File No. S360-30
Form C28-6680-1

OS

IBM Systems Reference Library

IBM System/360 Operating System
Tape Labels

This publication describes how the IBM System/360 Operating System processes magnetic tape volumes with 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, multi-data set, and multi-volume conditions.
- Tape characteristics.
- Nonstandard label processing routines and volume label editor routines.
- Component considerations.
- Using tape volumes created by other systems.
- External labels.

Preface

This publication provides programmers and operations analysts with information on how the IBM System/360 Operating System processes magnetic tape labels. The three main sections of the publication describe the specifications and processing of tapes with standard labels, tapes with nonstandard labels, and unlabeled tapes. You need only read 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 component considerations, 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:

IBM System/360 Operating System:

Concepts and Facilities, Form C28-6535

Job Control Language, Form C28-6539

Supervisor and Data Management Services, Form C28-6646

The following publications are referred to in the text:

IBM System/360 Operating System:

System Programmer's Guide, Form C28-6550

System Generation, Form C28-6554

Utilities, Form C28-6586

System Control Blocks, Form C28-6628

Second Edition (November 1968)

This is a major revision of, and obsoletes, C28-6680-0 and Technical Newsletter N28-2322. This edition includes information on the checkpoint/restart facilities. It describes how the Restart routine repositions magnetic tape volumes, and provides the information for writing a restart nonstandard label processing routine. This edition also includes information on: operating system support of user labels; the LABEL subparameters IN and OUT; using special characters in the volume serial number and data set identifier; and an extension of the standard data set label 2. Changes to the text, and small changes to illustrations, are indicated by a vertical line to the left of the change; changed or added illustrations are denoted by the symbol ● to the left of the caption.

This edition applies to release 17 of the IBM System/360 Operating System, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM SRL Newsletter, Form N20-0360, for the editions that are applicable and current.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form for readers' comments appears at the back of this publication. It may be mailed directly to IBM. Address any additional comments concerning this publication to the IBM Corporation, Programming Publications, Department 232, San Jose, California 95114.

Contents

Illustrations

Figures

Tables

Section I: Introduction to Tape Processing

Labels are used to identify magnetic tape volumes and the data sets they contain. With the IBM System/360 Operating System, you can process tape volumes with 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 optionally include standard user labels.

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 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.

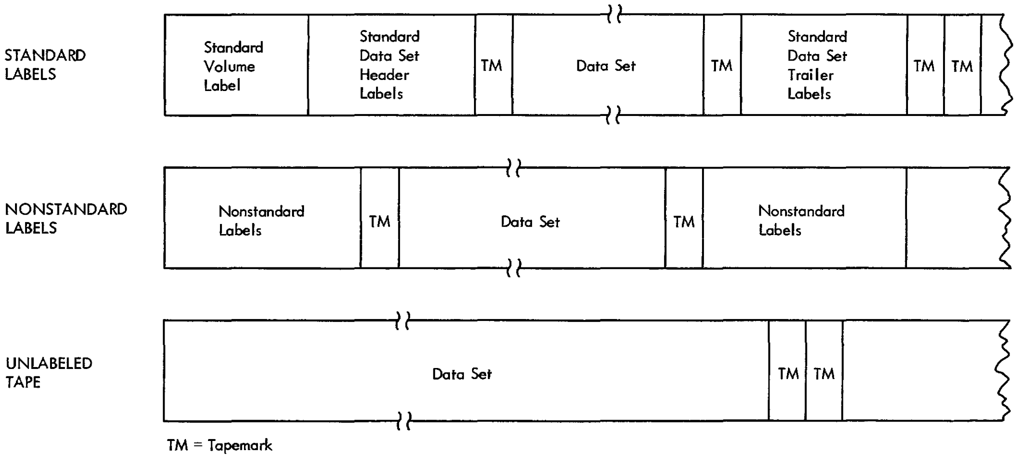


Figure 1. Basic Tape Layouts

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 -- the data set has standard labels.

SUL -- the data set has both standard and user labels.

NSL -- the data set has nonstandard labels.

NL -- the data set has no labels.

BLP -- bypass label processing. (BLP may be used when a data set having no labels is to be written. BLP is treated the same as NL, except that the system does not check for an existing volume label.)

If you do not specify the label type, the operating system assumes that the data set has 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), an1 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.

The data control block is completed at execution time when it is opened. Figure 2 illustrates the process and sequence of filling in the data control block from the various sources. Steps 3 and 7 are bypassed for tapes with nonstandard labels or no labels.

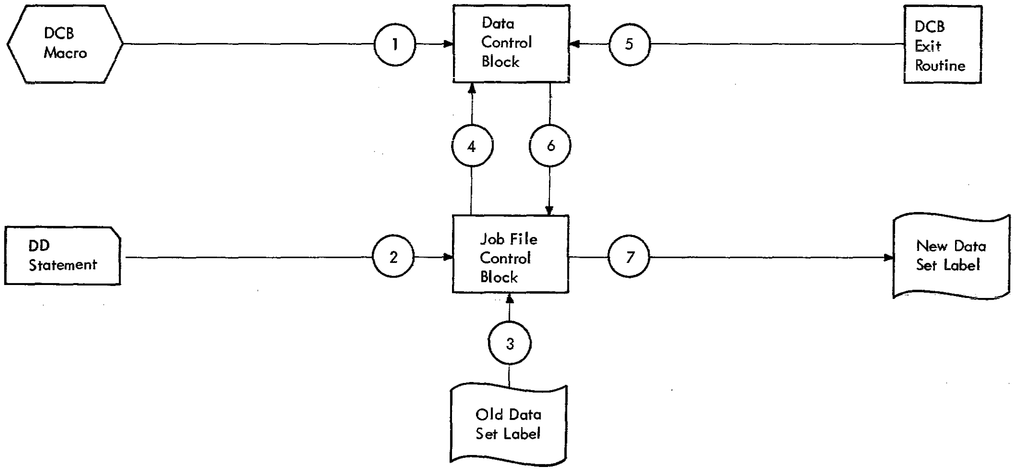


Figure 2. Sources and Sequence for Completing the Data Control Block

Forward Merge: 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: 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.

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 the Utilities and Job Control Language publications.

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 multi-volume 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 the Supervisor and Data Management Services publication.

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 actually 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 different 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 it occupies. Therefore, you must not use the catalog to retrieve an unlabeled data set 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, EOV, 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 EOV routine is entered when a tapemark is read, when the end of reel (reflective strip) is encountered, or when the processing program issues an FEOV (force end of volume) macro instruction.[1] The EOV 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 EOV 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, EOV 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 standard labels, nonstandard labels, or no labels.

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.

----

[1] If you use the EXCP (execute channel program) technique, your processing program must issue an EOV macro instruction to give control to the EOV routine after your program recognizes a tapemark or end of reel.

AVR checks the tape label whenever a device becomes ready, 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 is implied 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 (multi-volume 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 implied by the DISP parameter of the applicable DD statement.

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 upon 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 Magnetic Tape Units. The error conditions that can result from conflicting tape characteristics are explained in Section V of this publication.

The term "IBM 2400 Magnetic Tape Units" refers to the IBM 2401 and IBM 2402 Models 1-6, and IBM 2420 Model 7 Magnetic Tape Units; and the IBM 2403 and IBM 2415 Models 1-6, and IBM 2404 Models 1-3, 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 read and write data in nine tracks. The operating system supports nine-track tape in densities of 800 bpi (bytes per inch -- actually, bits per inch per track) or 1600 bpi. All Models 1-3 read and write a density of 800 bpi in the NRZI (non-return-to-zero IBM) 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 Models 4-7 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 nine-bit character). Tape characters are always recorded in the EBCDIC (extended binary coded decimal interchange) character code.

You may specify a density with the DEN parameter of the DD statement or DCB macro instruction (Table 1 shows the DEN parameter codes). However, since these tape units can only read or write in a single density, the operating system ignores any user specifications. For input, the tape is read in the density required by the tape unit. For output, the tape is written in the density required by the tape unit; the density of the existing data on the tape is ignored.

Table 1. DEN Parameter Codes for Specifying Tape Density

Code	Recording Density Model 2400	
	Seven-Track	Nine-Track
0	200 bpi	--
1	556 bpi	--
2	800 bpi	800 bpi
3	--	1600 bpi

Default value for seven-track is 200 bpi.

Default value for nine-track is 800 bpi.

Default value for dual density is 1600 bpi.

Nine-Track Dual-Density Feature

The operating system supports a special feature called dual density. This feature can be installed on all Models 4-6 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 (Table 1 shows the DEN parameter codes). If not specified, the default value is 1600 bpi.

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. If you request an unlabeled tape, you can specify either density. If you request a tape having standard or nonstandard labels, the density you specify must be the same as the density of the existing tape label. If a conflict occurs, the IBM-supplied editor routine requests the operator to demount the current tape and to mount a tape that conforms to the specified density. (Your installation can provide a volume label editor routine that rewrites the existing label in the density you specify.)

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

Seven-Track Tapes

The seven-track feature can be installed on all Models 1-3 to allow them to read and write seven-track tapes instead of nine-track tapes. This special feature consists of a seven-track read/write head (replacing the nine-track head) and control unit changes, including a translator. Data can be read or written in densities of 200 bpi, 556 bpi, or 800 bpi, with odd or even parity.

The translator writes eight-bit EBCDIC main storage characters as six-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 main storage characters are written on the tape; and during reading, the two high-order bits of the main storage characters are set to zeros.

The data conversion feature can also be installed with the seven-track feature. The data conversion feature makes it possible to write binary data on seven-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 seven-track feature. You specify the density with the DEN parameter of the DD statement or DCB macro instruction (Table 1 shows the DEN parameter codes). If not specified, the default value is 200 bpi.

Note: If you use densities other than 200 bpi for seven-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. (With the primary control program, you should not use the DEN parameter to describe a system output tape. If you specify a density with the DEN parameter, it overrides the density established at system generation and may produce a multi-density system output tape. Your installation's programs may not be able to print the data from a multi-density tape.)

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 (entire parameter is omitted) -- odd parity with no translation or conversion (same as nine-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. Standard labels on seven-track tape are always written in BCD, translate on, and even parity, regardless of the value of the TRTCH parameter.

Nonstandard labels on seven-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 hardware 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 tapemark 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.

Section II: 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 standard label processing. If you specify SUL, data management processes both standard and standard user labels.

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

SL
SUL

Label Definitions and Organization

Standard labels are 80-character records which are recorded in EBCDIC and odd parity on nine-track tape; or in BCD and even parity, translate on, on seven-track tape. The first four characters are always used to identify the labels. These identifiers are:

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 to UHL8	-- user header labels
UTL1 to 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 UHL1-8 or UTL1-8).

Figures 3 and 4 show the positions of the labels on various tape volume organizations. A tape with standard labels must contain a volume label and data set labels. User labels are optional.

Volume Label

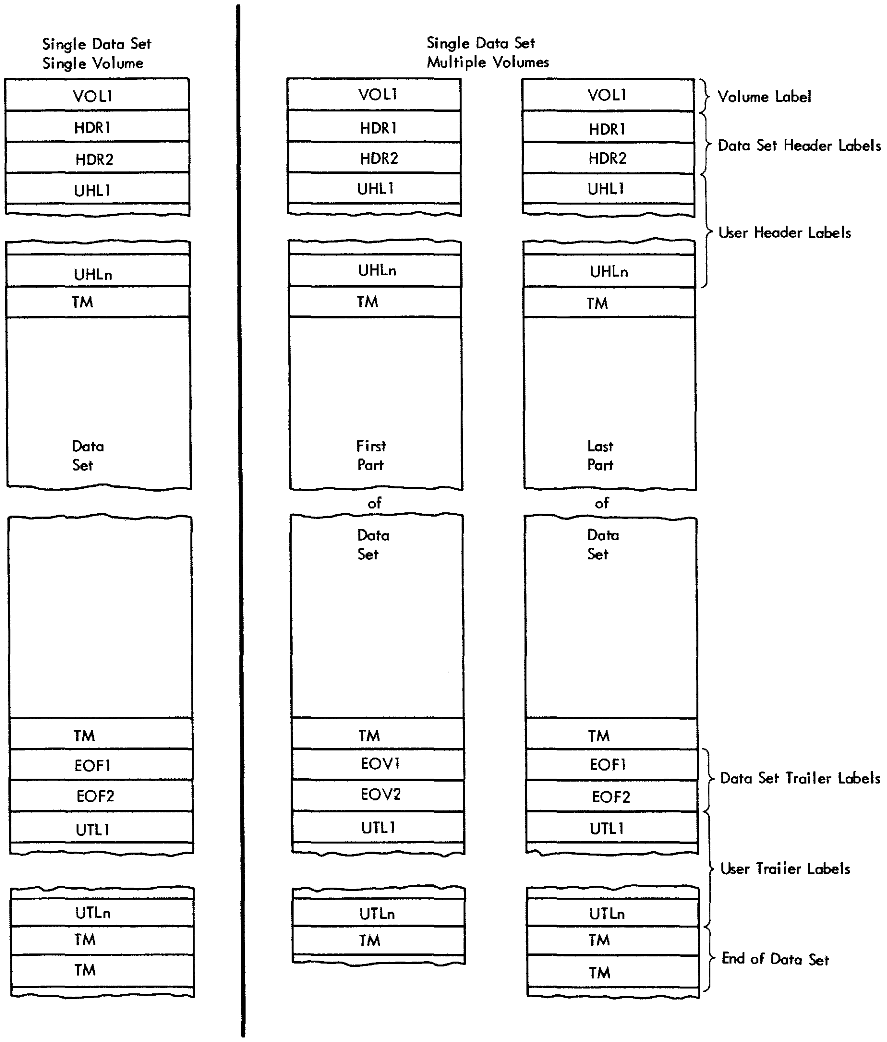
The standard volume label (VOL1) appears at the beginning of each tape. It identifies the volume and its owner, and is used to verify that the correct volume is mounted.

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

Data Set Header Labels

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

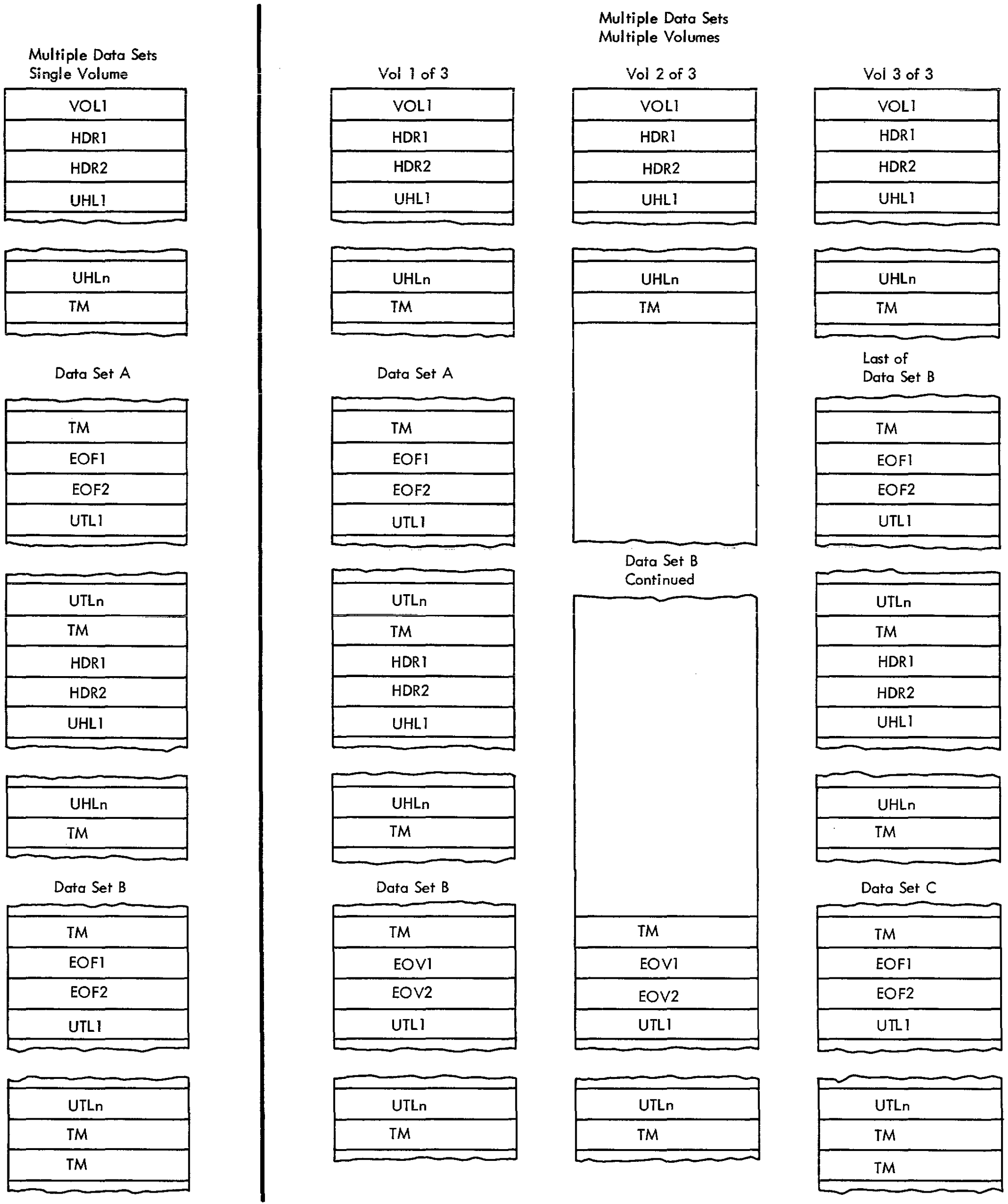
These labels are created automatically by data management each time a data set is recorded on tape.



