z/OS
2.5

DFSMS Using Magnetic Tapes

IBM

SC23-6858-50

Note

Before using this information and the product it supports, read the information in "Notices" on page 137.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2021-09-30

Contents

Figures

Tables

About this book

This book describes how DFSMS processes magnetic tape labels. It is intended for system and application programmers who manage data stored on tapes.

Required product knowledge

To use this book effectively, you should be familiar with:

- Data management
- Job control language.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to IBM Documentation (www.ibm.com/docs/en/zos).

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xv.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Documentation function
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS DFSMS Using Magnetic Tapes, SC23-6858-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OS Version 2 Release 5 (V2R5)

This information contains no technical changes for this release.

Summary of changes for z/OS Version 2 Release 4 (V2R4)

Changed

The following content is changed.

- Some formatting issues were fixed.

Summary of changes for z/OS Version 2 Release 3 (V2R3)

The following changes are made for z/OS Version 2 Release 3 (V2R3). The most recent updates are listed at the top of each section.

There have been no changes made to this manual for z/OS Version 2 Release 3 (V2R3).

Chapter 1. Introduction to tape processing

Labels are used to identify magnetic tape volumes and the data sets on those volumes. The term "volume" applies to reels, which are round, and to cartridges, which are rectangular. See "Cartridge label" on page 117. In this book the term "data set" is synonymous with "file." That is not true in all z/OS books. You can process tape volumes with the following labels:

- IBM standard labels
- Labels that follow standards published by:
  - International Organization for Standardization (ISO)
  - American National Standards Institute (ANSI)
- Nonstandard labels
- No labels

Your installation can install a bypass for any type of label processing; however, the use of labels is recommended as a basis for efficient control of your tape volumes. The operating system performs label processing and most tape positioning. Unlike many operating systems, application programs on z/OS generally do not perform these functions. Therefore, application programs can be device-independent.

IBM standard tape labels consist of *volume labels* and groups of *data set labels*. The volume label, identifying the volume and its owner, is the first record on the tape. The data set label, identifying the data set and describing its contents, precede and follow each data set on the volume:

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

Usually, the formats of ISO and ANSI labels, which are defined by the respective organizations, are similar to the formats of IBM standard labels. Unless otherwise specified, the term *standard label,* as used in this manual, refers to IBM, ISO, and ANSI standard labels. However, whereas ISO and ANSI labeled tapes are coded in the American National Standard Code for Information Interchange (ASCII), IBM labeled tapes are coded either in the extended binary-coded-decimal interchange code (EBCDIC) or in binary-coded-decimal interchange code (BCDIC).

Nonstandard tape labels can have any format and are processed by routines that the installation provides. Unlabeled tapes contain only data sets and tape marks.

Figure 1 on page 2 shows the IBM standard, ISO and ANSI standard, nonstandard, and unlabeled tape layouts for a single data set on a single volume. Detailed layouts and variations for each type are illustrated and described in the appropriate sections of this manual.

Tape volumes with standard tape labels may be defined to Resource Access Control Facility (RACF), a component of the SecureWay Security Server for z/OS, by volume serial under the TAPEVOL class of entities. RACF authorization checking is performed for every standard labeled tape if system-wide tape protection has been specified. No protection is specifically extended by the system to nonstandard labeled tapes, but the installation-written nonstandard tape label routines may provide such protection.

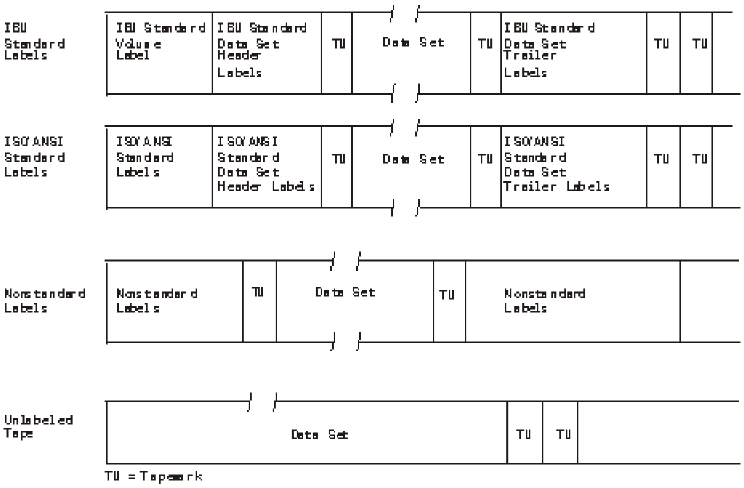


Figure 1. Basic tape layouts

Describing the labels

In the job control statements, you must provide a data definition (DD) 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:

Code
    Meaning

SL
    IBM standard labels

AL
    ISO/ANSI labels

SUL
    Both IBM Standard and user header or trailer labels

AUL
    Both ISO/ANSI and user header or trailer labels

NSL
    Nonstandard labels

NL
    No labels

BLP
    Bypass label processing

    The data set is treated in the same manner as if NL had been specified, except that the system does not check for an existing volume label. If your installation does not allow BLP, the data set is treated exactly as if NL had been specified. Your job can use BLP only if Job Entry Subsystem (JES), RACF, or DFSMSrmm allow it.

LTM
    Bypass a leading tape mark, if encountered, on unlabeled tapes.

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

RACF protects SL, AL, SUL, and AUL tape volumes. The installation may provide support for RACF protection of NSL tapes. When the first data set on the first volume specified in the DD statement is created, the volume may be automatically defined to RACF if PROTECT=YES is coded in the DD statement.

PROTECT=YES may be specified for SL, AL, NL, SUL, AUL, BLP, and NSL tape volumes. For NSL tapes, the first data set on the first volume is not a criterion for a valid PROTECT specification.

The data set sequence subparameter of the LABEL parameter specifies 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 on the reel.

When INOUT, OUTIN, or OUTINX is specified as the processing method in the OPEN macro, the LABEL parameter can 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 or OUTINX is specified and you want the data set processed for output only, code the subparameter OUT in the LABEL parameter. INOUT is not supported for ISO and ANSI tapes, but is treated as input if the subparameter IN is used 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. Beginning with RACF 1.7, the LABEL parameter value is also used by RACF to determine the security expiration. If not specified, the expiration date records as zeros, allowing the data set to be overwritten immediately, unless DFSMSrmm is managing the volume. 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. You can use a data class to specify all of your data set's attributes such as record length and record format but not data set name and disposition. Specify the name of the data class using the JCL keyword DATACLAS. If you do not specify a data class, the automatic class selection (ACS) routines assign a data class based on the defaults defined by your storage administrator. See *z/OS DFSMSdfp Storage Administration* for more information on data class.

Allocating tape data sets

An example of allocating a tape data set using DATACLAS in the DD statement follows. TAPE01 is the name of the data class.

```
//NEW DD DSN=DATASET.NAME,UNIT=TAPE,DISP=(,CATLG,DELETE),DATACLAS=TAPE01
```

Completing the data control block

The information contained in the DD statement or a dynamic allocation request is 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. Optionally, the processing program also supplies a DCB extension (DCBE). When completed, the DCB and DCBE contain full descriptive information about the data set, and provide the connection between the data set, the processing program, and the operating system.

The system can determine the optimum block size for new tape data sets. For more information, see *z/OS DFSMS Using Data Sets*.

Most of the information recorded in the DCB and DCBE is obtained in the following ways:

- The DCB and DCBE macro in the processing program or a DCB or DCBE created by a compiler
- The JFCB and other control blocks, which the system creates from the DD statement in the input stream or from dynamic allocation parameters
- Data class definition, if any
- The data set label (if this is an existing data set)

The DCB is completed at execution time, when it is opened. Figure 2 on page 4 illustrates the sequence of filling in the DCB information. Steps **3** and **8** are bypassed if the tapes have nonstandard labels or no labels.

