with IBM standard labels 23-25
  - with no labels 99-101
  - with nonstandard labels 78
- output header label routine 78
- output trailer label routine 79
- owner identification
  - in ANSI standard volume label 62
  - in IBM standard volume label 31

## P

- parity 8-10
- passed data sets
  - when to define attributes 5-6
- password protection
  - (see data set protection)
- phase encoded (PE) 8,39
- PL/I 119
- positioning tapes
  - with ANSI standard labels 49,55-56
  - with IBM standard labels 17-18,23-24
  - with no labels 97-98,100
  - with nonstandard labels 76-78
  - using OPEN and CLOSE parameters 7
- processing routines 6
- protection (see data set protection)

## R

- RDBACK parameter
  - with ANSI standard labels 51
  - explained 6
  - with IBM standard labels 17,19
  - with no labels 98
  - with nonstandard labels 87
  - restriction with data conversion 9
- read backward
  - with ANSI standard labels 51
  - with IBM standard labels 19
  - with no labels 98
  - with nonstandard labels 75
  - restriction with data conversion 9
- record format
  - with ANSI standard labels 51,68-69
  - with IBM standard labels 19,38
- record length
  - with ANSI standard labels 51,70
  - with IBM standard labels 19,38
- reel label 121
- reflective strip
  - function of 10
  - writing beyond
    - ANSI standard labels 58-59

- IBM standard labels 27
- relative generation number 4-5
- REREAD parameter 7
- Restart routine
  - with ANSI standard labels 59
  - function of 7
  - with IBM standard labels 28
  - with no labels 101
  - with nonstandard labels 78,89
  - work areas 123-127
- retention period
  - (see expiration date)
- reverse merge of DCB 3
- reverse read (see read backward)
- RPG 119

## S

- scratch tape
  - with ANSI standard labels 55,61
  - with IBM standard labels 24,30
  - with no labels 100
- security status code
  - in ANSI data set label 1 66
  - in IBM data set label 1 35
- security protection
  - (see data set protection)
- sense byte 10
- set identifier 64
- seven-track feature
  - with code translation 9,10
  - with data conversion feature 9,10
  - density requirements 9-10
  - lack of ANSI support for 9
  - models allowing 9
- SL subparameter 2,11
- sort/merge 119
- SRTEDMCT
  - (see mount switch)
- standard labels
  - (see ANSI or IBM standard labels)
- standard volume label
  - (see volume label)
- SUL subparameter 2,11
- SVC library
  - with nonstandard labels 75,90
  - with volume label editor 115
- system code
  - with ANSI standard labels 67
  - with IBM standard labels 36
- system generation
  - for bypass label processing 99
  - for nonstandard label routines 90-92
  - for seven-track density 9-10
- system input tapes (SYSIN) 10
- system output tapes (SYSOUT) 10,24

## T

- tape characteristics 8-10
- tape format 2
  - (see also format of data set label 1)
- tape labels
  - (see labels, tape)
- tapemarks
  - with ANSI standard labels 47
  - defined 10
  - with IBM standard labels 14
  - how processed 10
  - with nonstandard labels 79
- tape recording technique 10
- tape reposition routine 93
  - tape units 8-10
- track densities 8-10
- trailer labels
  - ANSI standard
    - deferred processing of 50
    - definition of 1
    - format of 66,72
  - IBM standard
    - deferred processing of 15
    - definition of 1,14
    - format of 32,37
- translation 9
- TRTCH parameter 10

## U

- unit check 103
- unit exception bit
  - when set 10
- unlabeled tapes
  - component supports of 119
  - other systems 118
  - processing of 95-101
  - replaced by labeled tape 99
  - specification of 2
  - volume layouts for 96
- user identification
  - (see owner identification)
- user labels
  - ANSI standard
    - deferred processing of 47-48
    - format of 72-73
    - location on volume 44-45
  - IBM standard
    - component support of 119
    - deferred processing of 15
    - format of 40-41
    - location on volume 12-13
    - processing requirements 14
- utilities 119
  - (see also IEHINITT utility program)

## V

- variable length records 9-10
- version numbers
  - with ANSI standard labels 65-66
  - explained 4
  - with IBM standard labels 34
- volume disposition 7
- volume label
  - ANSI standard
    - creation of 54
    - definition of 43
    - format of 61
    - processing of 47
    - verification of 103-115
  - IBM standard
    - creation of 22-23
    - definition of 11
    - format of 29
    - processing of 15-18
    - verification of 103-115
  - user 73
- volume label editor routines 103-115
- volume organizations
  - ANSI standard labels 44-45
  - basic layouts 1-2
  - IBM standard labels 12-13
  - nonstandard labels 75
  - unlabeled tapes 96
- VOLUME parameter 5-6
- volume positioning to a data set
  - with ANSI standard labels 49,55-56
  - with IBM standard labels 17,24
  - with unlabeled tapes 97,100
- volume protection and accessibility
  - in ANSI standard volume label 61-62
- volume sequence number
  - with ANSI standard labels
    - explained 64
    - restart procedure 59
    - volume switching 52
  - with IBM standard labels
    - location in header label 33-34
    - volume switching 21
- volume serial number
  - with ANSI standard labels
    - explained 64
    - input processing of 49
    - output processing of 55
    - restart procedure 59
    - switching volumes 52
  - assigning by system 100
  - assigned in JCL statements 29
  - assigned through IEHINITT 29
  - in external labels 121
  - with IBM standard labels
    - explained 17
    - input processing 17
    - output processing 24
    - switching volumes 21
  - with no labels 98
  - with nonstandard labels 77,91
  - volume label editor 106-107,113-114
- volume switching
  - with ANSI standard labels 52-53,58-59
  - with IBM standard labels 21,27

- with no labels 98
  - with nonstandard labels 76-77
  - provisions in EOV routines 6
- volume verification  
(*see also* volume serial number)
- done by system 103-106
  - done by user 110-115
- volumes created by other systems 117
- VOL1 label  
(*see* volume label)
- VTOC pointer 30

## W

- work areas
  - for AVR nonstandard label routines 91-92
  - for nonstandard label routines 79-82
  - for restart routines 123-127
  - for volume label editor routines 106-107,1

## 123

- 7-track tapes 9-10
- 9-track tapes 8-9



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

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

OS Tape Labels  
GC28-6680-5

**Reader's  
Comment  
Form**

Your comments about this publication will help us to improve it for you. Comment in the space below, giving specific page and paragraph references whenever possible. All comments become the property of IBM.

Please do not use this form to ask technical questions about IBM systems and programs or to request copies of publications. Rather, direct such questions or requests to your local IBM representative.

If you would like a reply, please provide your name, job title, and business address (including ZIP code).

**Fold on two lines, staple, and mail.** No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Fold and Staple

██████████  
First Class Permit  
Number 2078  
San Jose, California

---

**Business Reply Mail**

No postage necessary if mailed in the U.S.A.

---

Postage will be paid by:

**IBM Corporation  
Programming Center – Publishing  
Department D58  
Monterey and Cottle Roads  
San Jose, California 95193**



Fold and Staple

OS Tape Labels (File No. S360/370-30) Printed in U.S.A. GC28-6680-5



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)