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 EOV 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 3. Volume Organizations with 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 EOV 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 4. Volume Organizations with 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, i.e., a routine that processes input user header labels and a routine that creates output user header labels.) The exit list (EXLST) is described in the publication IBM System/360 Operating System: Supervisor and Data Management Services.

Data Set Trailer Labels

The data set trailer label group consists of standard data set label 1 (EOV1 or EOF1) and standard data set label 2 (EOV2 or EOF2). These labels duplicate the 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 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, i.e., a routine that processes input user trailer labels and a routine that creates output user trailer labels.) The exit list (EXLST) is described in the publication IBM System/360 Operating System: Supervisor and Data Management Services.

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 standard labels, data management writes the necessary tapemarks.

Label Processing

Label processing is handled by the I/O support routines of data management (Open, EOV, 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 Table 2. The figure shows the specific labels that are processed for each function, and which routines perform the functions. The summary in Table 2 is the general basis for the discussions of label processing that follow in this section.

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. 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.

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

•Table 2. Standard Label Processing by Data Management Routines

Processing	Volume Label	Header Labels[1]			Trailer Labels[1]		
	VOL1	HDR1	HDR2	UHL1-8	EOF1 or EOV1	EOF2 or EOV2	UTL1-8
First or Only Volume:[2]							
Checks labels on input tape.	Open	Open	Open	Open	EOV	bypassed	EOV
Checks existing labels on output tape before overwriting.	Open	Open	not checked	not checked	not checked	not checked	Open[5]
Writes new labels on output tape.	user[4]	Open	Open	Open	Close or EOV	Close or EOV	Close or EOV
Second or Subsequent Volumes[3]:							
Checks labels on input tape.	EOV	EOV	bypassed	EOV	EOV	bypassed	EOV
Checks existing labels on output tape before overwriting.	EOV	EOV	not checked	not checked	not checked	not checked	not checked
Writes new labels on output tape.	user[4]	EOV	EOV	EOV	Close or EOV	Close or EOV	Close or EOV

Notes:

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.

3. Includes the first volume of concatenated data sets with "like" characteristics.

4. Data management never writes or alters the volume label (VOL1). User creates the label with the IEHINITT utility program or a user program.

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).

Opening an Input Data Set

If standard labels are specified, the first record on the input tape must be a standard volume label (VOL1). At the time the data set is opened, data management checks the first record on the tape to ensure that the record is 80 bytes in length and contains the identifier VOL1 in the first four bytes. The various error conditions that can occur during verification of the first record are explained in Section V of 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 specified by the user. 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 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 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 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 zero.

2. When a data set is opened, the Open routine sets the data set sequence number in the UCB to one. 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 sets/multiple volumes aggregate.)

3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If equal, the tape is already positioned at the requested data set. If 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 to key in the correct password.

The password from the operator 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, 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.

The System Programmer's Guide publication 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 zero (EBCDIC or BCD). This zero 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 zero 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 (seven-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 forward spaces past the tapemark immediately after processing the HDR2 label. When this action occurs, 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 presented in reverse sequence.

Multi-volume 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

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 four 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 zero block count shown in the HDR 1 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 as hexadecimal '0B'. The EOV routine passes the following information to the exit routine:

Register 0 -- contains the block count shown in the label.
Register 1 -- contains 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 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 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 (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 multi-volume 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 facility. 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.

SL
SUL

For input, the volume serial numbers specified by the user are the basis for determining whether a volume switch is required. The labels do not assist in this determination. Data management does not care whether the trailer labels are identified as EOV or EOF. This allows you to create different volumes of a data set at different times, with EOF labels in each volume. This also allows you to terminate processing of a multi-volume data set after processing a volume containing EOV labels.

To determine whether or not 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 zero.

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 EOV routine issues a mount message to the operator.

When multiple tape units are being used, the EOV routine also cnecks 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 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 is not verified at this time.

Checking the Next Volume

When volume switching is performed for multiple volume input, the EOV 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 multi-volume data set or is a concatenated data set with "like" characteristics.

- Multi-volume data set: The data set sequence number is irrelevant for the second and subsequent volumes of a multi-volume data set. The EOV 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, except that protection (password) is not verified. If the data set name is the same, data management considers that the protection carries forward from the first volume. If the data set name is not the same, or the password was not verified for the first volume, processing is abnormally terminated.

- Concatenated data sets: The EOV routine handles concatenated data sets with "like" characteristics. Such data sets are not necessarily the first on the volume, so the EOV routine 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 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 EOV 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 EOV routine before the 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 FEOV macro instruction is executed, processing of trailer labels is omitted.

Creating a Volume Label

Data management never writes or alters the standard volume label (VOL1). The VOL1 label is usually written once, 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 volume labels. IEHINITT initializes the tape by writing, in the following order:

1. A volume label (VOL1) with user-specified volume serial number and owner identification. 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 the Utilities publication.

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 Section V) 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.

Opening an Output Data Set

If standard labels are specified, the first existing record on the output tape must be a 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 80 bytes in length and contains the identifier VOL1 in the first four bytes. The various error conditions that can occur during verification of the first record are explained in Section V of 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 will compare it to 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 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, the tape will be spaced through and removed from its reel.

- 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 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 following 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.

If the data set name is correct, data management requests the operator to key in the required password. The password from the operator 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, 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.

SL
SUL

The *System Programmer's Guide* publication 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 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 later in this section. The process of updating the JFCB is explained in the Introduction to the 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).

End of Volume on Output

Data management's EOV routine automatically switches volumes when an end-of-volume condition occurs (reflective strip or FEOV macro instruction). 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 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 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 EOV1 and EOV2 (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 EOV 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 multi-volume data set, the data set sequence number is irrelevant for the second and subsequent volumes. The EOV routine assumes that the data set continues at the beginning of the new volume.

Special End-of-Volume Conditions

When an end-of-volume condition (reflective strip) occurs during the writing of data, the EOV routine writes the trailer labels as described above. If the reflective strip is encountered while writing the trailer labels, the EOV 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 the CLOSE macro instruction is executed, data management checks a switch in the data control block to determine if the last operation was a non-write operation (read or point). If a non-write operation was performed last, no trailer labels are written. Otherwise, the Close routine creates the trailer labels for the data set.

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 zero 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 is different from the data set used originally. The system writes data set header labels (PCP tape only) and job separators at the beginning of the SYSOUT data set used during restart.

System Input and Output Tapes (PCP)

The primary control program (PCP) of the operating system provides special processing of tape volumes that are used for system input (SYSIN) or system output (SYSOUT). This special processing does not apply to priority schedulers.

Tapes containing the input stream (SYSIN) can be labeled or unlabeled, with no effect on processing. Data management bypasses all label processing on such tapes.

A system output tape (SYSOUT) is normally a multi-data set volume with standard labels. Each data set on the volume is named SYSOUT. Each HDR1 label contains the characters SYSOUT in the six high-order bytes of the data set name field, so that the first ten bytes of the label read HDR1SYSOUT. The EOV1 or EOF1 trailer labels are identified the same way. Data management automatically maintains and updates the data set sequence count so that the SYSOUT data sets are written in sequence on the tape (a data set sequence number specified in the LABEL parameter is ignored).

Format of the Standard Volume Label (VOL1)

The 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 a standard labeled tape. It is recorded in EBCDIC on nine-track tape units, or in BCD on seven-track tape units.

Figure 5 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 the following system control blocks:

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

Location of areas within these control blocks can be found in the System Control Blocks publication.

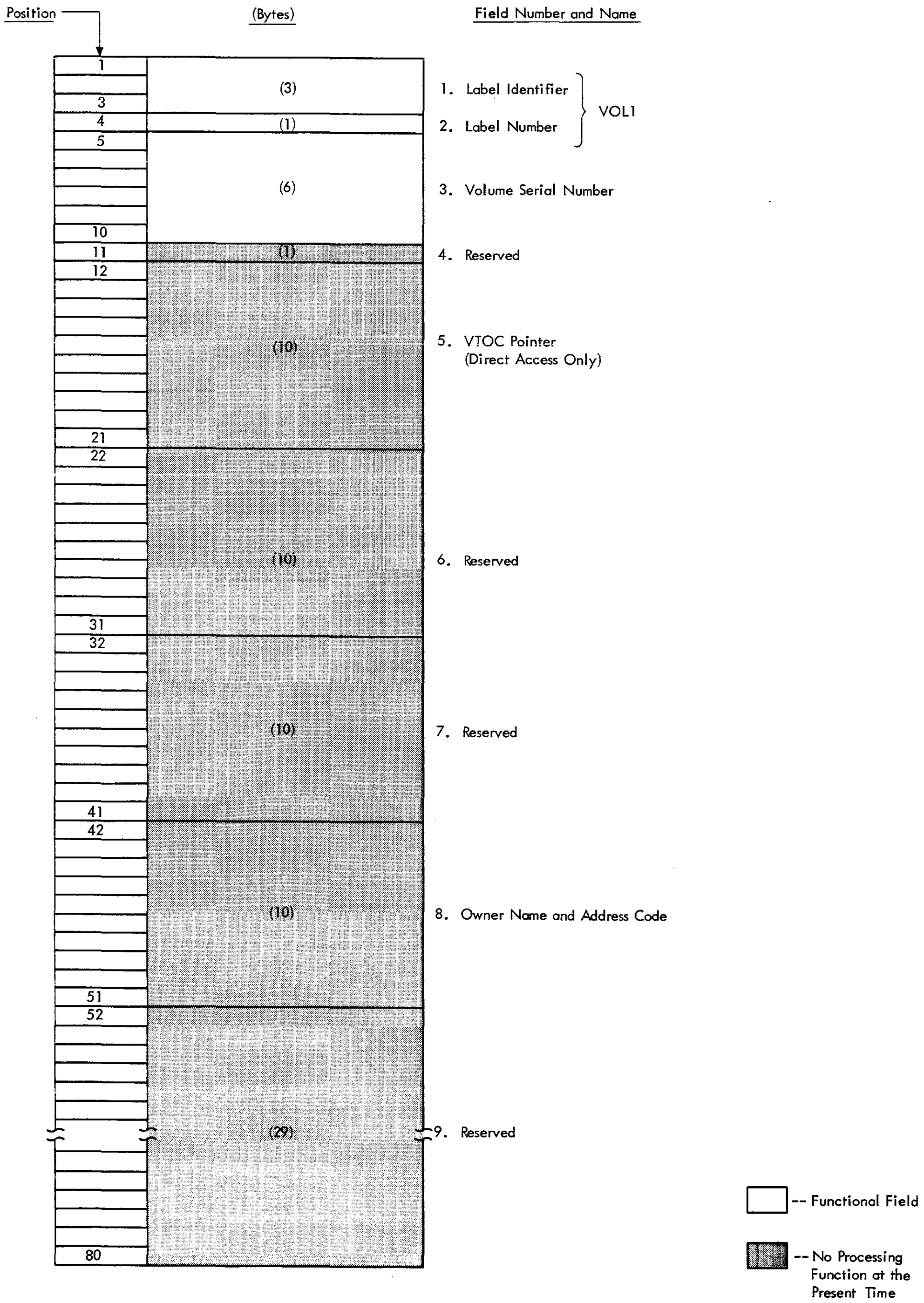


Figure 5. Format of Standard Volume Label

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.

     Volume labels are not created or altered by data management. The labels are written by the IEHINITT utility program or a user's program.

2. Label Number (1 byte)

   - CONTENTS: The relative position of this label within a set of labels of the same type; always a 1 for the 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 to the volume when it enters the system. 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 alphameric, national, or special characters. (If the code includes special characters, other than a hyphen, it must be enclosed in apostrophes when the volume is requested on a DD statement.)

   - 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 less than 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, installation, department, person, etc., to which the volume belongs. Any code or name is acceptable.

   - PROCESSING: Not used or verified. The IEHINITT utility program writes the code specified by the user. If the code is less than ten bytes, 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 Standard Data Set Label 1 (HDR1/EOV1/EOF1)

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 nine-track tape units, or in BCD on seven-track tape units.

Figure 6 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 the System Control Blocks publication.

Position	(Bytes)	Field Number and Name	
1 3	(3)	1. Label Identifier	HDR1/EOV1/EOF1
4	(1)	2. Label Number	
5 21	(17)	3. Data Set Identifier	
22 27	▨ (6)	4. Data Set Serial Number	
28 31	▨ (4)	5. Volume Sequence Number	
32 35	(4)	6. Data Set Sequence Number	
36 39	▨ (4)	7. Generation Number	
40 41	▨ (2)	8. Version Number	
42 47	▨ (6)	9. Creation Date	
48 53	(6)	10. Expiration Date	
54	(1)	11. Data Set Security	
55 60	(6)	12. Block Count	
61 73	▨ (13)	13. System Code	
74 80	▨ (7)	14. Reserved	

□ -- Functional Field

▨ -- No Processing Function at the Present Time

Figure 6. Format of Standard Data Set Label 1

1. Label Identifier (3 bytes)

   - CONTENTS: Three characters that identify the label, as follows:

     HDR -- for a header label (at the beginning of a data set).

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

     EOF -- for a trailer label (at the end of a data set).

   - PROCESSING: This field is read to verify that the record is a standard data set label. Data management accepts either EOV or EOF on the trailer label; the identifier is not used to determine if a volume switch is necessary. When creating trailer labels, the EOV routine writes EOV, 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; 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 right-most 17 bytes of the data set name (includes GnnnVnn if 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 *Job Control Language* publication 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 if 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.

4. Data Set Serial Number (6 bytes)

   - CONTENTS: The volume serial number of the tape volume containing the data set. For multi-volume 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, national, or special characters.

- 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 multi-volume 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 one, 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 one.

     When creating labels, the Open and Close routines obtain the user-specified data set sequence number from the JFCB (a zero is changed to one). The EOV 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 0000 to 9999 indicating the absolute generation number (the first generation is recorded as 0000). 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 if 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 if 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 byyddd where:

  b = blank

  yy = year (00-99)

  ddd = day (001-366)

- PROCESSING: Not used or verified. When creating labels, the date is obtained from the JFCB. This is the date when the job entered the system (set by the operator at IPL), and not necessarily the current date at the time 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 date is shown in the format byyddd 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 higher, a message informs the operator and gives him the option to use the tape or mount 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 the user 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, as follows:

  0 -- No security protection.

  1 -- Security protection. Additional identification of the data set is required before it can be processed.

- PROCESSING: For input, data management inspects this field on a single volume data set, on each concatenated data set, and on the first volume of a multi-volume data set. This field is not inspected on the second and subsequent volumes of a multi-volume 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 from the operator and the data set name in Field 3 of this label.

When creating 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: For input, the zero count in the header label is recorded in the DCB. The DCB count is incremented as the data set is read. The final DCB count is compared to 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 will compare equally to the count in the header label.

  When creating 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 Standard Data Set Label 2 (HDR2/EOV2/EOF2)

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 nine-track tape units, or in BCD on seven-track tape units.

Figure 7 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:

Position	(Bytes)	Field Number and Name
1–3	(3)	1. Label Identifier } HDR2/EOV2/EOF2
4	(1)	2. Label Number } HDR2/EOV2/EOF2
5	(1)	3. Record Format
6–10	(5)	4. Block Length
11–15	(5)	5. Record Length
16	(1)	6. Tape Density
17	(1)	7. Data Set Position
18–34	▨ (17)	8. Job/Job Step Identification
35–36	(2)	9. Tape Recording Technique
37	(1)	10. Printer Control Characters
38	▨ (1)	11. Reserved
39	(1)	12. Block Attribute
40–80	▨ (41)	13. Reserved

□ - Functional Field

▨ - No Processing Function at the Present Time

●Figure 7. Format of Standard Data Set Label 2

- 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 the System Control Blocks publication.

1. Label Identifier (3 bytes)

   - CONTENTS: Three characters that identify the label, as follows:

     HDR -- for a header label (at the beginning of a data set).

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

     EOF -- for a trailer label (at the end of a data set).

   - PROCESSING: This field is read to verify that the record is a standard data set label. Data management accepts either EOV or EOF on a trailer label; the identifier is not used to determine if a volume switch is necessary. When creating 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; always a 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
     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 32764 which 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 four-byte length indicator in the blocks).