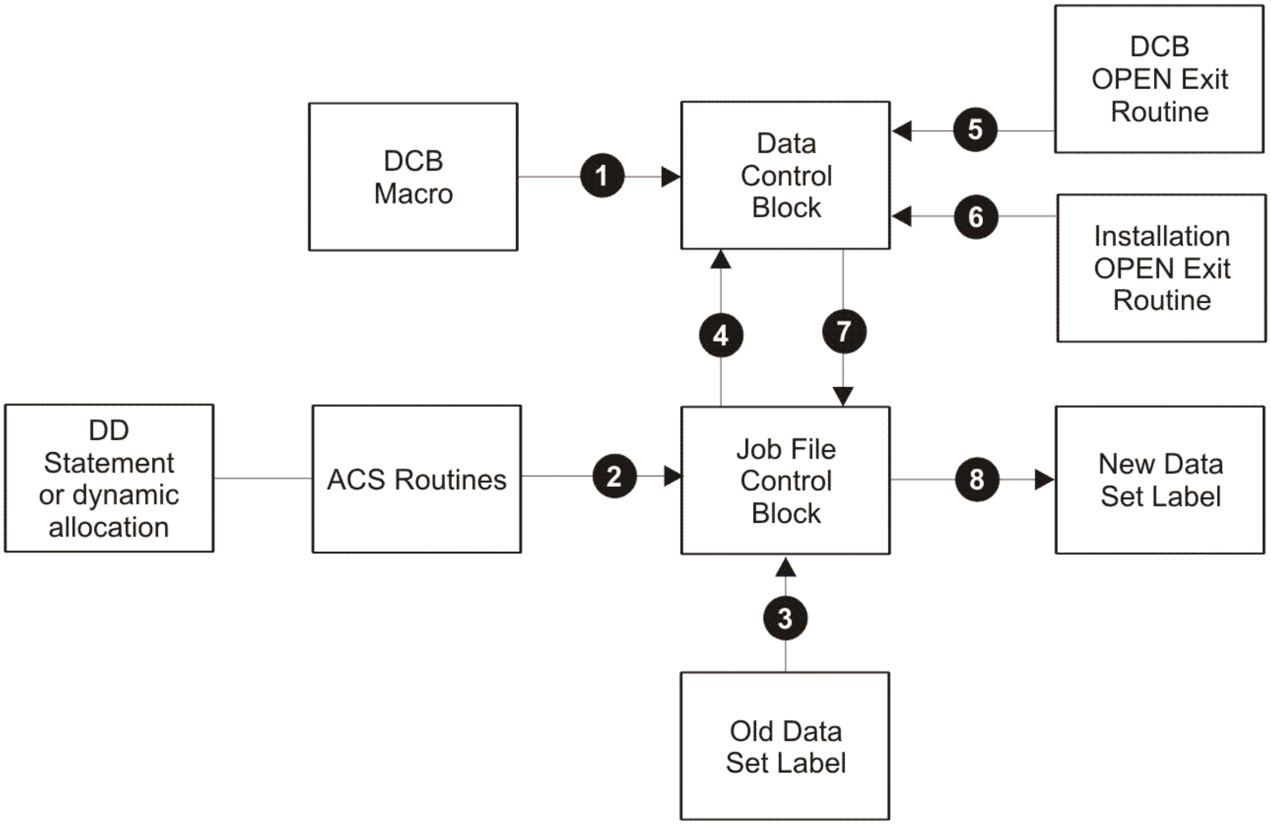


Figure 2. Sources and sequence for completing the data control block

Forward Merge (Steps 3 and 4): If the tape data set is being opened for INPUT, INOUT, RDBACK or EXTEND, or it is opened for OUTPUT in conjunction with DISP=MOD, then information from the standard data set label is merged into vacant fields of the JFCB. (Any fields that were already specified by the DD statement are not changed.) Then, for any OPEN option, information from the JFCB is merged into vacant fields of the DCB and DCBE. (Any fields that were already specified by the DCB and DCBE macro are not changed.) When the forward merge is completed, your processing program can use the DCB open user exit routine to modify the DCB or DCBE. For a description of the DCB open user exit routine, see *z/OS DFSMS Using Data Sets*. After the optional DCB OPEN user exit routine, the system calls the installation OPEN exit routine. For more information about installation exits, refer to *z/OS DFSMS Installation Exits*.

Reverse Merge (Steps 7 and 8): After the DCB is completed, the merging process is reversed. For an input data set, information from the DCB is used to fill in any vacant fields of the JFCB. For an output data set, the DCB information overrides the JFCB information (except the data set organization field), and the updated JFCB provides the information for creating the new labels.

Tape information recorded elsewhere

The operating system records and maintains information about tape volumes and uses that information to manage the volumes and data sets.

Data sets managed by Storage Management Subsystem

If you have the Storage Management Subsystem (SMS) active, you can move tape data sets to Direct Access Storage Device (DASD) to take advantage of the space, performance, and availability capabilities of SMS. If you put these tape data sets on DASD, you will have less tape activity, use fewer tapes, and need less operator intervention. DFSMShsm can move large numbers of appropriate data sets to tape much more efficiently than if users controlled it directly.

For more information about SMS, see *z/OS DFSMS Introduction*. Also, *z/OS DFSMSdfp Storage Administration* explains how to move tape data sets to DASD.

Data sets and volumes managed by DFSMSrmm

Data sets on magnetic tape and tape volumes can be managed by DFSMSrmm. DFSMSrmm provides tape management functions including validation, overwrite prevention, return to scratch, movement control between libraries and locations, and others. DFSMSrmm maintains this information in a control data set.

For more information about DFSMSrmm, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Cataloged data sets

The operating system has facilities that 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 and omitting the volume serial. The system finds the associated information in the catalog and issues a command, to either the operator or the tape library, to mount the cartridge on a drive.

Other tape information repositories

Operating system extensions such as DFSMShsm and DB2® maintain information about tapes that they write.

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 data 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.G0001V00. The fourth generation of the data set is identified as A.PAYROLL.G0004V00. The *absolute* generation and version numbers are in the form G*xxxx*V*yy*, where:

xxxx
: is a decimal number (0001 to 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 to 99) identifying a version of the same generation. Only the latest version is cataloged.

You usually refer to a generation data set 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.

These are methods to specify attributes for a new generation data set:

- When a generation data group base is established, the GDG is cataloged. The GDG can be cataloged in either an SMS or non-SMS managed catalog. If the catalog resides on an SMS managed volume, then the model DSCB can not exist. (All SMS data sets must be cataloged and the model DSCB has the same name as the GDG. Therefore the model DSCB can not be defined on an SMS volume because its name would conflict with the existing GDG name.) If the catalog resides on a non-SMS managed volume, then the model DSCB can exist; however, if the generation is SMS-managed, then it will not use the model

DSCB. The LIKE, REFDD, and DATACLAS attributes can be used for both SMS and non-SMS generation data sets.

- Only explicitly specified DCB attributes, REFDD, LIKE, and DATACLAS are used with SMS to allocate a new generation data set. If a model DSCB exists, it is ignored. However, model DSCBs are still valid. The LIKE keyword specifies the allocation attributes of a new data set by copying the attributes of a cataloged model data set. The following attributes are copied from the model data set to the new data set:

  – Record format (RECFM)
  – Record length (LRECL)
  – Data set organization (DSORG)
  – Record organization (RECORG) (to VSAM only)
  – Key length (KEYLEN) (to DASD only)
  – Key offset (KEYOFF) (to VSAM only)
  – Type, PDS, or PDSE (DSNTYPE) (to DASD only)
  – Space allocation (AVGREC and SPACE) (to DASD only)

  Examples of allocating a tape generation data set by supplying its DCB attributes through DATACLAS and LIKE are:

  ```
  //DDNAME DD DSN=HLQ.----.LLQ(+1),DISP=(NEW,CATLG),DATACLAS=dc_name

  //DDNAME DD DSN=HLQ.----.LLQ(+1),DISP=(NEW,CATLG),LIKE=dsname
  ```

If you do not specify a block size in the JCL, the system calculates a block size. Note that the block size is not obtained from the data class or LIKE data set. For further information, see *z/OS DFSMS Using Data Sets*. The expiration date from the data class is stored in the JFCB. Note that the expiration date from the LIKE data set is not used.