     Format U -- Maximum block length.

- PROCESSING: The number in the label is converted to binary and merged to 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.

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 four-byte length indicator in the records).

  Format U -- Zeros.

- PROCESSING: The number in the label is converted to binary and merged to 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:

Seven-track tape	Nine-track tape
0 -- 200 bpi	---
1 -- 556 bpi	---
2 -- 800 bpi	2 -- 800 bpi
---	3 -- 1600 bpi

- PROCESSING: Not used or verified. When creating labels, the information for this field is obtained from the JFCB.

7. Data Set Position (1 byte)

- CONTENTS: A code indicating a volume switch, 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 a 0 in this field, and the EOV 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 -- and writes a 0 if the numbers are equal; a 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 eight bytes contain the name of the job; the ninth byte is a slash (/); and the final eight bytes contain the name of the job step.

- PROCESSING: Not used or verified. When creating 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 (seven-track tape only):

  Tb̸ -- odd parity with translation
  Cb̸ -- odd parity with conversion
  Eb̸ -- even parity with no translation
  ET -- even parity with translation
  b̸b̸ -- odd parity with no translation or conversion

  This field is recorded as blanks for nine-track tape. The only technique available for nine-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 to 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.

10. Printer Control Characters (1 byte)

- CONTENTS: A code or blank indicating whether a control character set was used to create the data set, and the type of control characters used:

  A -- Contains USASI control characters.
  M -- Contains machine control characters.
  b -- Does not contain control characters.

- PROCESSING: The specification in the label is converted to a bit code and merged to 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.

SL
SUL

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 or blank indicating the block attribute used to create the data set:

  B -- blocked records
  S -- spanned records
  R -- blocked and spanned records
  b -- not blocked and not spanned records

- PROCESSING: The specification in the label is converted to a bit code and merged to 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 (UHL1-8, UTL1-8)

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

Figure 8 shows the format of user labels. The format is used for user header labels (UHL1-8) and user trailer labels (UTL1-8). 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.

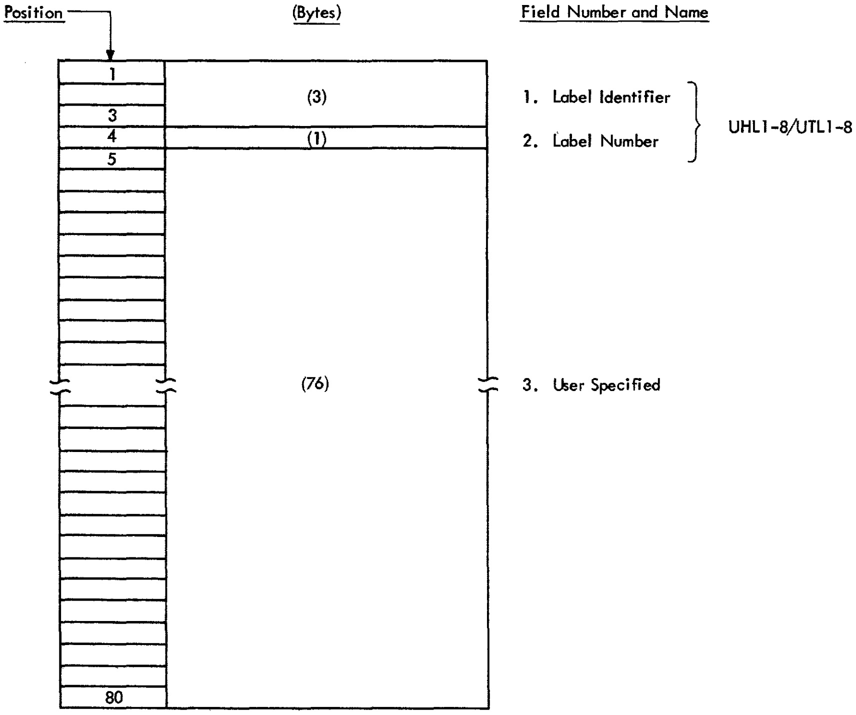


Figure 8. Format of User Label

1. Label Identifier (3 bytes)

   - CONTENTS: Three characters that identify the label, as follows:

     UHL -- for a user header label (at the beginning of a data set).
     UTL -- for a 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; 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.

3. User Specified (76 bytes)

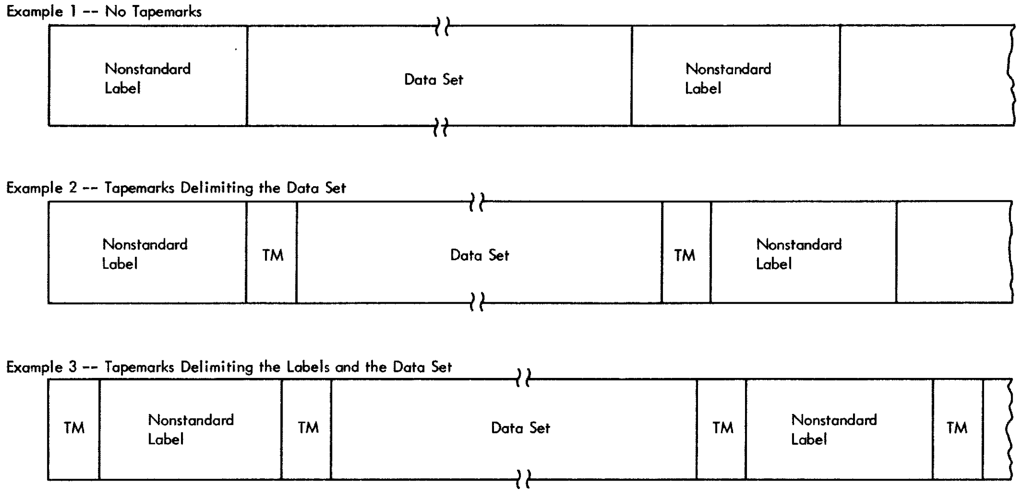
   - CONTENTS: Specified by the user.

   - PROCESSING: Specified in the DCB exit list.

Section III: Nonstandard Labels

Nonstandard labels do not conform to the 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 tape cannot be a standard volume label. In other words, the first record on the tape cannot be 80 characters in length with the identifier VOL1 as its first four characters.

Figure 9 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 multi-volume 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 condition for forward and backward read operations.



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 9. Examples of Tape Organizations with Nonstandard Labels

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).

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.

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 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. The various error conditions that can occur during verification of the first record are explained in Section V of 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. 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 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 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. (When the FEOV macro instruction is used to force an end-of-volume condition, the EOV routine does not give control to your input trailer label routine.) 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 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.

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. The various error conditions that can occur during verification of the first record are explained in Section V of this publication.

When the control program ensures that the first record on the tape is not a standard volume label, the Open or EOV 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 EOV 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).

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 will be 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 (prior to 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 in length and contains the identifier VOL1 in the first four bytes, the tape is not accepted. The control program issues a message 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 EOV -- 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 EOV or Close routines. The EOV routine handles end-of-volume conditions (reflective strip or FEOV macro instruction). The Close routine handles end-of-data 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 upon 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

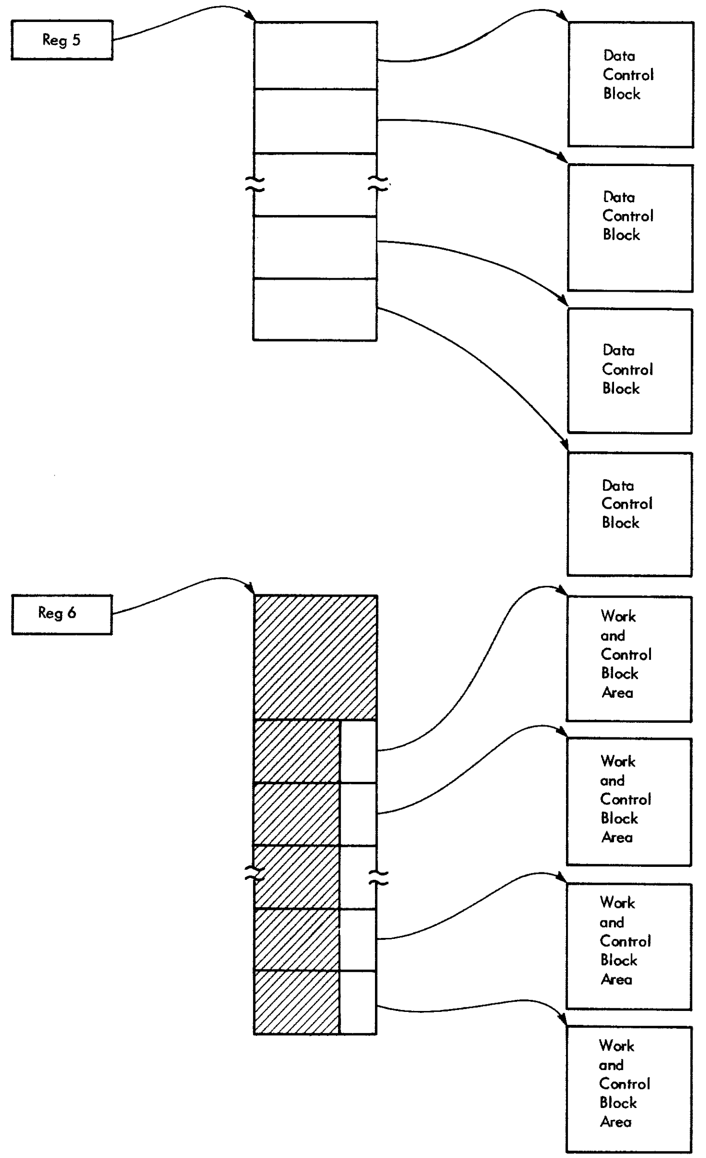
In processing nonstandard labels, you must perform many of the functions that the control program performs in processing standard labels. All input/output operations, such as reading labels, writing labels, and positioning volumes, must be performed by using the EXCP (execute channel program) macro instruction. 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 10.

- 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 11.

- 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 is shown in Figure 12.

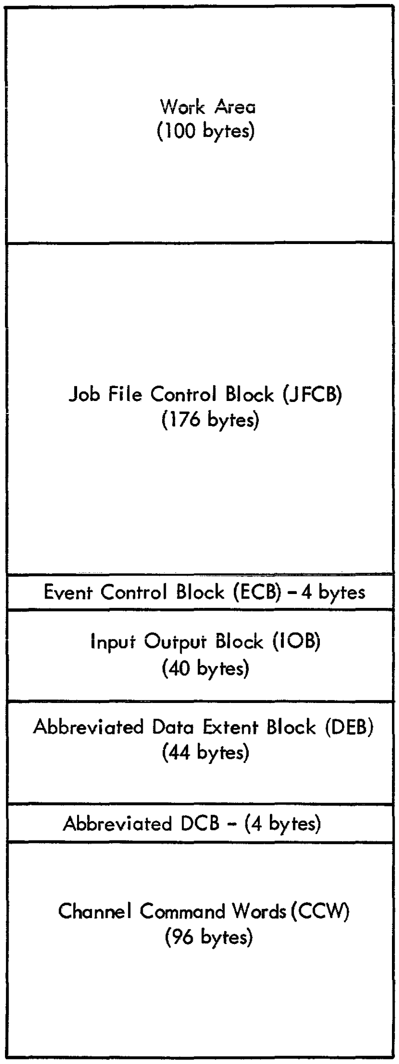
General flowcharts of nonstandard label processing routines are shown in Figures 13, 14, 15, and 16. These flowcharts suggest the logic that you could use in your routines. The logic is shown separately for routines receiving control from the Open, Close, EOV, or Restart routines of the control program. Each block in the flowcharts is numbered, and the number corresponds to an item in the list of explanations that follow.


Reg 5
Data Control Block
Data Control Block
Data Control Block
Data Control Block
Reg 6
Work and Control Block Area
Work and Control Block Area
Work and Control Block Area
Work and Control Block Area

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 three 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 11.

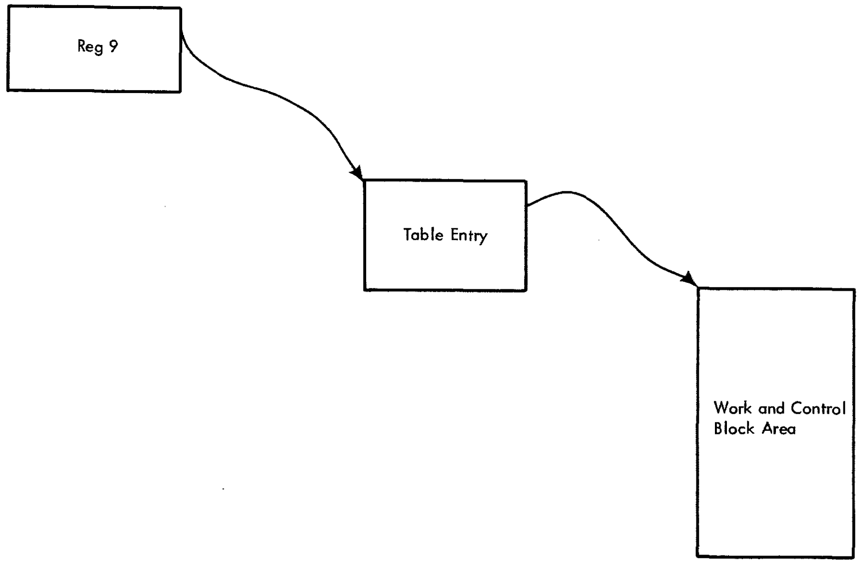
Figure 10. Status of Control Information and Pointers


Work Area
(100 bytes)
Job File Control Block (JFCB)
(176 bytes)
Event Control Block (ECB) - 4 bytes
Input Output Block (IOB)
(40 bytes)
Abbreviated Data Extent Block (DEB)
(44 bytes)
Abbreviated DCB - (4 bytes)
Channel Command Words (CCW)
(96 bytes)

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. (The macro definition and how to add it to the macro library are in the Appendix.) 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.

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, register 11 points to the DEB (if you receive control from the Close routine), or register 5 points to the DEB (if you receive control from the EOV routine).

Figure 11. Format of Combined Work and Control Block Area



Register 9 contains the starting address of 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 the Appendix.

•Figure 12. Status of Control Information and Pointers From the Control Program's Restart Routine

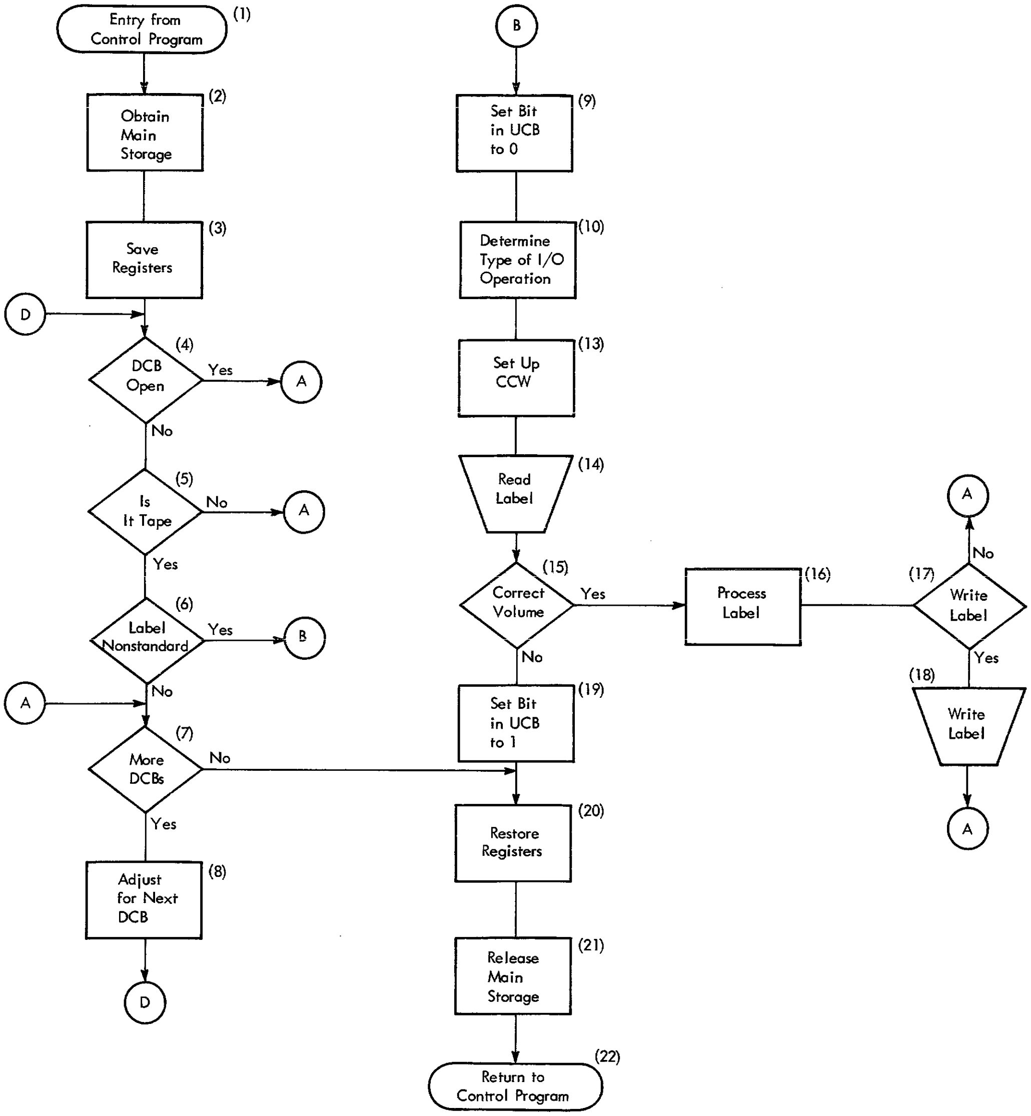


Figure 13. General Flow of a Nonstandard Label Processing Routine After Receiving Control From the Open Routine

NSL

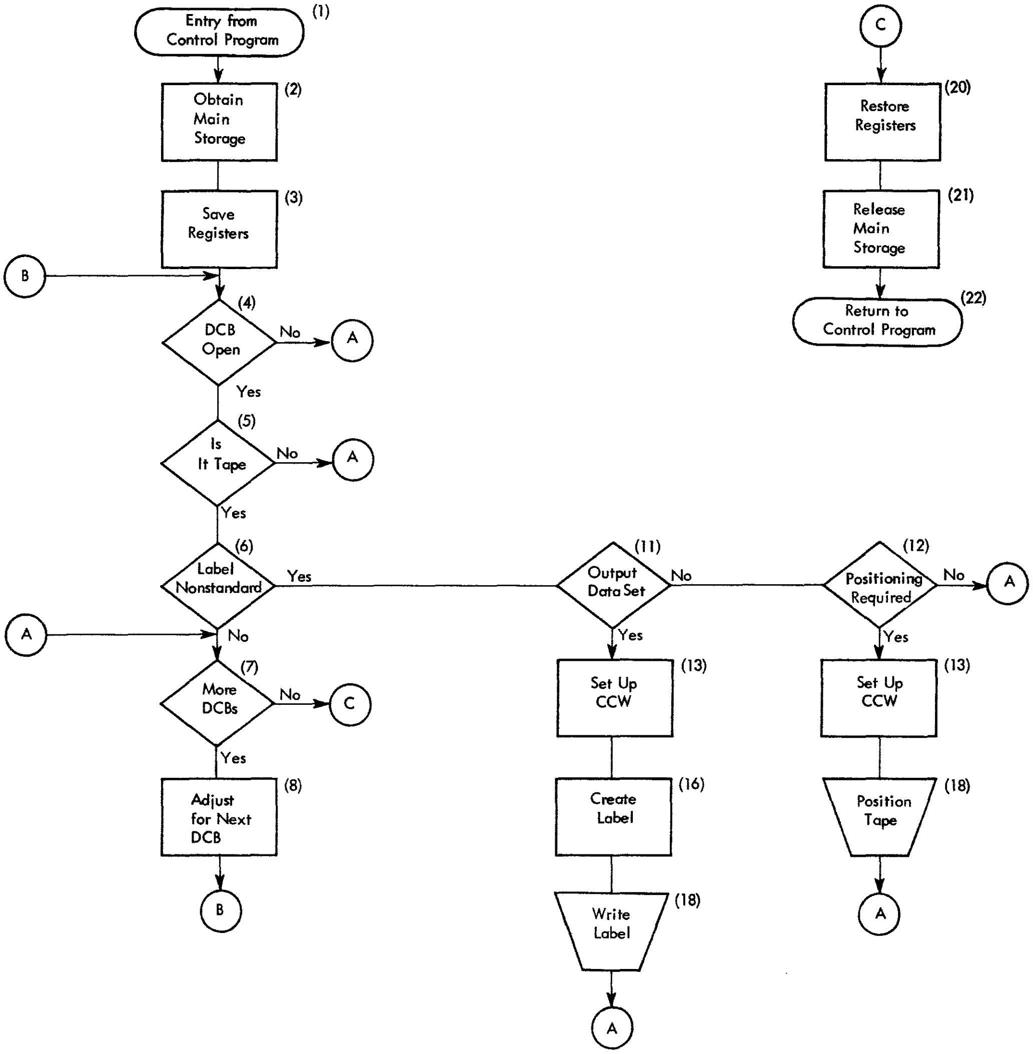


Figure 14. General Flow of a Nonstandard Label Processing Routine After Receiving Control From the Close Routine

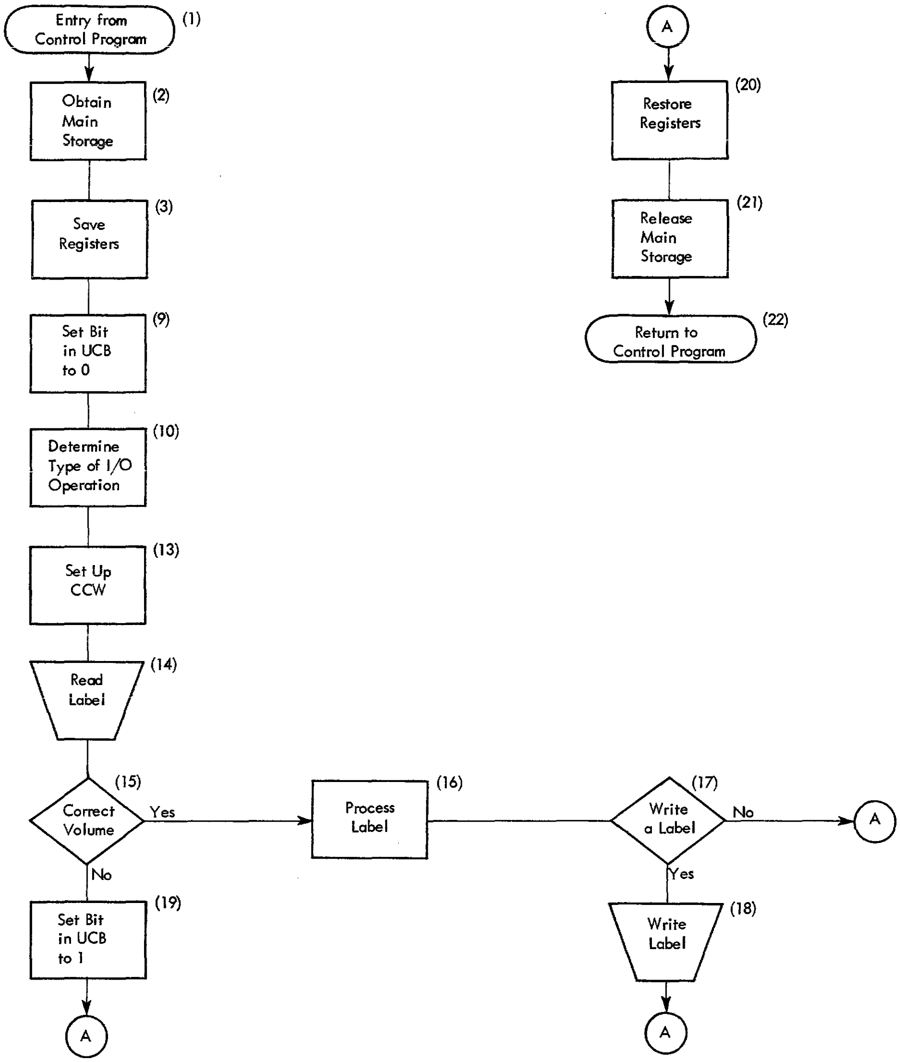
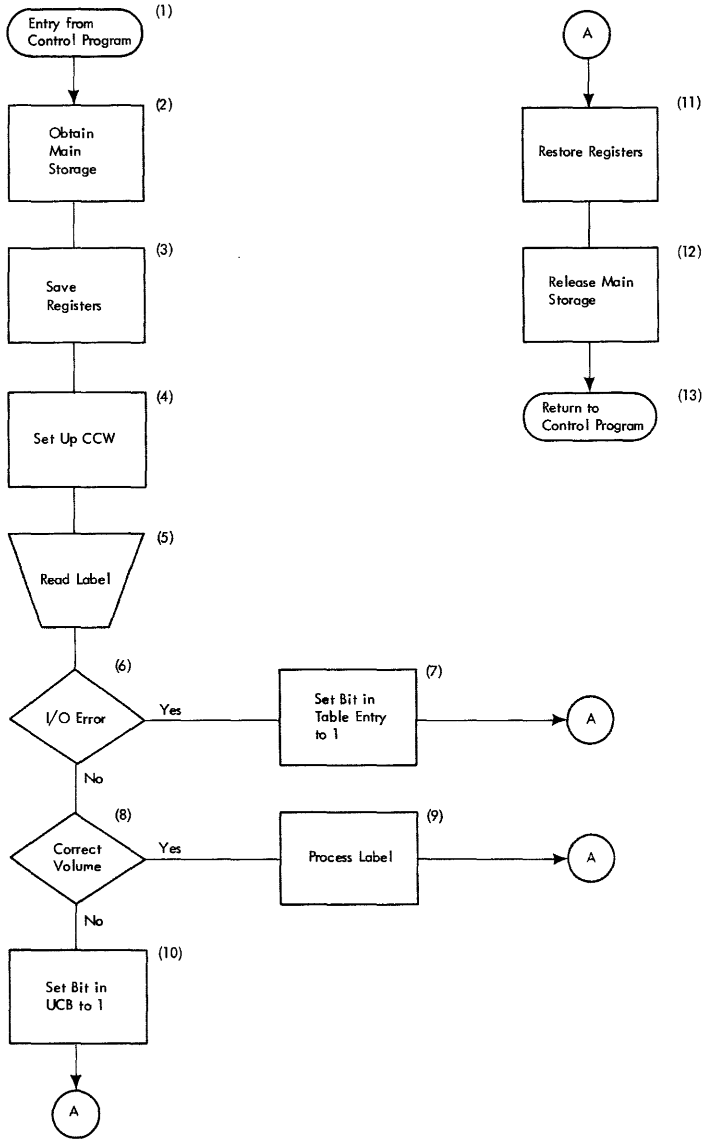


Figure 15. General Flow of a Nonstandard Label Processing Routine After Receiving Control From the EOV Routine



• Figure 16. General Flow of a Nonstandard Label Processing Routine After Receiving Control From the Restart Routine

Explanations of Logic Blocks -- Figures 13, 14, and 15

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 open, test bit 3 of the DCBOFLGS field of the DCB; if this bit is zero, the DCB has not been opened. (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 UCB3TAPE field of the unit control block (UCB); this bit is one for a tape data set. The symbolic names of all fields in the UCB are defined by the IEFUCBOB macro instruction. (The macro definition and how to add it to the macro library are in the Appendix.) The address of the UCB is contained in the DXDEBUCB field of the data extent block (DEB) as defined by the IECDSECT macro instruction. (The macro definition and how to add it to the macro library are in the Appendix; and a method of finding the address of the UCB is described in the System Programmer's Guide publication.)

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 one in its high-order bit position.

NSL

8. Add 4 to the contents of register 5; add 8 to the contents of register 6.

9. Set the high-order bit to zero in the SRTEDMCT field of the UCB.

10. To determine the type of I/O operation specified in the OPEN macro instruction, 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 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 one, the data-set mode is output; if this bit is zero, 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.)

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 will differ depending on the label formats used in the installation.

16. Label processing routines will 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.)

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

20. Use the load multiple instruction (LM).

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 save area pointer into
                                               reg 1
        LM       2,14,REGSAVE                  restore caller's regs
        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
                                               area
        LA       15,456(4)                     use area in OPEN workarea
        XCTL     EPLOC=(6),DCB=0,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                    save area pointer to reg 1
        LM       2,14,REGSAVE                  restore caller's regs
        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),DCB=0,SF=(E,(15))
MODNAME DC       C'IGGxxxxx'
```

Explanations of Logic Blocks -- Figure 16

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. 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.)

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 unrecoverable 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 will differ depending on the label formats used in the installation. The volume serial number for the mounted volume will be 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 instruction (LM).

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                   save area pointer to reg 1
        LM       2,14,REGSAVE                 restore caller's regs
        FREEMAIN R,LV=size,A=(1)
        BALR     15,0                         use 15 for temp base
        USING    *,15
        MVC      RSJFCB(8),MODNAME            module name overlays first
                                              8 bytes of the JFCB in
                                              work area
        LA       15,RSJFCB+8                  second 8 bytes of JFCB
                                              used for remote supv parm
                                              list required by XCTL
        XCTL     EPLOC=RSJFCB,SF=(E,(15))
MODNAME DC       C'IGC0K05B'
```

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 the System Generation publication.

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 Header	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 (i.e., 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, in an area provided by AVR, the location of the volume serial number field within the label. (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.)

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 Section V of 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 two-word area whose contents are as follows:

Word 1 -- contains the address of the beginning byte of the 80-byte work area.

Word 2 -- contains the address of a one-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 six-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 proper control program module. The modules and the object deck setups are as follows:

PCP -- 18K Scheduler: Module IEFALOC4

```
                (object deck)

INCLUDE             SYSLMOD(IEFALOC4)
ALIAS               IEFXV001,IEFWD000
ENTRY               IEFWD000
NAME                IEFALOC4(R)
```

PCP -- 44K Scheduler: Module IEFALOC1

```
                (object deck)

INCLUDE             SYSLMOD(IEFALOC1)
ALIAS               IEFXA
ENTRY               IEFXA
NAME                IEFALOC1(R)
```

PCP -- 100K Scheduler: Module GO

```
                    (object deck)

    INCLUDE             SYSLMOD(IEFALLOC)
    ALIAS               IEFXA
    ENTRY               IEFXA
    NAME                IEFALLOC(R)
```

MFT -- 30K Scheduler: Module IEFXVAVR

```
                    (object deck)

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

MFT -- 44K Scheduler: Module IEFXVAVR

```
                    (object deck)

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

MVT: Module IEFXVAVR

```
                    (object deck)

    INCLUDE             SYSLMOD(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 PCP configuration.

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

             (object deck)

    INCLUDE SYSLMOD(IEFALOC4)
    ALIAS   IEFXV001,IEFWD000
    ENTRY   IEFWD000
    NAME    IEFALOC4(R)
/*
```

Section IV: 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 17. 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.

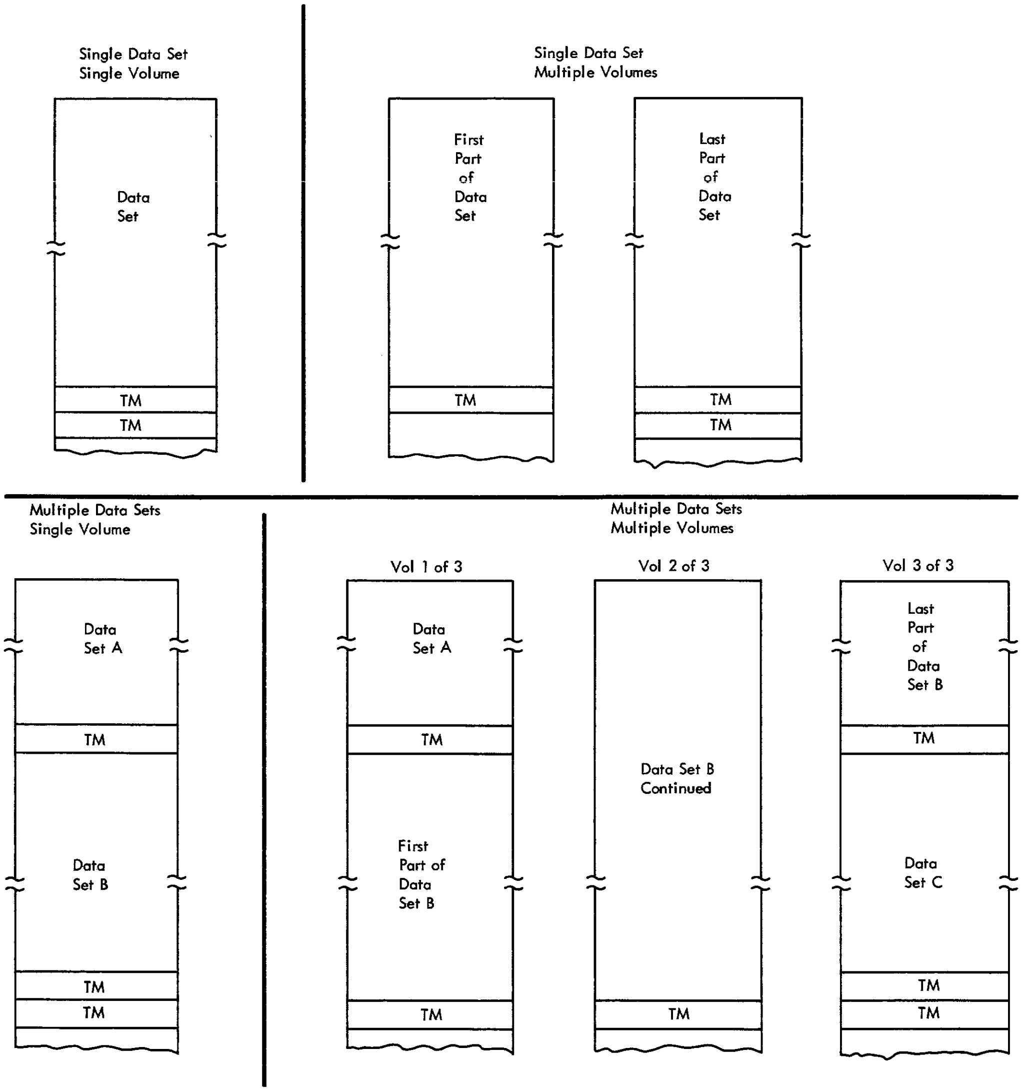


Figure 17. Organizations for Unlabeled Tapes

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 condition is signaled by the reflective strip at the beginning of the tape.