Information on creating and retrieving generation data groups can be found in *z/OS DFSMS Using Data Sets*, *z/OS MVS JCL User's Guide*, and *z/OS DFSMSdfp Utilities*.

Concatenated data sets

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

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 backwards.

Because only one DCB is associated with all the concatenated data sets, you must inform the system if the data sets have certain allocation characteristics that do not match (such as record format). To do this, your processing program must set the unlike-attributes bit in the DCB, as explained in *z/OS DFSMS Using Data Sets*. If the EXCP, POINT, or CNTRL macros are not used, you can concatenate tape, DASD, and sysin data sets if they have certain allocation characteristics that match. In this case, you do not have to set on the unlike-attributes bit in the DCB.

When standard labels, buffers gotten by OPEN, and the queued sequential access method (QSAM) are used together, the data set unlike characteristic of "block length" does not apply, and you can concatenate data sets in any order of block size. An exception is a tape block size that exceeds 32 760 bytes, but the application program does not have the BLKSIZE parameter on the DCBE macro. Normal DCB, JFCB, and label merging are performed to obtain the tape block size when labels are not present or are bypassed.

More information on concatenated data sets can be found in *z/OS DFSMS Using Data Sets*.

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 conveniently refer to the data set 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, and so forth) 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.

If you use a volume reference, all volumes must be in the same library or the request fails.

Multiple volumes and multiple data sets

To place a single data set on multiple volumes, code the multiple volume serial numbers in the related DD statement or request a *nonspecific volume*. To request a nonspecific volume or volumes, you omit volume serial numbers from the DD statement. This is also called a scratch volume request. If you request *specific volumes* and cataloging, all the specified volume serial numbers are associated with the new data set in the catalog. If you use fewer volumes than you specify, you cannot retrieve the data set properly using the catalog.

To place multiple data sets on a single volume, code the same volume serial number on each of the related DD statements or use the VOLUME=REF parameter on the DD statements for the second and subsequent data sets volume. When you create or retrieve a data set of file sequence greater than one, you must use the LABEL parameter to specify the data set's sequence number. This is not required if retrieval is accomplished through the catalog.

To place multiple data sets on multiple volumes, code a set of volume serial numbers on each of the related DD statements or use the VOLUME=REF parameter. This collection of volumes is called an *aggregate*.

Note: VOLUME=REF cannot be used if you specified volume serial numbers for the referred-to data set and that data set was written on fewer volumes.

If you code a set of volume serial numbers for each of the standard labelled data sets, the system works most efficiently if the first serial number is for the last volume occupied by the preceding data set. This is possible only if the volumes do not contain standard labels.

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 position of the data set relative to the entire multi-volume group. You can retrieve data sets in a different order than they were written. However, with standard labelled tape, you will get a warning message, IEC709I or IEC710I. For SL and AL, OPEN reads the existing labels to determine their position within the volume group. From existing labels, OPEN calculates the appropriate position for the data set being opened. For unlabeled tapes, the system cannot determine data set positions in an existing multi-volume group after the first volume. Therefore, you should not use the catalog to retrieve such data sets.

All data sets on a RACF-protected volume are protected. Data sets that span multiple volumes should have all volumes protected under a single RACF, a component of the Security Server for z/OS, tape volume set profile. If a data set on a RACF-protected volume is extended to a new volume that is not RACF-protected, the new volume is defined to RACF as part of the same volume set as the previous volume. Therefore, RACF protection is automatically extended to the new volume.

Processing methods and routines

You can specify the method of processing (INPUT, OUTPUT, EXTEND, RDBACK, INOUT, OUTIN, or OUTINX) by using an operand of the OPEN macro. If you do not specify the method, INPUT is assumed.

A data set can be processed as either input or output (INPUT, OUTPUT, or EXTEND). 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, or OUTINX). For INOUT, the data set is an input data set first and then, without reopening, an output data set. For OUTIN or OUTINX, the data set is an output data set first and then, without reopening, an input data set. INOUT is not supported for ISO and ANSI tapes except when allowed by the installation label validation exit.

Open/Close/End-of-Volume routines

The open/close/end-of-volume (O/E/EOV) routines of the system perform tape label processing.

Opening a data set

The open routine is entered when the processing program issues an OPEN call that is generated by a macro or by a compiler. The open routine completes the specified DCB, and prepares and positions the data set for processing. It analyzes input header labels (or trailer labels if the tape is read backwards), or creates output header labels.

The open routines also resolve many possible conflicts, such as specifying a label type that differs from the type on the tape or the tape drive not being able to read due to the tape technology being incompatible. Tape conflict resolution can be handled by DFSMSrmm, tape management installation exit and by Open and EOV label editor installation exits. See "IBM standard volume label editor routines" on page 37 and "ISO/ANSI volume label verification and editor routines" on page 79.

If DFSMSrmm is in use, additional tape volume validation is performed and information recorded in the DFSMSrmm control data set. For multifile tapes, if using DFSMSrmm, the positioning of standard labeled tape volumes is much faster than normal label positioning during OPEN processing. For more information on high-speed-positioning, see *z/OS DFSMS Installation Exits*.

Closing a data set

The close routine is entered when the processing program issues a CLOSE macro. If the processing program terminates without closing the data set, the operating system calls the close routine, which restores the fields of the DCB to the conditions that existed before the data set was opened. It also logically disconnects the data set from the processing program, and creates output trailer labels, and provides for tape disposition. If DFSMSrmm is in use, statistics about processing are recorded in the DFSMSrmm control data set.

End of data set or volume

The EOV routine is entered when a tape mark is read, when the end of volume is encountered while writing, or when the processing program issues a force-end-of-volume (FEOV) macro. If you use the execute channel program (EXCP) technique, your processing program must issue an EOV macro to give control to the EOV routine after your program recognizes a tape mark or end of reel or cartridge. The EOV routine processes trailer labels on the current volume (or header labels if the tape is read backwards), and determines if additional volumes are needed to continue the data set. If another volume is needed, or another data set is concatenated, then 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, then EOV gives control to the user's end-of-data routine that is specified in the DCB.

EOV routines also resolve conflicts in the same way as the open routines.

If DFSMSrmm is in use, then the function performed at OPEN and CLOSE is performed at EOV as well.

Checkpoint/Restart routine

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 DCBs to the conditions that existed when the checkpoint was taken.

The restart routine can handle tapes with IBM standard labels, or no labels.

Note: Tapes with ISO/ANSI labels are not supported by checkpoint/restart as checkpoint volumes.

All VSE/ESA® tapes having either a leading tape mark or embedded checkpoint records can be handled by checkpoint/restart, except for VSE/ESA 7-track tapes written in translate mode that contain embedded checkpoint records.

Automatic volume recognition

Under automatic volume recognition (AVR), 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 when that volume is requested.

Note: If the drive is needed earlier for another request, then the volume will be unloaded.

AVR checks the tape label during allocation by the scheduler, and records the volume serial number. This action merely determines which volumes are mounted on which devices; AVR does not verify or reject the volumes based on their serial numbers. The following conditions cause the volume to be unloaded:

- The volume is unlabeled.
- The volume has a nonstandard label and either there was no installation exit or the installation exit rejected the volume.
- The volume serial number is a duplicate of one that is already mounted.

In an automated tape library, a mismatch between the external volume label and the tape volume label causes an error. When AVR detects an error, it causes the volume to be unloaded with a message identifying the error.

Tape disposition

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

The positioning parameters of the OPEN, FEOV, 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. REREAD cannot be specified for FEOV.

REWIND
Rewind the volume to the load point. REWIND cannot be specified for OPEN or CLOSE TYPE=T.

DISP
Perform the disposition processing that was requested. This can be LEAVE, REWIND, or REWIND and UNLOAD, depending on the volume attributes. DISP cannot be specified for FEOV.