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 80 bytes in length and contains the identifier VOL1 in the first four bytes, the tape is not accepted. The various error conditions that can occur during verification of the first record are explained in Section V of 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 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 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 should precede 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 zero.

2. When a data set is opened, the Open routine sets the data set sequence number in the UCB to one. (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 equal, the tape is already positioned at the requested data set. If 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.

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. Multi-volume 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 EOV routine handles both end-of-data set and end-of-volume conditions. These conditions occur when:

- A tapemark is read.
- The end of the reel (reflective strip) is encountered.
- An FEOV (force-end-of-volume) 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 facility. 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 basis 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 tapemarks. 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 zero.

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 multi-volume 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 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 it reaches the end of the data set, the user's end-of-data 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 four bytes, the tape is not accepted. The various error conditions that can occur during verification of the first record are explained in Section V 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 is a system generation option (OPTIONS=BYLABEL). If this option is not specified at system generation, BLP has the same effect as 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 seven-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 a 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 mount 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. These volume serial numbers are generated in the form Lxxxyy where:

xxx -- is a number the Open routine increments (by one) each time an output data set is opened on a nonspecific 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 multi-volume data set is assigned a different volume serial number.

If a data set is to be cataloged, you should specify the volume serial numbers for all the volumes required. This will prevent 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 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 will be the first or only data set on the volume, so the tape will be 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 two greater than the number of data sets existing 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, 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 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.

NL
BLP

End of Volume on Output

Data management's EOV 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 EOV 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 the CLOSE macro instruction is executed, data management checks a switch in the data control block to determine if the last operation was a non-write operation (read or point). If a non-write operation was performed last, no tapemarks are written. Otherwise, the Close routine writes two tapemarks following the new data set. If another data set is later added to the tape (multi-data set organization), its first record overwrites the second tapemark.

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 zero or a positive number, the tape is positioned at the interrecord gap following the last record of the data set).

Section V: Volume Label Verification and Volume Label Editor Routines

If you specify that an input or output tape has standard labels, the operating system attempts to verify the existence of a 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 the nonexistence of 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 request mounting of another tape. 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 nine-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 seven-track record, unit check occurs.

- If a dual-density nine-track tape unit is used, the record is read in its existing density. If the record is a seven-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 seven-track tape is an 800 bpi nine-track tape -- because the sensing facility will not receive the "burst" that indicates 1600 bpi.

- If a seven-track tape unit is used, the 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 nine-track record, 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 18, 19, and 20. Figure 18 shows the actions when standard labels are specified. Figure 19 shows the actions when nonstandard labels are specified; and Figure 20 shows the actions when no labels are specified.

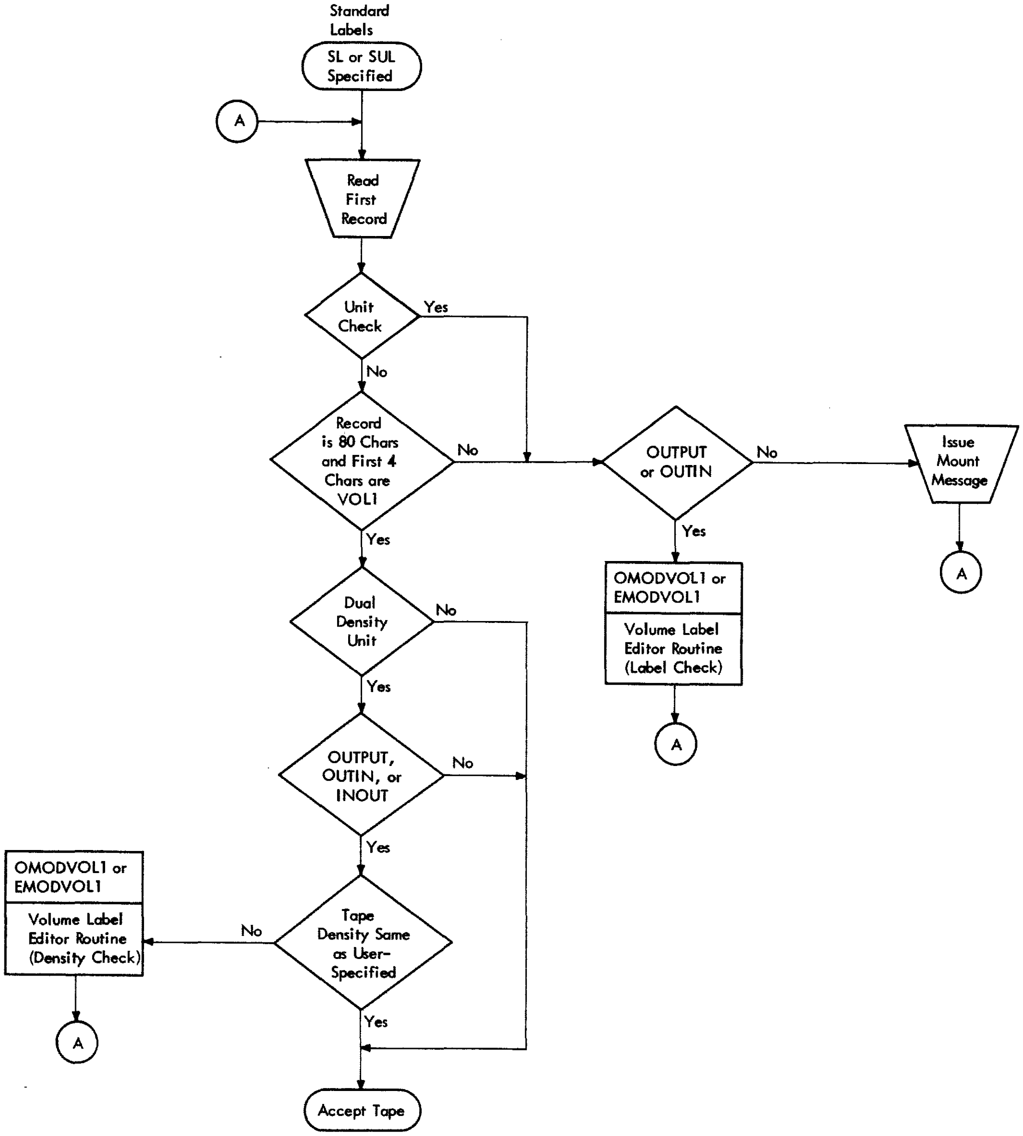


Figure 18. Verification of First Record When Standard Labels Are Specified

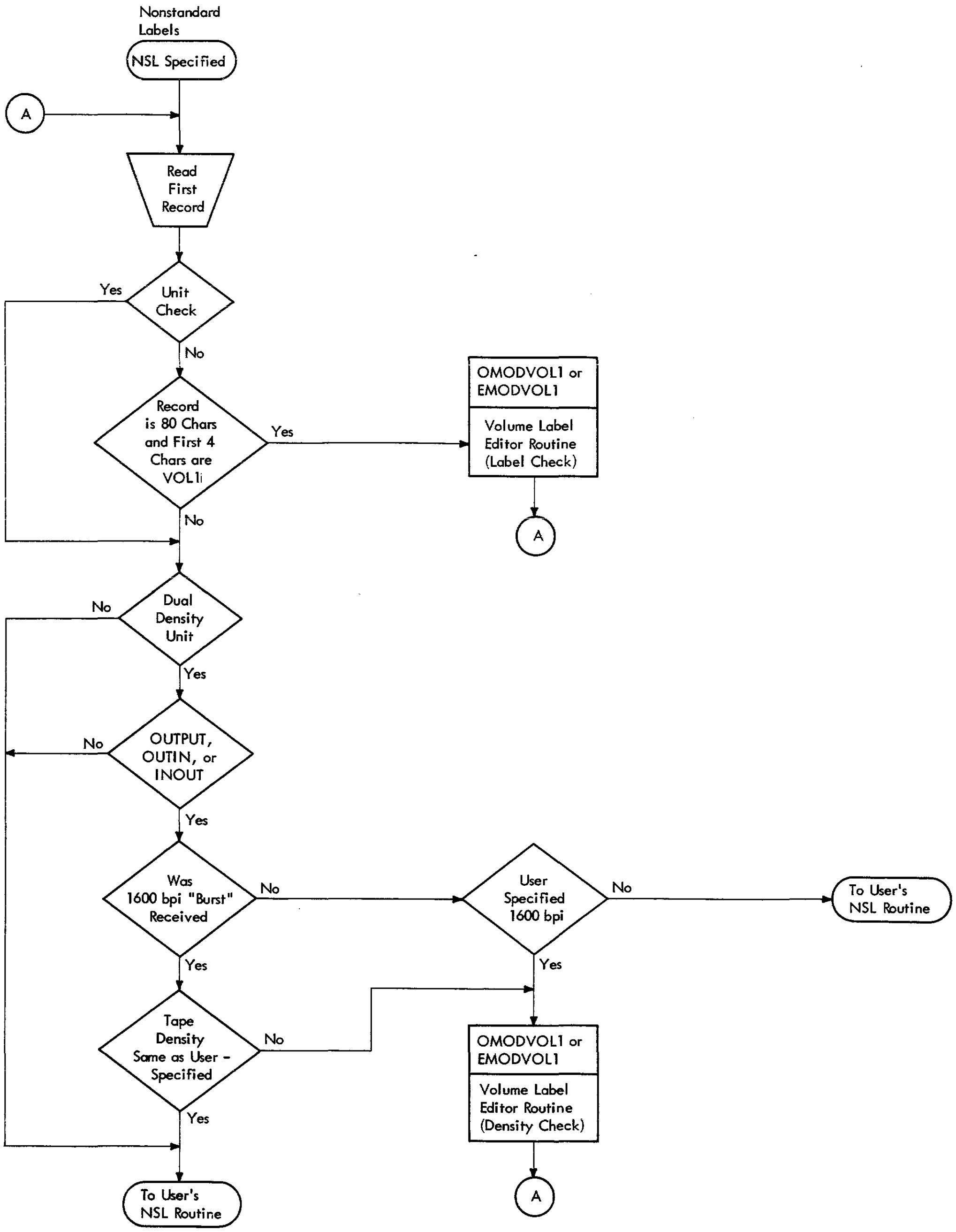


Figure 19. Verification of First Record When Nonstandard Labels Are Specified

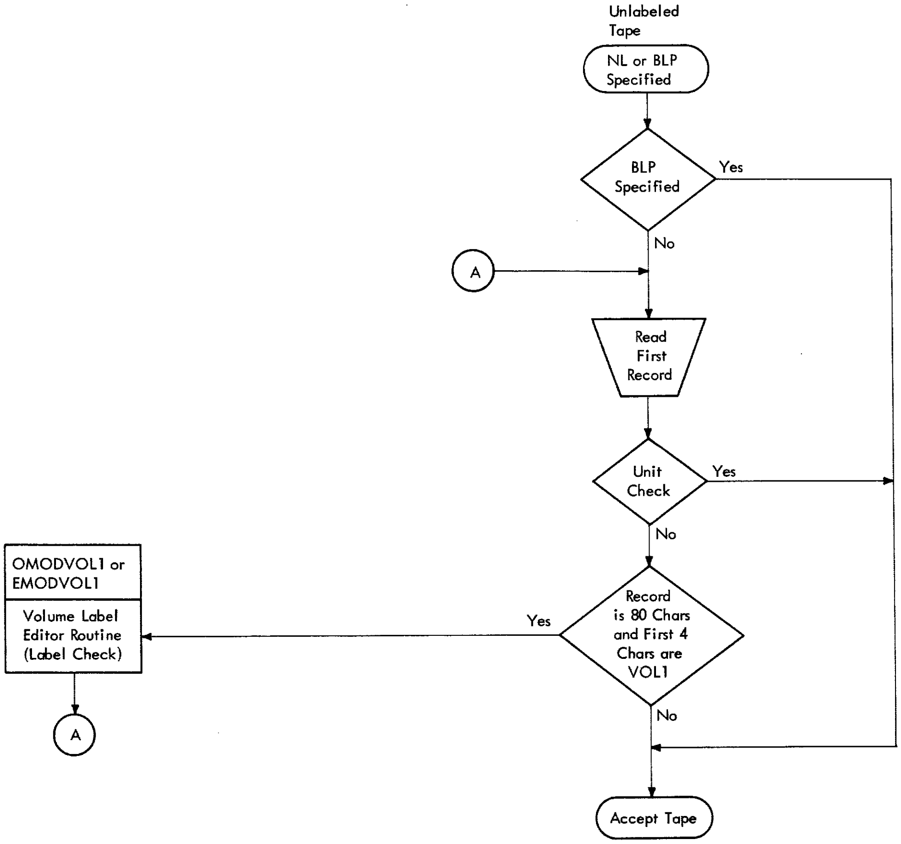


Figure 20. Verification of First Record When Unlabeled Tape Is Specified

Volume Label Editor Routines

When data sets are written on tape, data management's Open or EOV 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 requests the operator to demount the current tape volume, and to mount a new tape volume that conforms to the label or density specifications.

You can replace the IBM-supplied editor routines with installation routines that resolve the conflict without operator intervention. Your editor routines can resolve label and density conflicts by writing labels, by "cancelling" labels, 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 EOV routine for handling the second and subsequent volumes of a multi-volume 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 Section III of this manual (See Section III - Programming Conventions) in so far as size, design, register usage, entry points, and work areas are concerned. As discussed in Section III, 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 EOV.

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 EOV routine. Transfer between the modules must be by name.

Program Functions

Table 3 presents the four conditions under which the Open or EOV 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 21 and 22. 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 EOV 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 follow. 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 22 (the EOV editor routine) does not contain logic blocks corresponding to blocks 5, 18, and 16 in Figure 21 (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 EOV 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 13 in the flowcharts.

Table 3. Editor Routine Entry Conditions

Program Specification	Mounted Volume Characteristics	Transfer Occurs on a	Possible Editor Routine Action
SL - 800 bpi or 1600 bpi density	SL -- CD	Density check[1]	Overwrite the standard label with a standard label. The first write from loadpoint sets the recording density on a dual-density device. (See Figure 21 or 22 -- blocks 15b, 16 and explanation).
NSL[2] - 800 bpi or 1600 bpi density	NSL or NL -- CD	Density check[1]	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 EOV. (See Figure 21 - blocks 15, 15b, 16. If your installation supports protection and retention date checking on NSL volumes, see block 6).
SL	NSL or NL[3]	Label check[4]	Write a standard volume label. (see Figure 21 - blocks 15, 15a, 16. If your installation supports protection and retention date checking on NSL volumes, see block 6).
NL or NSL	SL[3]	Label Check	Overwrite standard label with a tapemark, i.e., "cancel." (See Figure 21 - blocks 15, 15a, 16) Depending on whether NL or NSL is specified by the program, Open or EOV will either position tape (NL) or transfer control to your nonstandard label routines (NSL) when control is returned to them.

Legend:

SL	- 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.

[1] Dual-density devices only.

[2] 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.

[3] If the volume is mounted on a dual-density device a density condition may also exist. It will be corrected by the write operation.

[4] 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.

<u>Note:</u> The Open and EOV routines position the tape at load point before transferring control to the editor routines.

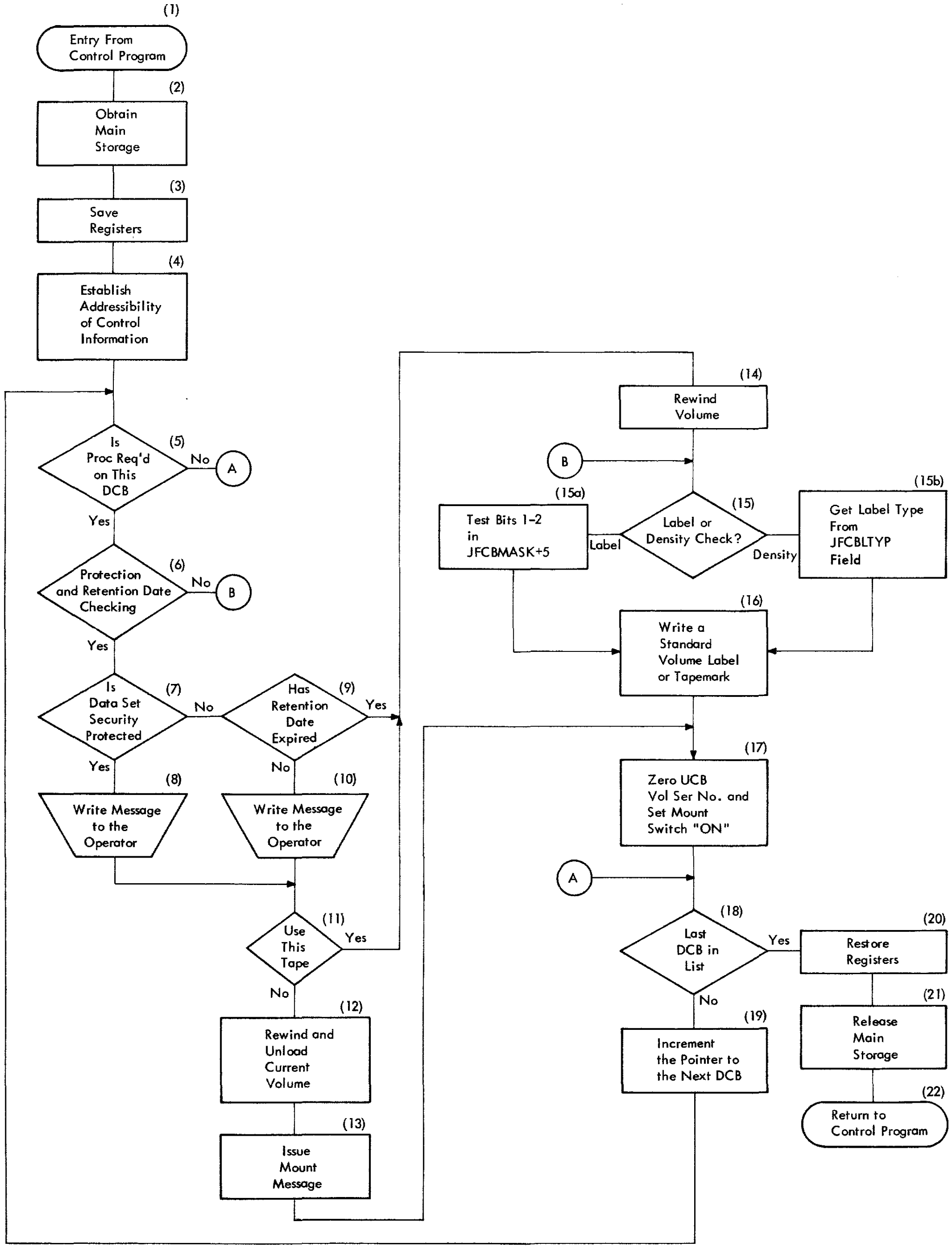

(1)
Entry From Control Program
(2)
Obtain Main Storage
(3)
Save Registers
(4)
Establish Addressibility of Control Information
(5)
Is Proc Req'd on This DCB
No
A
Yes
(6)
Protection and Retention Date Checking
No
B
Yes
(7)
Is Data Set Security Protected
No
(9)
Has Retention Date Expired
Yes
Yes
(8)
Write Message to the Operator
No
(10)
Write Message to the Operator
(11)
Use This Tape
Yes
No
(12)
Rewind and Unload Current Volume
(13)
Issue Mount Message
(14)
Rewind Volume
B
(15a)
Test Bits 1-2 in JFCBMASK+5
Label
(15)
Label or Density Check?
Density
(15b)
Get Label Type From JFCBLTYP Field
(16)
Write a Standard Volume Label or Tapemark
(17)
Zero UCB Vol Ser No. and Set Mount Switch "ON"
A
(18)
Last DCB in List
Yes
No
(19)
Increment the Pointer to the Next DCB
(20)
Restore Registers
(21)
Release Main Storage
(22)
Return to Control Program

Figure 21. General Flow of an Editor Routine After Receiving Control From the Open Routine

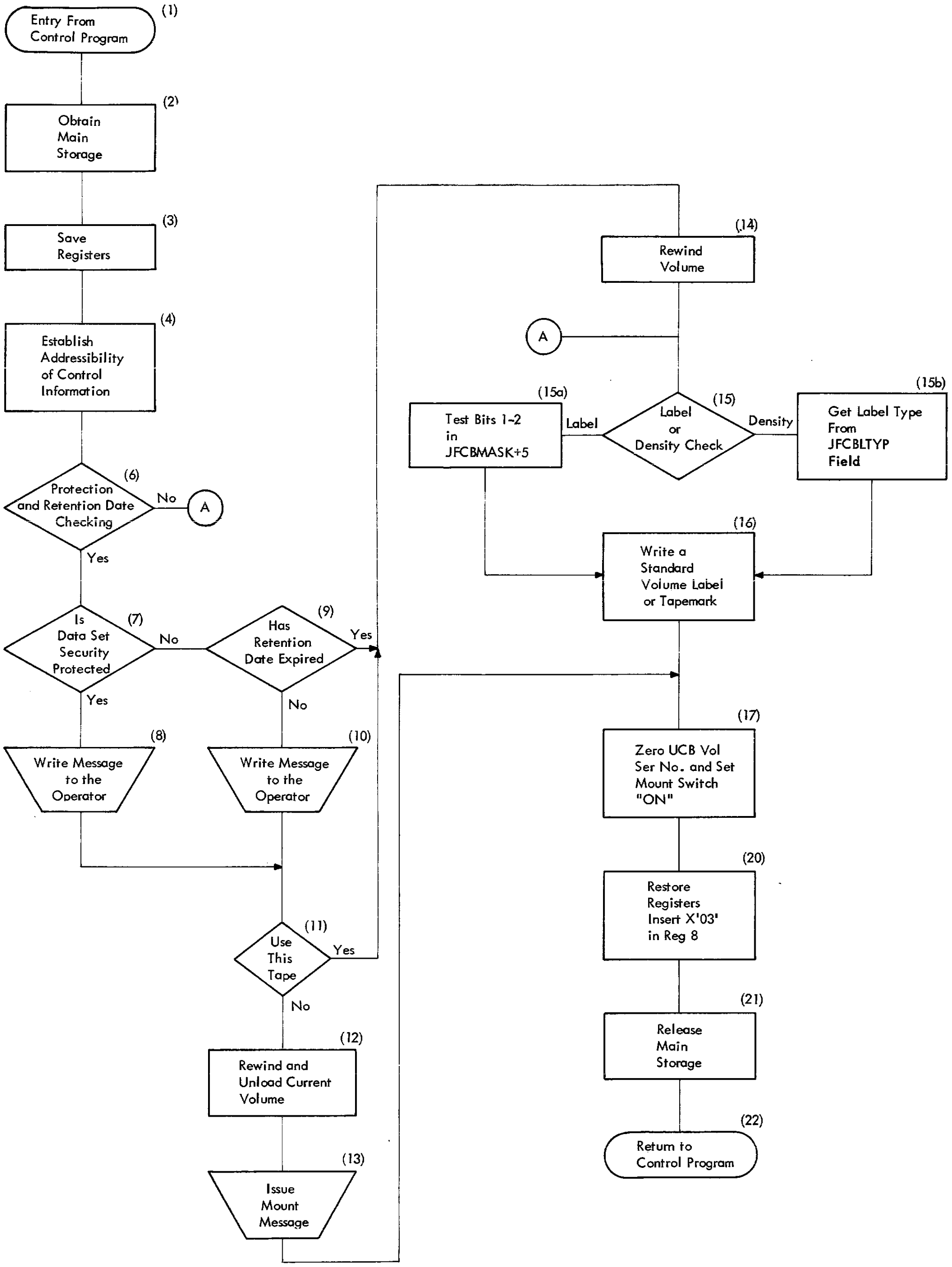

Entry From Control Program
(1)
Obtain Main Storage
(2)
Save Registers
(3)
Establish Addressibility of Control Information
(4)
Protection and Retention Date Checking
(6)
No
A
Yes
Is Data Set Security Protected
(7)
No
Has Retention Date Expired
(9)
Yes
Yes
Write Message to the Operator
(8)
No
Write Message to the Operator
(10)
Use This Tape
(11)
Yes
No
Rewind and Unload Current Volume
(12)
Issue Mount Message
(13)
Rewind Volume
(14)
A
Test Bits 1-2 in JFCBMASK+5
(15a)
Label
Label or Density Check
(15)
Density
Get Label Type From JFCBLTYP Field
(15b)
Write a Standard Volume Label or Tapemark
(16)
Zero UCB Vol Ser No. and Set Mount Switch "ON"
(17)
Restore Registers Insert X'03' in Reg 8
(20)
Release Main Storage
(21)
Return to Control Program
(22)

Figure 22. General Flow of an Editor Routine After Receiving Control From the EOV Routine

Logic Block Explanations

1. Your exception routine receives control via an XCTL macro instruction issued by the Open or EOV 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 instruction (STM).

4. Figure 10 in Section III of 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 EOV 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 shown in the Appendix symbolically defines the fields of the work and control block area (see Figure 11 in Section III).

   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 (see Appendix) 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.

   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) specified; no label/ nonstandard label on mounted volume.
2	1	No label (NL) or nonstandard label (NSL) specified; standard label 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 programs 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. Do not place a CCW at the location defined by DXCCW7 in your editor routine for Open. Do not place a CCW in locations DXCCW11 or DXCCW12 in your editor routine for EOV.

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 workarea. 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.

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 one, control was received as a result of a density check.

    Test bit 0 of the byte at JFCBMASK+5. If set to one, control was received as the result of a label check.

15a. 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.

15b. 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 EOV routines as a nonstandard label or no label; i.e., 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 EOV and re-verification of the label, the specification for label type and density will have been met; and Open or EOV will transfer control to your nonstandard label routines if NSL is specified or position the tape for writing if NL has been specified.

Section II of this publication contains the format and field descriptions for standard tape volume 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 six 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; i.e., 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. Exclusive OR (XC) the SRTEVOLI field with itself. OR(OI) 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 four bytes. You must also increment the pointer to the work and control block area for each DCB. You increment this pointer by eight bytes.

20. Use the load multiple instruction (LM). 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 Section III -- Explanation of Logic Blocks -- item 22.

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 EOV)
             .
             .
             .
ENTRY       EMODVOL1
NAME        EMODVOL1(R)
/*
```

<u>Caution</u>: You should not attempt to insert routines into the SVC library when you are running in a multi-tasking environment. Also, if your editor routines (for Open or EOV) 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.

Section VI: 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 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 5, 6, and 7 in Section II of 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, EOV1, 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 initialize the block count field in the data control block (for forward read) or to verify the count after processing (for backward 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 imbedded 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. The Job Control Language publication 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 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 Section III 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 the tape at the data set, you must specify the correct data set sequence number in the DD statement. On an unlabeled volume created by the operating system, the first data set on the tape is not preceded by a tapemark. If a tapemark should precede the first data set, you must add 1 to the data set sequence number. You must make similar adjustments to allow for any other excess tapemarks.

If a multi-volume data set from another system has a leading tapemark on each volume, the operating system cannot process it as an unlabeled multi-volume data set. The operating system always assumes that continued data sets are first in sequence on the tape, and the presence of the leading tapemarks makes each the second in sequence. However, the operating system can process this aggregate if you treat each volume as a concatenated data set with a data set sequence number of two.

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. Table 4 shows the component support for each type of label processing.

Table 4. Component Support of Label Processing Features

Item	Assembler	Linkage Editor	Sort/ Merge	Utilities	ALGOL	COBOL		FORTRAN			PL/I	RPG
						E	F	E	G	H		
Uses Data Management Facilities for Label Processing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard Labels (SL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard User Labels (SUL)	No	No	Yes	Yes	No	No	No	No	No	No	No	No
Supports Nonstandard Labels (NSL)[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Unlabeled Tape (NL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Bypass Label Processing Option (BLP)	Yes	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	No

[1] NSL can be specified only when installation-written routines that write and process the nonstandard labels have been incorporated into the operating system.

Appendix B: IECDSECT, IEFJFCBN, and IEFUCBOB Macro Instructions

If you want to use the IECDSECT, IEFJFCBN, and IEFUCBOB macro instructions, you must either add these macro definitions to the macro library (SYS1.MACLIB) or place them in a separate partitioned data set and concatenate this data set to the macro library. This section contains the following:

- The formats of the macro instructions.
- The job control and utility statements needed to add the macro definition to the library.
- The macro definition to be added to the library.

IECDSECT MACRO INSTRUCTION

This macro instruction defines the symbolic names of all fields in the work area used by the Open, Close, TCLOSE, and EOV routines. Code this macro instruction with blank name and operand fields, and precede it with a DSECT statement. Note: The IEFJFCBN macro instruction is used in the assembly of IECDSECT. The macro definition for IEFJFCBN must be present in the macro library (SYS1.MACLIB) for successful definition of all fields in the work area.

Name	Operation	Operand
	IECDSECT	

Control Statements Required

```
//jobname     JOB     {parameters}
//stepname    EXEC    PGM=IEBUPDTE,PARM=NEW
//SYSPRINT    DD      SYSOUT=A
//SYSUT2      DD      DSNAME=SYS1.MACLIB,DISP=OLD
//SYSIN       DD      DATA
./       ADD  NAME=IECDSECT,LIST=ALL
              .
              .
              .
              IECDSECT macro definition
              .
              .
              .
./       ENDUP
/*
```

IECDSECT Macro Definition

```
          MACRO
          IECDSECT
          SPACE 1
*                       THIS MACRO IS USED TO DEFINE THE WORK AREA
*                            FOR ALL MODULES OF OPEN,CLOSE,TCLOSE
*                            AND END OF VOLUME FOR O/S 360
          SPACE 1
*                       THIS MACRO DEFINES A WORK AREA WITH THE
*                            FOLLOWING FORMAT
          SPACE 1
*                            1.LABELS AND DSCB
*                                 LABELS
*                                      VOLUME LABEL
*                                      FILE LABEL 1
*                                      FILE LABEL 2
*                                 DSCB
*                                      FORMAT 1
*                                      FORMAT 3 KEY
*                                      FORMAT 3 DATA
*                                      CORE ADDRESS OF NEXT DSCB
*                                 MESSAGE AREA................ 100 BYTES
*                            2.JFCB........................... 176 BYTES
*                            3.ECB.............................. 4 BYTES
*                            4.IOB............................. 40 BYTES
*                            5.DEB............................. 44 BYTES
*                            6.DCB.............................. 4 BYTES
*                            7.CCW'S........................... 96 BYTES
          SPACE 1
*                                                   TOTAL *** 464 BYTES
          SPACE 2
* ***
* ***
* ***
* ***
          SPACE 1
*                       VOLUME LABEL
          SPACE 1
DXLBL     DS    0CL80
VOLLABI   DS    CL3                    LABEL IDENTIFIER
VOLNO     DS    CL1                    VOLUME LABEL NUMBER
VOLSERNO  DS    CL6
VOLSEC    DS    CL1
          DS    0CL10                  RESERVED
VOLVTOC   DS    CL5
          DS    CL5
          DS    CL10         RESERVED
          DS    CL10         RESERVED
VOLOWNER  DS    CL10         OWNER NAME AND ADDRESS CODE
          DS    CL29         RESERVED
          SPACE 1
*                       FILE LABEL 1
          SPACE 1
          ORG   DXLBL
FL1LABI   DS    CL3          LABEL IDENTIFIER
FL1NO     DS    CL1          FILE LABEL NUMBER
FL1ID     DS    CL17         FILE IDENTIFIER
FL1FILSR  DS    CL6          FILE SERIAL NUMBER
FL1VOLSQ  DS    CL4          VOLUME SEQUENCE NUMBER
FL1FILSQ  DS    CL4          FILE SEQUENCE NUMBER
FL1GNO    DS    CL4          GENERATION NUMBER
FL1VNG    DS    CL2          VERSION NUMBER OF GENERATION
FL1CREDT  DS    CL6          CREATION DATE
FL1EXPDT  DS    CL6          EXPIRATION DATE
```