The CLOSE macro has the following additional operand that allows you to release tape data sets and the volumes on which they reside:

FREE
Release the data set associated with this DCB. FREE specifies that the data set is to be released for use by another job and that the device on which the data set is mounted can be freed for allocation to another job.

DISP is the default.

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

Verifying block count

When a program finishes reading the portion of a data set that is on a tape reel or cartridge, the system checks that the expected number of blocks was read. The system does this if the volume contains IBM standard or ISO/ANSI labels and the program uses BSAM, QSAM, or EXCP. With EXCP, the program maintains the block count in the DCB and the programmer codes REPOS=Y on the DCB macro. For more information, see "Verifying block count in label when reading" on page 42, and "Verifying block count in label when reading" on page 91.

When a program finishes reading or writing data on a cartridge (not a reel), the system does additional block count checking unless the program is using EXCP and the programmer did not code REPOS=Y on the DCB macro. The system can check for this on all cartridges, including unlabeled tapes. This type of block count checking is independent of the checking that is done only on labeled tapes.

The additional checking is to compare the count accumulated in the data control block with a count that the system calculates based on information maintained by the tape subsystem (hardware). If the two counts differ, the system does not call the block-count-unequal exit. Instead, the system calls the optional DCB ABEND exit for abnormal completion code 117 or 237. If the exit is not defined, or if it does not ignore or recover, then the system issues an ABEND macro. For more information on DCB ABEND exit, see *z/OS DFSMS Using Data Sets*. For more information on the installation DCB ABEND exit, see *z/OS DFSMS Installation Exits*.

For more information on maintaining and verifying block count when using EXCP, refer to *z/OS DFSMSdfp Advanced Services*.

Bypassing label processing

If you do not want the system to verify the label type, specify BLP (instead of NL) in the LABEL parameter of the DD statement. A loadpoint read will be done to check for existing labels. If labels are present, the system will still accept the tape as if NL were specified.

There are other reasons for using the BLP option. For example, you may want to overwrite a 7-track tape that differs from your current parity or density specifications. If such a tape is mounted, the system makes four attempts to read the initial record (to determine that it is not an IBM standard label) before accepting the tape. Each read may result in a long error recovery attempt. You can eliminate the four read operations by specifying BLP instead of NL.

There are some situations where the system will detect that a particular tape may not be the correct volume. Your job may have requested a specific volume or a non-specific volume. If your job requests that label processing be bypassed in these cases, the system demounts the existing tape and asks for a specific or non-specific volume. This action gives the tape operator the chance to insure that the requested volume is mounted.

When a job requests that label processing be bypassed, the OPEN or EOV routines will demount the currently mounted tape if the drive is in a ready state and any of the following is true:

- The tape has been processed by automatic volume recognition (AVR)
- The tape has been processed by a MOUNT command
- The mounted volume has been passed or retained from a prior step of the same job
- The mounted volume is still mounted after having been processed in the same step

To use the BLP option, the user is responsible for proper tape positioning because Open/Close/EOV cannot determine if the tapes are labeled or unlabeled. BLP is identical to NL processing, except that the check for an existing label is bypassed. If the tape has standard labels, you code LABEL=(2,BLP) to read the first data set and LABEL=(5,BLP) to read the second data set. A BLP positions second and subsequent volumes of a multivolume data set as if it were unlabeled. This positioning is incorrect if the data set has labels.

To use BLP to read a multivolume standard labelled data set you can specify each volume on its own DD statement. Leave the *ddname* field blank for each volume after the first so they are treated as concatenated data sets. You can code LABEL=(2,BLP) on each DD statement.

If BLP is requested under DFSMSrmm, then DFSMSrmm checks to see if the volume is labeled. If the volume and the label do not match, then the requested volume is rejected.

Controlling the use of BLP

The BLP option is associated with job class; that is, the installation has the option to allow or disallow BLP by job class. It is recommended that BLP be disallowed completely, or if this is not possible, that one job class be set aside as the "BLP" and controlling class. This can be done by using JES initialization options to allow BLP for only that controlling class and by using installation JCL exits to force TYPERUN=HOLD for all jobs specifying that class. The system operator can then monitor that class and release only jobs authorized to use BLP. For the method of specifying BLP for the JES reader, see *z/OS JES2 Initialization and Tuning Reference*.

Resource Access Control Facility (RACF), a component of the secureway security server for z/OS, protection and BLP JCL

To use Resource Access Control Facility (RACF) for tape volumes or data sets, the BLP JCL option must be disallowed completely, or restricted to authorized users.

A system programmer can specify that RACF is to control which users can use the BLP option. This is done by activating the TAPEVOL class and defining profile ICHBLP in the FACILITY class. For further information see *z/OS Security Server RACF Security Administrator's Guide*.

For more information on additional controls over some BLP usage, see the *z/OS DFSMSrmm Implementation and Customization Guide*.

Tape characteristics

The following paragraphs describe the data recording characteristics of magnetic tape. The discussion includes density, parity, number of tracks, translation, conversion, and tape marks. The error conditions that can result from conflicting tape characteristics are explained under "IBM standard volume label editor routines" on page 37, "ISO/ANSI volume label verification and editor routines" on page 79, and " Nonstandard volume label verification and editor routines" on page 101.

z/OS supports the following devices:

- IBM 3420 Magnetic Tape Unit Models 3, 4, 5, 6, 7, and 8
- IBM 3422 Magnetic Tape Subsystem
- IBM 3424 Magnetic Tape Unit (available only in Brazil, S.A.)
- IBM 3430 Magnetic Tape Subsystem
- IBM 3480 Magnetic Tape Subsystem
- IBM 3490 Magnetic Tape Subsystem
- IBM Virtual Tape Server
- IBM 3590 High Performance Tape Subsystem
- IBM System Storage 3592 Tape Subsystem

Nine-track tapes and dual density feature

The operating system supports 9-track tape reels in densities of 800 bits per inch (bpi), 1600 bpi, and 6250 bpi.

The dual-density feature, available on some tape units, permits the unit to read and write in either of two densities. The density combinations available are 800/1600 bpi and 1600/6250 bpi. You specify the density with the DEN parameter of the DD statement or DCB macro instruction (Table 1 on page 12 shows the DEN parameter codes). If the DEN parameter code is not specified, the greater density is assumed.

Table 1. DEN parameter codes for specifying tape density

DEN Value“1” on page 12	**Recording Density**	
	7-track	**9-track**
1	556	—
2	800	800 (NRZI)“2” on page 12
3	—	1600 (PE)“3” on page 12
4	—	6250 (GCR)“4” on page 12

Notes:

1. If the DEN parameter is not supplied by any source, the highest applicable density is assumed.
2. Nonreturn-to-zero inverse mode
3. Phase-encoded mode
4. Group-coded recording mode

The DEN parameter is ignored for input. The system automatically sets the tape unit to the density of the tape.

For output with dual density, the tape is written in the density you specify or, if unspecified, the default density. If your request is for a standard labeled tape, and the label of the mounted volume is written in the wrong density, the system will rewrite the label to agree with your specification, if you are opening the first data set on the volume. If this is not the first data set on the volume, the system will change your density specification to agree with the density of the volume. Tapes created with the dual-density feature and those created without it are interchangeable.

Seven-Track tapes

The 7-track feature allows a tape unit to read and write 7-track tapes on reels. This special feature consists of a 7-track read/write head (instead of a 9-track head) and control unit changes, including a translator. Data can be read or written in densities of 556 or 800 bpi with either odd or even parity. Nine-track tapes cannot be read on tape units with the 7-track feature installed. The translator takes 8-bit EBCDIC characters from your buffer and writes them as 6-bit binary coded decimal interchange code (BCDIC) tape characters and translates the opposite way during a read operation. 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 6 low-order bits of the characters in your buffer are written on tape; during reading, the 2 high-order bits are set to zeros.

ISO and ANSI standards do not include a specification of 7-track magnetic tape for information interchange. Therefore, the 7-track feature is not applicable for tapes with ISO or ANSI labels.