```
FL1FSEC   DC    C'0'         FILE SECURITY INDICATOR
FL1BLKCT  DS    CL6          BLOCK COUNT
FL1SYSCD  DS    CL13         SYSTEM CODE
FL1RES    DS    0CL7         RESERVED FOR FUTURE USE
          DS    CL1
FL1RES1   DS    CL6
          SPACE 1
*                       FILE LABEL 2
          SPACE 1
          ORG   FL1ID
FL2RECFM  DS    CL1          RECORD FORMAT
FL2BLKL   DS    CL5          BLOCK LENGTH
FL2LRECL  DS    CL5          BLOCKING FACTOR/RECORD LENGTH
FL2DEN    DS    CL1          DENSITY
FL2FILP   DS    CL1          FILE POSITION
FL2JSID   DS    0CL17        JOB/STEP IDENTIFICATION
FL2JOBD   DS    CL8          JOB IDENTIFICATION
FL2JSSP   DC    C'/'         SLASH
FL2STEPD  DS    CL8          STEP IDENTIFICATION
FL2TRTCH  DS    CL2          TAPE RECORDING TECHNIQUE
FL2CNTRL  DS    CL1          CARRIAGE CONTROL CHARACTER
          DS    CL1          RESERVED FOR FUTURE USE
FL2DLKA   DS    CL1          BLOCK ATTRIBUTE
FL2RES    DS    CL41         RESERVED FOR FUTURE USE
          SPACE 1
*                       DATA SET CONTROL BLOCK
          SPACE 1
          ORG   DXLBL
DXDSCB    DS    0CL96
DSCFMTID  DC    C'1'
DSCFILSR  DS    CL6          FILE SERIAL NUMBER
DSCVOLSR  DS    CL2
DSCCREDT  DS    CL3          CREATION DATE IN DISCONTINUOUS BIN
DSCEXPDT  DS    CL3          EXPIRATION DATE IN DISCONTINUOUS BIN
DSCNOEXT  DS    CL1
DSCBLDBL  DS    CL1
          DS    CL1
DSCSYSCD  DS    CL13         SYSTEM CODE
          DS    CL7
DSCFILTY  DS    CL2          FILE TYPE
DSCRECFM  DS    CL1          RECORD FORMAT
DSCOPTCD  DS    CL1          OPTION CODE
DSCBLKL   DS    CL2          BLOCK LENGTH
DSCLRECL  DS    CL2          RECORD LENGTH
DSCKEYL   DS    CL1          KEY LENGTH
DSCRKP    DS    CL2          KEY LOCATION
DSCDSIND  DS    CL1
DSCSCALO  DS    CL4
DSCLSTAR  DS    CL5
DSCTRBAL  DS    CL2
DSCEXTYP  DS    CL1          EXTENT TYPE INDICATOR
DSCEXTSQ  DS    CL1          EXTENT SEQUENCE NUMBER
DSCLOWLM  DS    CL4
DSCUPPLM  DS    CL4
DSCEXT1   DS    CL10
DSCEXT2   DS    CL10
DSCNEXT   DS    CL5          POINTER TO NEXT RECORD
DSCCORE   DS    CL4          CORE ADDRESS OF NEXT DSCB RECORD
DSCBEND   EQU   *
          SPACE 1
*                       DATA SET CONTROL BLOCK -FORMAT 3- KEY PORTION
          SPACE 1
          ORG   DXDSCB
DXDSCB3K  DS    0CL40
DSCBF3C   DC    X'03030303'
```

```
DSCBEXSK  DS    0CL40
DSCBEXTY  DS    CL1          EXTENT TYPE INDICATOR
DSCBEXSQ  DS    CL1          EXTENT SEQUENCE NUMBER
DSCBLLMT  DS    CL4          CCHH LOWER LIMIT
DSCBULMT  DS    CL4          CCHH UPPER LIMIT
DSCBEX2   DS    CL10         ADDITIONAL EXTENT
DSCBEX3   DS    CL10         ADDITIONAL EXTENT
DSCBEX4   DS    CL10         ADDITIONAL EXTENT
          SPACE 1
*                       DATA SET CONTROL BLOCK -FORMAT 3- RECORD PORTION
          SPACE 1
          ORG   DXDSCB
DSCBFMID  DC    C'3'         FORMAT ID
DSCBEXSD  DS    0CL90        ADDITIONAL EXTENTS
DSCBEX5   DS    CL10         ADDITIONAL EXTENT
DSCBEX6   DS    CL10         ADDITIONAL EXTENT
DSCBEX7   DS    CL10         ADDITIONAL EXTENT
DSCBEX8   DS    CL10         ADDITIONAL EXTENT
DSCBEX9   DS    CL10         ADDITIONAL EXTENT
DSCBEXA   DS    CL10         ADDITIONAL EXTENT
DSCBEXB   DS    CL10         ADDITIONAL EXTENT
DSCBEXC   DS    CL10         ADDITIONAL EXTENT
DSCBEXD   DS    CL10         ADDITIONAL EXTENT
DSCBNEXT  DS    CL5          CCHHR OF NEXT FORMAT 3 DSCB
          SPACE 1
*                       MESSAGE AREA
          SPACE 1
          ORG   DXDSCB
REPLYLTH  DS    CL1
REPLYADR  DS    CL3
REPLYECB  DS    CL4
MSGLSTSZ  DS    CL4
MESSAGEA  DS    CL60
REPLY     DS    CL10
*
          ORG   MESSAGEA
*
*               DEFINITION OF LENGTH OF MESSAGE COMPONENTS
MSERL     EQU   3            MESSAGE SERIAL NUMBER LENGTH
MINSTL    EQU   6            MSG INSTRUCTION LTH INC MSG SER
MUNL      EQU   3            MESSAGE UNIT NAME LENGTH
MVOLL     EQU   6            MESSAGE VOLUME SERIAL LENGTH
* MTXTL         LENGTH MAY BE DEFINED BY EACH MODULE TO FIT REQUIREMENT
* MSGLTH        LENGTH OF FULL MSG DEFINED BY EACH MODULE
*               MESSAGE FORMAT IS 'IEC000A M 000,00000  (TEXT)
MSGIOSUP  DC    CL3'IEC'     I/O SUPPORT MESSAGE IDENTITY
MSGSER    DS    0CL3         MESSAGE SERIAL NUMBER
          ORG   MSGSER+MSERL-1
MSGSERLO  DS    CL1          VOLUME SERIAL LO ORDER BYTE
          ORG   MSGSER
MSGINSTR  DC    CL6'000A M'  MESSAGE INSTRUCTION INCL MSGSER
          ORG   MSGINSTR+MINSTL-1
MSGACTN   DS    CL1          MESSAGE ACTION REQD BY OPERATOR
          DC    C' '
MSGUN     DC    CL3'000'     UNIT NAME THAT MSG REFERS TO
          DC    C','
MSGVOLSR  DC    CL6'000000'  VOLUME SERIAL THAT MSG REFERS TO
          DC    C','
MSGTEXT   DS    0CL38
          SPACE 1
*                       JOB FILE CONTROL BLOCK
          SPACE 1
          ORG   DSCBEND
DXJBF     DS    0CL176
          IEFJFCBN
          SPACE 1
*                       EVENT CONTROL BLOCK
          SPACE 1
DXECB     DS    0CL4
          DC    X'00000000'
          SPACE 1
*                       INPUT/OUTPUT BLOCK
          SPACE 1
DXIOB     DS    0CL32
IOBFLAG1  DC    X'00'
IOBFLAG2  DC    X'00'
IOBSENSE  DS    0H
IOBSENS0  DS    CL1
IOBSENS1  DS    CL1          SENSE BYTE 1
IOBECBPT  DS    XL1
          DC    AL3(DXECB)
IOBCSW    DS    0D
IOBCOMAD  DC    X'00000000'  KEY,0000,COMMAND ADDRESS
IOBSTAT0  DC    X'00'        STATUS BYTE 0
IOBSTAT1  DC    X'00'        STATUS BYTE 1
IOBCNT    DC    X'0000'      COUNT
IOBSIOCC  DS    XL1
IOBSTART  DC    AL3(DXCCW)
IOBWGHT   DS    XL1
IOBDCBPT  DC    AL3(DXDCB)
          DS    XL1
          DS    XL3
IOBINCAN  DC    X'0000'
IOBERRCT  DS    XL2
DXDAADDR  DS    D            DIRECT ACCESS ADDRESS (MBBCCHHR)
          SPACE 1
*                       DATA EXTENT BLOCK
          SPACE 1
DYYYY     DS    0CL44
DXDEB     EQU   DYYYY-4
DXDEBDEB  DC    X'00000000'
DXDEBOFL  DS    0CL1
DXDEBIRB  DC    X'00000000'
DXDEBSYS  DC    X'00000000'
DXDEBUSR  DC    X'00000000'
DXDEBECB  DC    X'00000000'
DXDEBID   DS    0CL1
DXDEBDCB  DC    AL4(DXDCB)
DXDCBAD   EQU   DXDEBDCB
DXDEBAPP  DS    CL4
DXDEBMOD  DS    0CL1
DXDEBUCB  DS    F
DXDEBBIN  DS    H
DXDEBSCC  DS    H
DXDEBSHH  DS    H
DXDEBECC  DS    H
DXDEBEHH  DS    H
DXDEBNTR  DS    H
          SPACE 1
*                       DATA CONTROL BLOCK
          SPACE 1
DXXXX     DS    0F
DXDCB     EQU   DXXXX-44     POINTER TO RELATIVE BEGINNING OF DCB
DXDCBDEB  DC    A (DXDEB)
          SPACE 1
*                       CHANNEL CONTROL WORDS
          SPACE 1
          CNOP  0,8
DXCCW     DS    0CL96
DXCCW1    DS    D
```

```
DXCCW2    DS    D
DXCCW3    DS    D
DXCCW4    DS    D
DXCCW5    DS    D
DXCCW6    DS    D
DXCCW7    DS    D
DXCCW8    DS    D
DXCCW9    DS    D
DXCCW10   DS    D
DXCCW11   DS    D
DXCCW12   DS    D
          SPACE 1
DSECTSIZ  EQU   464           CORE AREA REQUIRED FOR THIS MACRO
          MEND
```

IEFUCBOB MACRO INSTRUCTION

This macro instruction defines the symbolic names of all fields in the unit control block (UCB). Code this macro instruction with blank name and operand fields, and precede it with a DSECT statement.

Name	Operation	Operand
	IEFUCBOB	

Control Statements Required

```
//jobname     JOB     {parameters}
//stepname    EXEC    PGM=IEBUPDTE,PARM=NEW
//SYSPRINT    DD      SYSOUT=A
//SYSUT2      DD      DSNAME=SYS1.MACLIB,DISP=OLD
//SYSIN       DD      DATA
./       ADD  NAME=IEFUCBOB,LIST=ALL
              .
              .
              .
              IEFUCBOB macro definition
              .
              .
              .
./       ENDUP
/*
```

IEFUCBOB Macro Definition

```
          MACRO
          IEFUCBOB
UCBOB     EQU   *                       UNIT CONTROL BLOCKS
          DS    0F
SRTEJBNR  DS    XL1                            JOB INTERNAL NUMBER
SRTECHAN  DS    XL1                     ALLOC.CHANNEL MASK
UCBID     DS    XL1                     UCB IDENTIFICATION
SRTESTAT  DS    XL1                     STATUS BITS
SRTEONLI  EQU   128                            ONLINE
SRTECHGS  EQU   64                             CHANGE ONLINE/OFFLINE
```

```
SRTERESV  EQU   32                            RESERVED DEVICE
SRTEUNLD  EQU   16                            UNLOAD THIS DEVICE
SRTEALOC  EQU   8                             BIT 4 ALLOCATED
SRTEPRES  EQU   4                             BIT 5 PERMANENTLY RESIDENT
SRTESYSR  EQU   2                             BIT 6 SYSRES
*                                                   OR PRIMARY CONSOLE
SRTEDADI  EQU   1                             BIT 7 DADSM INTERLOCK
*                                                   OR TAPE CONTAINS
*                                                   STANDARD LABELS, OR
*                                                   ALTERNATE CONSOLE
UCBCHA    DS    XL1                    FLAG1 AND CHANNEL ADDRESS
UCBUA     DS    XL1                    UNIT ADDRESS
UCBFL2    DS    XL1                    FLAG2
UCBDTI    DS    XL1                    DEVICE TABLE
UCBETI    DS    XL1                    ERROR TABLE
UCBSTI    DS    XL1                    STATUS TABLE
UCBLCI    DS    XL1                    LOGICAL CHANNEL TABLE
UCBATI    DS    XL1                    ATTENTION TABLE
UCBWGT    DS    XL1                    WEIGHT
UCBNAME   DS    CL3                    UNIT NAME IN 3 EBCDIC CHARACTERS
UCBTYP    DS    XL4                    DEVICE TYPE
UCBTBYT1  EQU   UCBTYP                        BYTE 1 OF UCBTYPE-MODEL
UCB1FEA0  EQU   128                           BIT 0 OF OPTION FIELD
UCB1FEA1  EQU   64                            BIT 1 OF OPTION FIELD
UCB1FEA2  EQU   32                            BIT 2 OF OPTION FIELD
UCB1FEA3  EQU   16                            BIT 3 OF OPTION FIELD
UCB1FEA4  EQU   8                             BIT 4 OF OPTION FIELD
UCB1FEA5  EQU   4                             BIT 5 OF OPTION FIELD
UCB1FEA6  EQU   2                             BIT 6 OF OPTION FIELD
UCB1FEA7  EQU   1                             BIT 7 OF OPTION FIELD
UCBTBYT2  EQU   UCBTYP+1                      BYTE 2 OF UCBTYPE-OPTIONS
UCBTBYT3  EQU   UCBTYP+2                      BYTE 3 OF UCBTYPE-CLASS
UCB3TAPE  EQU   128                           BIT 0 OF CLASS - TAPE
UCB3COMM  EQU   64                            BIT 1 OF CLASS - COMMUNIC.
UCB3DACC  EQU   32                            BIT 2 OF CLASS - DIRECT AC
UCB3DISP  EQU   16                            BIT 3 OF CLASS - DISPLAY
UCB3UREC  EQU   8                             BIT 4 OF CLASS - UNIT REC.
UCB3CHAR  EQU   4                             BIT 5 OF CLASS - CHAR.READ
UCBTBYT4  EQU   UCBTYP+3                      BYTE 4 OF UCBTYPE-DEVICE
UCBLTS    DS    XL2                    LAST 12*
UCBSNS    DS    XL6                    SENSE INFORMATION
SRTEVOLI  DS    CL6                    VOLUME SERIAL
SRTESTAB  DS    XL1                    STATUS B
SRTEBSVL  EQU   128                           BIT 0 SHARED VOLUME
SRTEBVSC  EQU   64                            BIT 1 VOLUME SECURITY
SRTEBALB  EQU   32                            BIT 2 ADDIT.VOL.LABEL PROC
SRTEBPRV  EQU   16                            BIT 3 PRIVATE
SRTEBPUB  EQU   8                             BIT 4 PUBLIC
SRTEBVQS  EQU   4                             BIT 5 VOLUME TO BE QUIESCE
*                                             BIT   TO MOUNT ANOTHER
SRTEBJLB  EQU   2                             BIT 6 JOBLIB VOLUME
SRTEBNUL  EQU   1                             BIT 7 CONTROL VOLUME
SRTEDMCT  DS    XL1                    DATA MANAGEMENT COUNT
SRTEFSCT  DS    XL2                    FILE SEQ. COUNT
SRTEFSEQ  DS    XL2                    FILE SEQ. NUMBER
UCBSQC    DS    2F                     SEEK QUEUE CONTROL WORD
UCBSKA    DS    2F                     MBBCCHHR FOR LAST SEEK
SRTEUSER  DS    XL1                    CURRENT NUMBER OF USERS
SRTEECBA  DS    XL3                    DA ECB ADDRESS
```

```
*THE FOLLOWING DESCRIBES ONE OF THE 10 SUB-UCBS FOR THE 2321--

         ORG   SRTEUSER
DATACELL DS    0CL16                        10 OF THESE ARE PRESENT FOR 2321
DCELBBNR DS    XL2                          BIN NUMBER
DCELSTAB DS    X                            STATUS B
DCELSTAT DS    X                            STATUS A
DCELVOLI DS    C16                          VOLUME SERIAL NUMBER
DCELJBNR DS    X                            INTERNAL JOB NUMBER
DCELDMCT DS    X                            DATA MANAGEMENT COUNT
DCELVTOC DS    XL3                          TTR OF VTOC START
DCELUSER DS    X                            CURRENT NUMBER OF USERS
         MEND
```

IEFJFCBN MACRO INSTRUCTION

This macro instruction defines the symbolic names of all fields in the job file control block (JFCB). Code this macro instruction with blank name and operand fields, and precede it with a DSECT statement.

Name	Operation	Operand
	IEFJFCBN	

Control Statements Required

```
//jobname      JOB      (parameters)
//stepname     EXEC     PGM=IEBUPDTE,PARM=NEW
//SYSPRINT     DD       SYSOUT=A
//SYSUT2       DD       DSNAME=SYS1.MACLIB,DISP=OLD
//SYSIN        DD       DATA
./       ADD   NAME=IEFJCBN,LIST=ALL
               .
               .
               .
               IEFJFCBN macro definition
               .
               .
               .
./       ENDUP
/*
```

IEFJFCBN Macro Definition

```
         MACRO
         IEFJFCBN
INFMJFCB EQU   *
JFCBDSNM DS    CL44                         DATA SET NAME
JFCBELNM DS    CL8                          ELEMENT NAME OR VERSION
JFCBISDM DS    CL1                          TASK SCHEDULER - DATA
*                                           MANAGEMENT INTERFACE BYTE
JFCBSYSC DS    CL13                         SYSTEM CODE
JFCBLTYP DS    CL1                          LABEL TYPE AND USER'S-LABEL
*                                           INDICATOR
         DS    CL1                          NOT USED
```

```
JFCBFLSQ  DS    CL2                    FILE SEQUENCE NUMBER
JFCBVLSQ  DS    CL2                    VOLUME SEQUENCE NUMBER
JFCBMASK  DS    CL8                    DATA MANAGEMENT MASK
JFCBCRDT  DS    CL3                    DATA SET CREATION DATE
JFCBXPDT  DS    CL3                    DATA SET EXPIRATION DATE
JFCBIND1  DS    CL1                    INDICATOR BYTE 1
JFCBRLSE  EQU   64                      BITS 0 AND 1 - EXTERNAL
*                                       STORAGE RELEASE INDICATOR
JFCBLOCT  EQU   16                      BITS 2 AND 3 - DATA SET
*                                       HAS BEEN LOCATED
JFCBNEWV  EQU   4                       BITS 4 AND 5 - NEW VOLUME
*                                       ADDED TO DATA SET
JFCBPMEM  EQU   1                       BITS 6 AND 7 - DATA SET IS
*                                       A MEMBER OF A PODS OR GDG
JFCBIND2  DS    CL1                    INDICATOR BYTE 2
JFCBSTAT  EQU   64                      BITS 0 AND 1 - DATA SET
*                                       STATUS (NEW, OLD, OR MOD)
JFCBSCTY  EQU   16                      BITS 2 AND 3 - DATA SET
*                                       SECURITY INDICATOR
JFCBUFNO  DS    0AL1
JFCBUFRQ  DS    AL1
JFCBFTEK  DS    0BL1
JFCBFALN  DS    BL1
JFCBUFL   DS    AL2
JFCEROPT  DS    BL1
JFCTRTCH  DS    0BL1
JFCKEYLE  DS    0AL1
JFCMOD    DS    0BL1
JFCCODE   DS    0BL1
JFCSTACK  DS    0BL1
JFCPRTSP  DS    BL1
JFCDEN    DS    BL1
JFCLIMCT  DS    AL3
JFCDSORG  DS    BL2
JFCRECFM  DS    BL1
JFCOPTCD  DS    BL1
JFCBLKSI  DS    AL2
JFCLRECL  DS    AL2
JFCNCP    DS    AL1
JFCNTM    DS    AL1
JFCRKP    DS    AL2
JFCCYLOF  DS    AL1
JFCDBUFN  DS    AL1
JFCINTVL  DS    AL1
JFCCPRI   DS    BL1
JFCSOWA   DS    AL2
JFCBNTCS  DS    CL1                    NUMBER OF OVERFLOW TRACKS
JFCBNVOL  DS    CL1                    NUMBER OF VOLUME SERIAL
*                                      NUMBERS
JFCBVOLS  DS    CL30                   VOLUME SERIAL NUMBERS (THE
*                                      FIRST FIVE)
JFCBEXTL  DS    CL1                    LENGTH OF BLOCK OF EXTRA
*                                      VOLUME SERIAL NUMBERS
*                                      (BEYOND FIVE)
JFCBEXAD  DS    CL3                    TRACK ADDRESS OF BLOCK OF
*                                      EXTRA VOLUME SERIAL NUMBERS
JFCBPQTY  DS    CL3                    PRIMARY QUANTITY OF D.A.
*                                      STORAGE REQUIRED
JFCBCTRI  DS    CL1                    INDICATES WHETHER CYLINDERS
*                                      TRACKS, OR RECORDS ARE
*                                      SPECIFIED IN JFCBPQTY AND
*                                      JFCBSQTY
JFCBSQTY  DS    CL3                    SECONDARY QUANTITY OF D.A.
*                                      STORAGE REQUIRED
JFCBIND3  DS    CL1                    INDICATOR BYTE 3
```

```
JFCBCNTG  EQU   64                      BITS 0 AND 1 - CONTIGUOUS
*                                       STORAGE INDICATOR
JFCBMXIG  EQU   16                     BITS 2 AND 3 - MAXIMUM
*                                       AVAILABLE EXTENT INDICATOR
JFCBALXI  EQU   4                       BITS 4 AND 5 - ALL EXTENTS
*                                       INDICATOR
JFCBRNDC  EQU   1                       BITS 6 AND 7 - ROUND
*                                       CYLINDER INDICATOR
JFCBDQTY  DS    CL3                    QUANTITY OF D.A. STORAGE
*                                      REQUIRED FOR A DIRECTORY
JFCBSPNM  DS    CL3                    CORE ADDRESS OF THE JFCB
*                                      WITH WHICH CYLINDERS ARE
*                                      SPLIT
JFCBABST  DS    CL2                    RELATIVE ADDRESS OF FIRST
*                                      TRACK TO BE ALLOCATED
JFCBSBNM  DS    CL3                    CORE ADDRESS OF THE JFCB
*                                      FROM WHICH SPACE IS TO BE
*                                      SUBALLOCATED
JFCBDRLH  DS    CL3                    AVERAGE DATA RECORD LENGTH
JFCBVLCT  DS    CL1                    VOLUME COUNT
JFCBSPTN  DS    CL1                    NUMBER OF TRACKS PER
*                                      CYLINDER TO BE USED BY THIS
*                                      DATA SET WHEN SPLIT
*                                      CYLINDERS IS INDICATED
JFCBLGTH  EQU   176               LENGTH OF JFCB
JFCBEND   EQU   *
          MEND
```

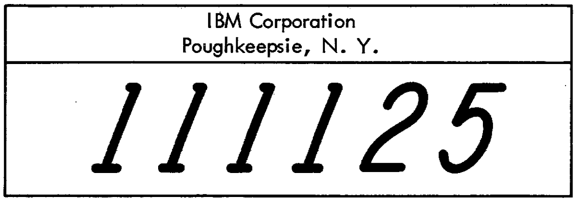
Appendix C: External Labels

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 23.



IBM Corporation
Poughkeepsie, N. Y.

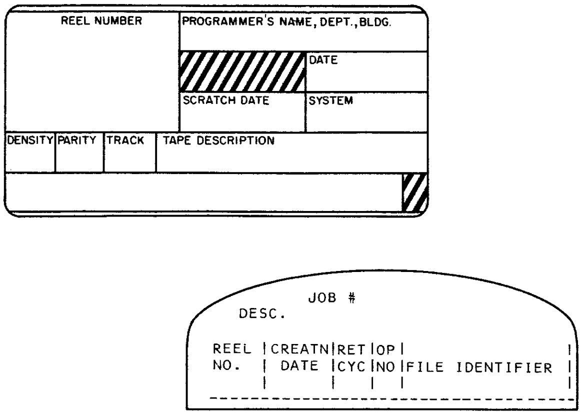
111125

Figure 23. 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 24.



REEL NUMBER	PROGRAMMER'S NAME, DEPT., BLDG.		
	▨	DATE	
	SCRATCH DATE	SYSTEM	
DENSITY	PARITY	TRACK	TAPE DESCRIPTION

JOB #

DESC.

```
REEL  |CREATN|RET |OP |                  |
NO.   | DATE |CYC |NO |FILE  IDENTIFIER  |
      |      |    |   |                  |
-----------------------------------------
```

Figure 24. External Labels for Contents Identification

Appendix D: 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 25 shows the format of the restart table entry. A description of each field follows the illustration.

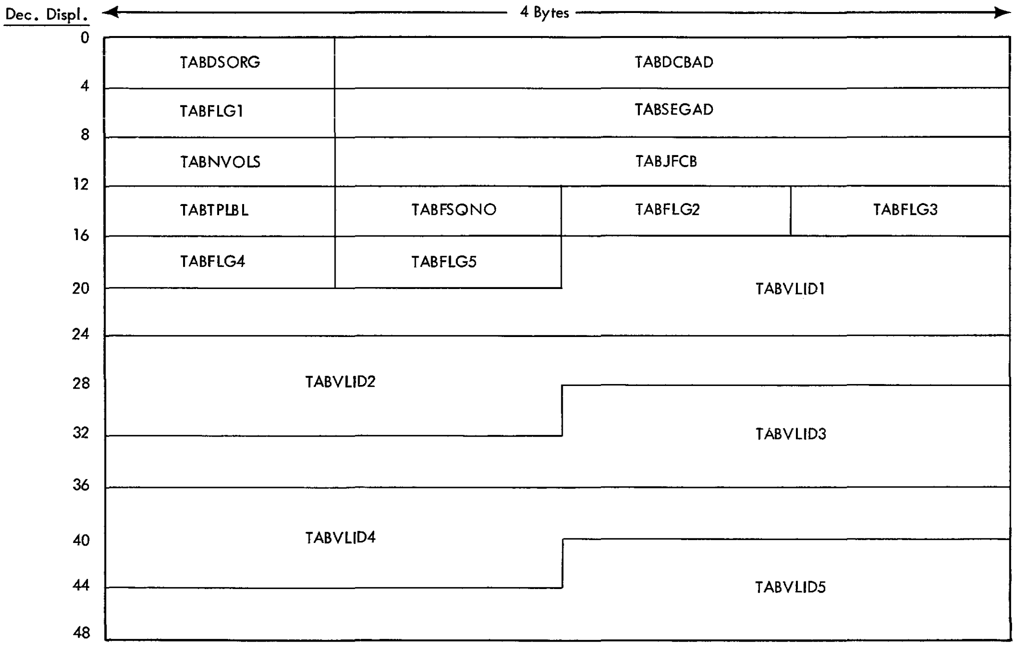


Figure 25. Restart Table Entry

Field Name	Bytes	Field Description
TABDSORG	1	This field describes the data set organization being used:

Bits	Settings	Meaning
0	1	Indexed Sequential Orgainization
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

Field Name	Bytes	Field Description
TABDCBAD	3	Address of the DCB
TABFLG1	1	This field contains the following information:

Bits	Settings	Meaning
0	1	Data set was specified in DD statement as NULLFILE or SYSCHECK
1	1	Data set was specified in DD statement as 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
6-7		Reserved for future use

Field Name	Bytes	Field Description
TABSEGAD	3	Address of the restart work and control block area for this data set.
TABNVOLS	1	The total number of volumes for this data set, as specified in the DD statement.
TABJFCB	3	The relative track address (TTR) of the JFCB.
TABTPLBL	1	This field contains the following tape label information:

Bits	Settings	Meaning
0	1	I/O error in NSL processing
1-2		Reserved
3	1	Bypass label processing
4		Reserved for future use
5	1	Nonstandard Labels
6	1	Standard Labels
7	1	No Labels

Field Name	Bytes	Field Description
TABFSQNO	1	Data set sequence number.
TABFLG2	1	This field contains the following information:

Bits	Settings	Meaning
0	1	More than five volumes associated with this data set
1	1	Partitioned Organization concatenation
2-7		Reserved

Field Name	Bytes	Field Description
TABFLG3	1	Reserved for future use.
TABFLG4	1	Reserved for future use.

TABFLG5	1	Reserved for future use.
TABVLID1	6	The volume serial number of the first volume to be mounted for this data set.
TABVLID2	6	The volume serial number of the second volume.
TABVLID3	6	The volume serial number of the third volume.
TABVLID4	6	The volume serial number of the fourth volume.
TABVLID5	6	The volume serial number of the fifth volume.

Work and Control Block Area

Figure 26 shows the format of the restart work and control block area. A description of the control block fields follows the illustration.

Symbolic Block Name	Dec. Disp.	4 Bytes				
RSDEB (48 bytes)	0	RSDEBTCB				Abbreviated Data Extent Block
	4	RSDEBDEB				
RSDCB (48 bytes)	8	RSDEBOFL	RSDEBIRB			Abbreviated Data Control Block
	12	RSDEBSYS				
	16	RSDEBUSR				
	20	RSDEBECB				
	24	RSDEBID	RSDEBDCB			
	28	RSDEBAPP				
	32	RSDEBMOD	RSDEBUCB			
	36	RSDEBBIN		RSDEBSCC		
	40	RSDEBSHH		RSDEBECC		
	44	RSDEBEHH		RSDEBNTR		
	48	RSECBAD				Event Control Block
	52	RSDCBDEB				
RSIOB (40 bytes)	56	RSIOBFG1	RSIOBFG2	RSIOBSN1	RSIOBSN2	Input/Output Block
	60	RSIOBECB				
	64	RSIOBCSW				
	72	RSIOBCPA				
	76	RSIOBDCB				
	80	RSIOBRCP				
	84	RSIOBECT		RSIOBINC		
	88	RSIOBDAD				
RSCCWLST (24 bytes)	96	RSCCW1				Channel Program
	104	RSCCW2				
	112	RSCCW3				
RSUBSEG (176 bytes)	120	RSJFCB				Job File Control Block
RSSTATUS (8 bytes)	296	RSSTAT1	RSSTAT2	RSSTAT3	RSSTAT4	Restart Status
	300	Reserved				
	304					

Figure 26. Restart Work and Control Block Area

Field Name	Bytes	Field Description
RSDEBTCB	4	Address of TCB for this DEB.
RSDEBDEB	4	Address of the next DEB in the same task.
RSDEBOFL	1	Data set status flags.
RSDEBIRB	3	IRB address used for appendage asynchronous exits.
RSDEBSYS	4	Address of first IOB in the system purge chain.
RSDEBUSR	4	Address of first IOB in the user purge chain.
RSDEBECB	4	Address of a parameter list used to locate the purge ECB for an SVC purge request.
RSDEBID	1	A hex '0F' to identify this block as a DEB.
RSDEBDCB	3	Address of DCB associated with this DEB (RSDCB).
RSDEBAPP	4	Address of the I/O appendage vector table.
RSDEBMOD	1	Device modifier.
RSDEBUCB	3	Address of UCB.
RSDEBBIN	2	Bin number of direct access volume (data cell drive).
RSDEBSCC	2	Cylinder address for start of an extent limit.
RSDEBSHH	2	Track address for the start of an extent limit.
RSDEBECC	2	Cylinder address for the end of an extent limit.
RSDEBEHH	2	Track address for the end of an extent limit.
RSDEBNTR	2	Number of tracks allocated to a given extent.
RSECBAD	4	Event Control Block (ECB).
RSDCBDEB	4	Address of DEB associated with this DCB (RSDEB).
RSIOBFG1	1	Flag byte 1, as follows:

Bits	Settings	Meaning
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
6	1	IOB unrelated flag (i.e., nonsequential)
7	0	START
	1	RESTART

RSIOBFG2	1	Flag byte 2, as follows:

Bits	Settings	Meaning
0	1	Halt I/O has been issued
1	1	Sense will not be perfomed 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

RSIOBSN1	1	First sense byte (device dependent).
RSIOBSN2	1	Second sense byte (device dependent).
RSIOBECB	4	Address of the ECB to be posted (RSECBAD).
RSIOBCSW	8	CSW
RSIOBCPA	4	Address of the channel program to be executed (RSCCW1).
RSIOBDCB	4	Address of the DCB associated with this IOB (RSDCB).
RSIOBRCP	4	Restart address used by I/O supervisor error routines during error correction.
RSIOBECT	2	Value used to increment block count field in DCB for magnetic tape.
RSIOBINC	2	Used by I/O supervisor error routines to count temporary errors during retry.
RSIOBDAD	8	This field is used for direct access only.
RSCCW1	8	Channel program area.
RSCCW2	8	Channel program area.
RSCCW3	8	Channel program area.
RSJFCB	176	Work area for Job File Control Block.
RSSTAT1	1	Status byte 1.
RSSTAT2	1	Reserved for future use.
RSSTAT3	1	Reserved for future use.
RSSTAT4	1	Reserved for future use.

Index

READER'S COMMENT FORM

IBM System/360 Operating System
Tape Labels

Form C28-6680-1

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

	Yes	No
• Does this publication meet your needs?	□	□
• Did you find the material:		
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
Well illustrated?	□	□
Written for your technical level?	□	□

- What is your occupation? ______________________________
- How do you use this publication?

As an introduction to the subject?	□	As an instructor in a class?	□
For advanced knowledge of the subject?	□	As a student in a class?	□
For information about operating procedures?	□	As a reference manual?	□

Other ______________________________

- Please give specific page and line references with your comments when appropriate.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE...

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

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

fold fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

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

POSTAGE WILL BE PAID BY . . .

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

Attention: Programming Publications, Dept. 232

fold fold

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]