The data conversion feature can also be installed with the 7-track feature. The data conversion feature makes it possible to write binary data on 7-track tape. It writes 3 characters from your buffer as 4 tape characters, and converts the opposite way during reading. Conversion is turned on and off by mode setting control commands and is mutually exclusive with translation. You must use the data conversion feature to process format-V (variable-length) tape records because the length field of such records contains binary data. You cannot use the data conversion feature with the read backward (RDBACK) processing method.

The operating system supports the various densities of the 7-track feature. You specify the density with the DEN parameter of the DD statement or DCB macro. Table 1 on page 12 shows the DEN parameter codes. If not specified, the default value is 800 bpi.

The job step is abnormally terminated if the DEN parameter specifies a density incompatible with the tape unit.

Mode information other than density is specified with the tape recording technique (TRTCH) parameter of the DD statement or DCB macro instruction. The codes for the TRTCH parameters are listed in Table 2 on page 13.

Table 2. Codes for the TRTCH parameter

Code	Meaning
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 9-track)

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

Nonstandard labels on 7-track tapes 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 tape marks should be carefully planned. System recognition of tape marks is ensured only when they are read in the density in which they were written.

Tape cartridges

The operating system supports several IBM magnetic tape subsystems, such as the IBM 3480, the 3490 Magnetic Tape Subsystems, the 3590 High Performance Tape Subsystem, the IBM System Storage 3592 Tape Subsystem, as well as virtual tape, all of which use tape cartridges, instead of tape reels. For these subsystems, only a single density is available for reading and writing; any density specified with the DEN parameter is ignored.

Compaction

A magnetic tape subsystem that uses tape cartridges and has compaction capabilities can record data in either of two formats: compacted format or standard, uncompacted format. The recording format is specified in the TRTCH parameter of the JCL DD statement, the dynamic allocation text units, or the DCB macro. If none of these sources specify the recording format when writing the first data set on a volume, OPEN uses the specification in the DEVSUP*xx* member in SYS1.PARMLIB, which is set by the system programmer. Table 3 on page 13 lists the parameters.

Table 3. TRTCH parameters

Parameter	Meaning
COMP	Record data in compacted format
NOCOMP	Record data in standard, uncompacted format
Null (entire parameter is omitted)	Record data in default format

Recording technology

Cartridge cases have about the same physical dimensions, but the physical tapes and the recording technologies differ. In the following cases, the drive is not capable of reading existing data, but it can read an existing volume serial number and the drive allows software to write over the data with a different recording technology, for example:

- 36-track tape mounted on a drive that is only capable of handling 18-track tapes
- 256-track tape mounted on a drive that is only capable of handling 128-track tapes
- EFMT2/EEFMT2-track tape mounted on a drive that is only capable of handling EFMT1-track tapes

In these cases the open or EOV functions can retrieve the volume identifier. When the first block is written on the tape and as an IBM standard or ISO/ANSI standard volume label, the tape subsystem extracts the volume serial number and saves it in an area accessible only to the drive. When Open or EOV functions later issue a read for the volume serial number on specific drives that cannot read at the recording technology, the I/O is unit-checked. If the device supports the 'vol in sense' capability, the VOLID is retrieved and inserted into the sense that is returned on the unit-checked I/O. If VOLSNS=YES is coded in the active DEVSUP*xx* member, Open/EOV will pick up the VOLID from the returned sense.

Note: This will only work between specific devices that support this functionality, for example:

- between an 18 track device attempting to read the first block on media recorded on a 36 track device, or
- on a 128 track device attempting to read the first bock on media recorded on a 256 track device
- on an EFMT1 track device attempting to read the first block on media recorded on an EFMT2/EEFMT2 capable device

The OPEN or EOV functions rewrite the tape with the drive's recording technology, possibly without standard labels, if all of the following are true:

- The application program is opening to write to the first data set
- The volume serial number is available from the vol ID mark or from the external label from the vision system
- DFSMSrmm, the label anomaly installation exit routine, or the volume label editor installation exit routine allows the rewrite
- The user has RACF UPDATE authority to the volume or RACF is not controlling tape security
- The DEVSUP*xx* member of SYS1.PARMLIB specifies VOLSNS=YES

Tape marks

A data set or label group on tape is usually followed by a tape mark delimiter. A tape mark is a special character written by a control command. (In the figures in this manual, tape mark is represented as TM.) The tape drive recognizes a tape mark during a read operation and signals a unit exception condition to the operating system. The tape mark is not read into virtual storage.

Beginning and End of tape

On 7-track and 9-track tape units, a reflective strip at the beginning of a tape indicates when the tape is positioned at its *load point*; a load point indicator bit is set in a sense byte, and is recognized by the operating system. A reflective strip also marks the logical end of the tape. If a reflective strip is encountered during a write operation, the hardware signals a unit exception condition in the interrupt to the processor.

IBM magnetic tape subsystems such as the IBM 3490, 3590, and 3592 use tape cartridges and not tape reels. Subsystems that use tape cartridges have internal mechanisms that sense the beginning and end of tape, and do not use reflective strips. The effect on software is the same as with reels.

Chapter 2. IBM standard labels

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

This chapter describes the organization, formats, and contents of IBM standard labels and IBM standard user labels. It also explains how the labels are processed or created, and the use of these labels in processing data sets.

Label definitions and organization

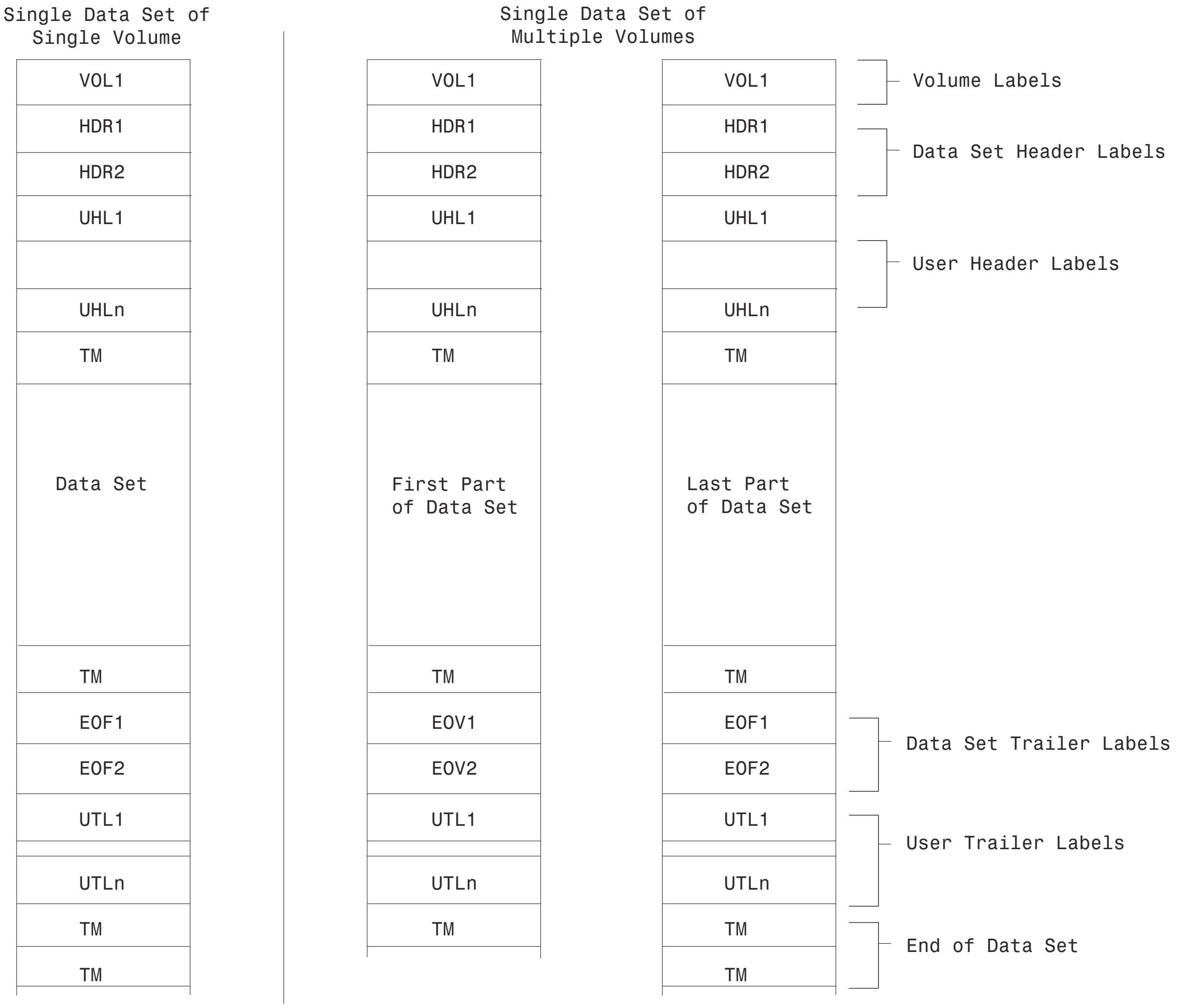
IBM standard labels are 80-character records written in the density specified using JCL. Cartridge and 9-track tape label records are recorded in EBCDIC. Seven-track tape label records are recorded in BCDIC, even parity, translate on. The first 4 characters are always used to identify the labels:

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

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

- Standard volume label (identified as VOL1)
- Standard data set label 1 (identified as HDR1, EOV1, or EOF1)
- Standard data set label 2 (identified as HDR2, EOV2, or EOF2)
- Standard user label (identified as UHL1-UHL8 or UTL1-UTL8)

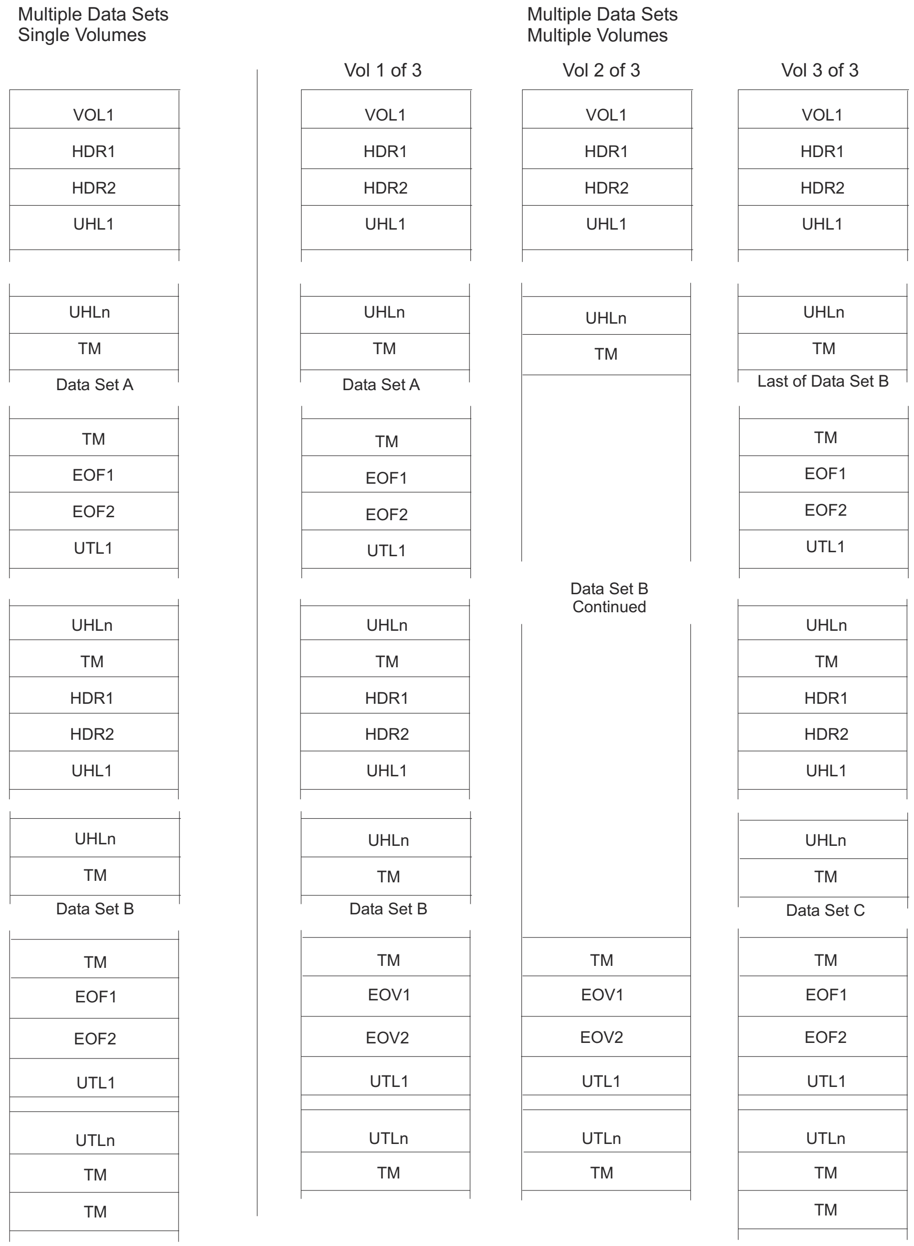
Figure 3 on page 16 and Figure 4 on page 17 show the positions of the labels with various tape volume organizations. A tape with IBM standard labels must contain a volume label and data set labels. User labels are optional.



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 IBM 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: This is called aggregate. More than one volume is needed to contain the data sets. Each volume begins with a volume label. The last volume is organized the same as a multiple data set/single volume layout, except that the first data set begins on an earlier volume. On the other volumes, the data set trailer labels at the end of the tape are identified as EOV instead of EOF, and the last trailer label group is followed by only one tape mark.

Figure 4. Volume organizations with IBM standard labels (multiple data sets)

Volume label

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

Volume labels are created by the EDGINERS or IEHINITT utility programs, or by input from the operator during the Open or EOV routine when:

- A label conflict is detected (a label conflict occurs when the type of label requested differs from the type of label read at load point from the mounted volume)
- The first block of a volume cannot be read (except a RACF-protected volume is rejected if failure to read was because the tape format, such as a nonsupported density, was not recognizable by the control unit)

For additional information about EDGINERS and the IEHINITT utility programs, see *z/OS DFSMSrmm Implementation and Customization Guide* and *z/OS DFSMSdfp Utilities*.

Data set header labels

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

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

User header labels

Optionally, a maximum of eight user header labels (UHL1 to 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 the system to write user header labels or to make user header labels available to your program, you must specify SUL on the DD statement and specify the address of a user header label routine in the DCB exit list. (The exit list can address several user header label routines, that is, routines that process input user header labels and create output user header labels.) The DCB exit list is described in *z/OS DFSMS Using Data Sets*.

Data set Trailer labels

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

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

These labels are created automatically by the system 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 the system to write user trailer labels or to make user trailer labels available to your program, you must specify SUL on the LABEL parameter of the DD statement and specify the address of a user trailer label routine in the DCB exit list. The exit list can address several user trailer label routines, that is, routines that process input user trailer labels and create output user trailer labels. The DCB exit list (EXLST) is described in *z/OS DFSMS Using Data Sets*.

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, and so forth. If such labels exist on an input tape, they are bypassed. They are omitted on output tapes.

System-Managed mountable tape external labels

An external label is required for every tape cartridge in the tape library. This label contains both machine-readable and operator-readable identification characters that are used by the tape library vision system. A tape library vision system, which is unique to an automated tape library, identifies the volume serial number of the cartridge. In a manual tape library, a machine readable external label is not required, but is advisable for migration purposes. Each label can have a maximum of six characters for identification purposes:

- Uppercase A - Z
- Numerics 0 - 9
- Trailing blank spaces (non-service volumes only)

The external label must be applied within the indentation on the side of the cartridge; however, a larger information label, supplied by the customer, can be used on the top of the cartridge. See Appendix A, "Using the IBM Tape Library," on page 113 for more information.

Tape marks

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

- There is no tape mark between the volume label and the first header label group on the volume.
- The tape mark that marks the end of the header label group also indicates the beginning of the data set to be processed.
- The tape mark that follows the data set also indicates the beginning of the trailer label group.
- A tape mark marks the end of the trailer label group. A second tape mark follows the trailer label group of the last data set on the volume, provided that the data set does not continue on another volume.

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

IBM standard labels format

Format of the IBM standard volume label (VOL1)

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

Figure 5 on page 20 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)

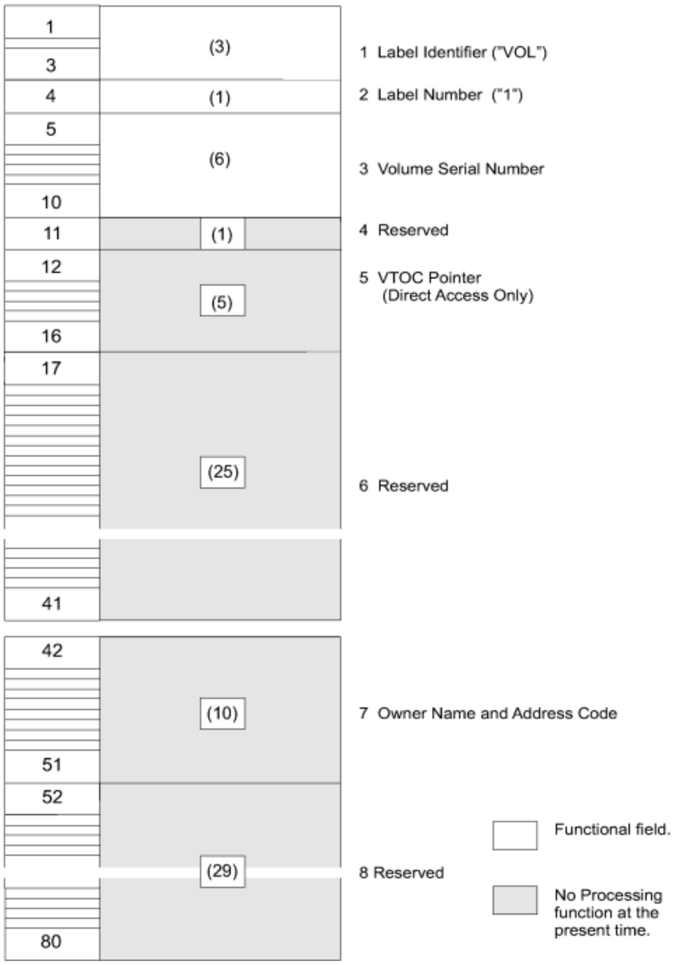


Figure 5. Format of IBM standard volume label

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.

  The labels can be initially created by the IEHINITT utility program, the DFSMSrmm-supplied EDGINERS utility program, and under certain circumstances during Open/EOV processing.

  For standard labeled tape volume requests, DFSMSrmm or the label anomaly installation exit routine or label editor installation exit can give permission for the OPEN or EOV routines to write or rewrite the VOL1 label. See "IBM standard volume label editor routines" on page 37 for more information.

2—Label Number (1 byte)

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

3—Volume Serial Number (6 bytes)

- *Contents*: A unique identification code that is assigned through DFSMSrmm or IEHINITT to the volume when it enters the system, or that is assigned by the operator when the Open or EOV routines label the volume. This code may also appear on the external surface of the volume for visual identification. The code is normally numeric characters (000000 to 999999), but may be any 6 alphanumeric characters. It may be from 1 to 6 characters, but, if fewer than 6 characters, the code must be left-justified, and the remainder is padded with blanks.

If the volume serial number is specified in the JCL statements, all national characters, the hyphen, and other special characters are accepted when enclosed in apostrophes. Their use is not recommended, however, because it can be difficult to recognize volume serial numbers when typewriter heads, print chains, and screen fonts with nonalphanumeric characters are used.

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

In a tape library there is an additional restriction. See "System-Managed mountable tape external labels" on page 19.

- *Processing*: When the user specifies a volume serial number, it is in the JFCB. The system copies it to the UCB. When the system processes the labels, it compares the number in the UCB 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 EDGINERS and IEHINITT utility programs can create this label with a volume serial number of up to 6 characters. The number is left-justified, and the remainder of this field is padded with blanks.

4—Reserved (1 byte)

- *Contents*: Reserved for possible future use—initialized as a blank or zero.
- *Processing*: Not used.

5—VTOC Pointer (5 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 EDGINERS and IEHINITT utility programs write blanks in this field.

6—Reserved (25 bytes)

- *Contents*: Reserved for possible future use—should be recorded as blanks. The EDGINERS and IEHINITT utility programs write blanks in this field. The IEHINITT utility program writes blanks in this field.

  Historical note: In certain levels of MVS/DFP™ Version 3, position 36 may have been written as "P" to signify that the data was compacted using IDRC. The system no longer does this.

7—Owner Name and Address Code (10 bytes)

- *Contents*: Indicates a specific customer, person, installation, department, and so forth, to which the volume belongs. Any code or name is acceptable.
- *Processing*: Not used or verified. The EDGINERS and IEHINITT utility programs write the text specified by the user, and the open and EOV routines write the text specified by the operator. If the code is less than 10 bytes, it is left-justified and the remainder of the field is padded with blanks. EDGINERS and IEHINITT allows EBCDIC alphanumeric and special characters.

8—Reserved (29 bytes)

- *Contents*: Reserved for possible future use—should be recorded as blanks.
- *Processing*: Not used or verified. The EDGINERS and IEHINITT utility programs write blanks in this field.

Format of IBM standard data set label 1 (HDR1/EOV1/EOF1)

IBM standard data set label 1 is 80 characters in length and describes the associated data set. The format is used for header labels (HDR1), end-of-volume trailer labels (EOV1), and end-of-data-set trailer labels

(EOF1). Data set label 1 is always followed by data set label 2 when the system is writing labels. Data set label 1 is recorded in EBCDIC on Cartridge and 9-track tape units, or in BCDIC on 7-track tape units.

Figure 6 on page 22 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)

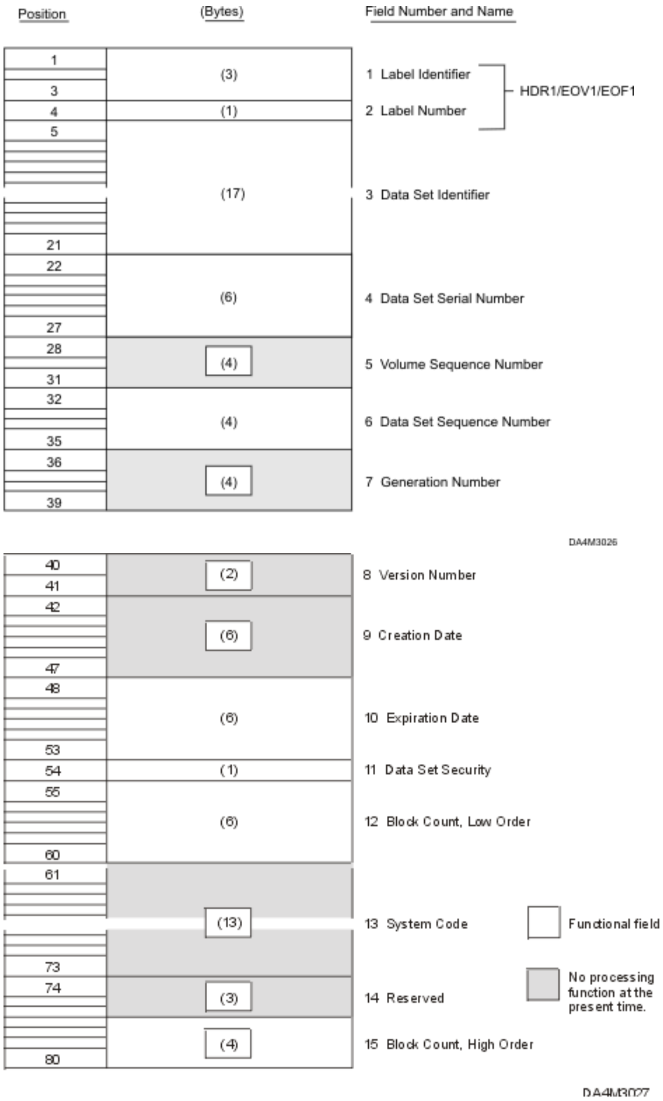


Figure 6. Format of IBM standard data set label 1

1—Label Identifier (3 bytes)

- *Contents*: Three characters that identify the label are as follows:
  - **HDR**
    Header label (at the beginning of a data set)

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

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

- *Processing*: The system checks this field to verify that the record is an IBM standard data set label.

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

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

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

2—Label Number (1 byte)

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

3—Data Set Identifier (17 bytes)

- *Contents*: The rightmost 17 bytes of the data set name (includes G*xxxx*V*yy* if the data set is part of a generation data group). If the data set name is less than 17 bytes, it is left-justified and the remainder of this field is padded with blanks. If the name contains embedded blanks or other special characters, you must enclose the name in apostrophes on the DD statement that requests this data set. *z/OS MVS JCL User's Guide* 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 password protection to determine whether the existing data set can be overwritten. If protection is not specified, the data set name is not checked.

  For input and output, when DFSMSrmm is used, the first file data set identifier is compared with the last 17 characters of the first file data set name recorded in the DFSMSrmm control data set. If there is no match, then the volume is rejected. This check ensures that the correct volume is mounted.

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

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

4—Data Set Serial Number (6 bytes)

- *Contents*: The volume serial number of the tape volume containing the data set. For multivolume data sets, this field contains the serial number of the first volume of the aggregate. The content requirements are the same as for the volume serial number field in the volume label.
- *Processing*: If this field has inconsistent values in the volumes in a volume set as described in Figure 3 on page 16 and Figure 4 on page 17, the system writes a warning message. This warns that one of the volumes might be the wrong volume. The message number is IEC709I. 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 to 9999) that indicates the order of the volume within the multivolume aggregate. This number is always 0001 for a single volume data set.
- *Processing*: If this field does not have a value one greater than the previous volume in a volume set as described in Figure 3 on page 16 and Figure 4 on page 17, the system writes a warning message IEC709I. This warns the volumes might be out of order. When creating labels, the open routine writes 0001 in this field if the data set sequence number is 1; otherwise, the open routine copies the corresponding value from an earlier label on the volume. The EOV and close routines write a number that is one higher than on the previous volume.

6—Data Set Sequence Number (4 bytes)

- *Contents*: A number (0001 to 65535) that indicates the relative position of the data set within a multiple data set group (aggregate). This number is always 0001 for a single data set organization. If the first byte is an EBCDIC numeric (character "0" to "9"), then the four bytes are the character representation of a number from 0001 to 9999. If the first byte is a question mark ("?"), then the remaining three bytes contain a number in binary. The maximum value that z/OS currently supports in these three bytes is X'00FFFF', which is 65535.
- *Processing*: This number in the first HDR1 label on the tape is referred to when the open routine positions the tape. If this number in the first HDR1 label and the requested data set sequence number in the JFCB are both greater than 1, the logical data set sequence number in the UCB is set to the number in the label. Otherwise, the logical data set sequence number in the UCB is set to 1.

  When creating labels, the open and close routines obtain the user-specified data set sequence number from the JFCB (a 0 is changed to 1). The EOV routine propagates the number.

7—Generation Number (4 bytes)

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

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

9—Creation Date (6 bytes)

- *Contents*: Date when the data set was created. The date is shown in the format cyyddd, where:

  ```
    c = century (blank=19; 0=20; 1=21; etc.)
   yy = year (00-99)
  ddd = day (001-366)
  ```

  Note: The `century` code gives the first two digits of the year, not the actual century. For example, a blank which translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- *Processing*: Not verified. When the system creates labels, the date is obtained from the JFCB. If a data set is allocated using JCL, the creation date is the date when allocation begins for the job step

responsible for creating the data set. If a data set is allocated dynamically, the creation date is the date when allocation begins for the data set.

10—Expiration Date (6 bytes)

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

```
  c = century (blank=19; 0=20; 1=21; etc.)
 yy = year (00-99)
ddd = day (001-366)
```

  Note: The `century` code gives the first two digits of the year, not the actual century. For example, a blank that translates into 19 indicates a year in the 1900s, not in the nineteenth century.

- *Processing*: For input, not used or verified. For output, the expiration date in the existing label is compared to the current date. If the date in the label is greater than the current date, the operator receives a message and is given the option of using the tape or mounting another. Other data sets that follow on the same volume expire on the same day.

  If you enter 99365 or 99366, the system permanently retains your data sets, even after 1999. Do not use those dates as expiration dates. Use them as no-scratch dates only. On a tape written on OS/400, 99999 may be specified as a no-scratch date. z/OS does not honor that as a no-scratch date on tapes.

  The expiration dates of scratch SMS-managed tape volumes, in both automatic and manual tape libraries are ignored. DFSMSrmm can override the expiration date or reject the volume without the operator's intervention.

  When creating labels, the system obtains the expiration date from the JFCB. If you do not specify a retention period or expiration date, the expiration date is recorded as zeros, and the data set expires.

  DFSMSrmm uses the expiration date to update the volume information in its control data set. For data sets without an expiration date, DFSMSrmm assigns one using the DFSMSrmm default retention period, but the label field is left as zeros.

11—Data Set Security (1 byte)

- *Contents*: A code number indicating the security status of the data set is as follows:
  - **0**
    No password protection

    1
    Password protection. Additional identification of the data set is required before it can be read, written, or deleted. The password is ignored if the volume is RACF defined.

    3
    Password protection. Additional identification of the data set is required before it can be written or deleted. The password is ignored if the volume is RACF defined.

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

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

  When the second or later data set on a volume is created, and there is no HDR1 label with which to determine security protection, open reads the EOF1 label of the preceding data set on the volume. The data set security level in the EOF1 label must match the security level requested for the new data set. If they are not equal, the system abnormally terminates the task. If the security levels are equal and indicate no security protection, open processing continues. If security protection is indicated, the system requests a password and verifies that the password furnished by the operator allows access to the data set name in the EOF1 label, the preceding data set. It is important to note that, in this case, only the 17-byte data set name in the EOF1 label is available. Therefore, either the data set name must be 17 or fewer characters in length, or the last significant 17 characters of the full data set name must be entered in the PASSWORD data set. It is recommended that password-protected tape data sets limit their data set names to 17 or fewer characters, or that the last 17 characters of the data set name be entered in the password data set together with the full data set name.

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

12—Block Count, Low Order (6 bytes)

- *Contents*: This field in the trailer label shows the low order six digits of the number of data blocks in the data set on the current volume. This field in the header label always contains zeros (000000).
- *Processing*: The DCB count is increased as the data set is read. The final DCB count is compared with the count in fields 12 and 15 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. A block count discrepancy causes processing to abnormally terminate if the appropriate user exit entry is not provided.

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

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

  If the data set was created with an EXCP DCB that had no device-dependent section, the block count is written as zero and is not verified.

13—System Code (13 bytes)

- *Contents*: A unique code that identifies the system: ‘IBM OS/VS 370’.
- *Processing*: Not used or verified.

14—Reserved (3 bytes)

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

15—Block Count, High Order (4 bytes)

- *Contents*: High order four digits of a ten-digit block count. This field in a header label always contains blanks. If the data set contains more than 999999 blocks on the volume, this field contains up to four higher order digits. Of these four digits, the system changes high order zeroes to blanks when creating the trailer label.
- *Processing*: See field 12.

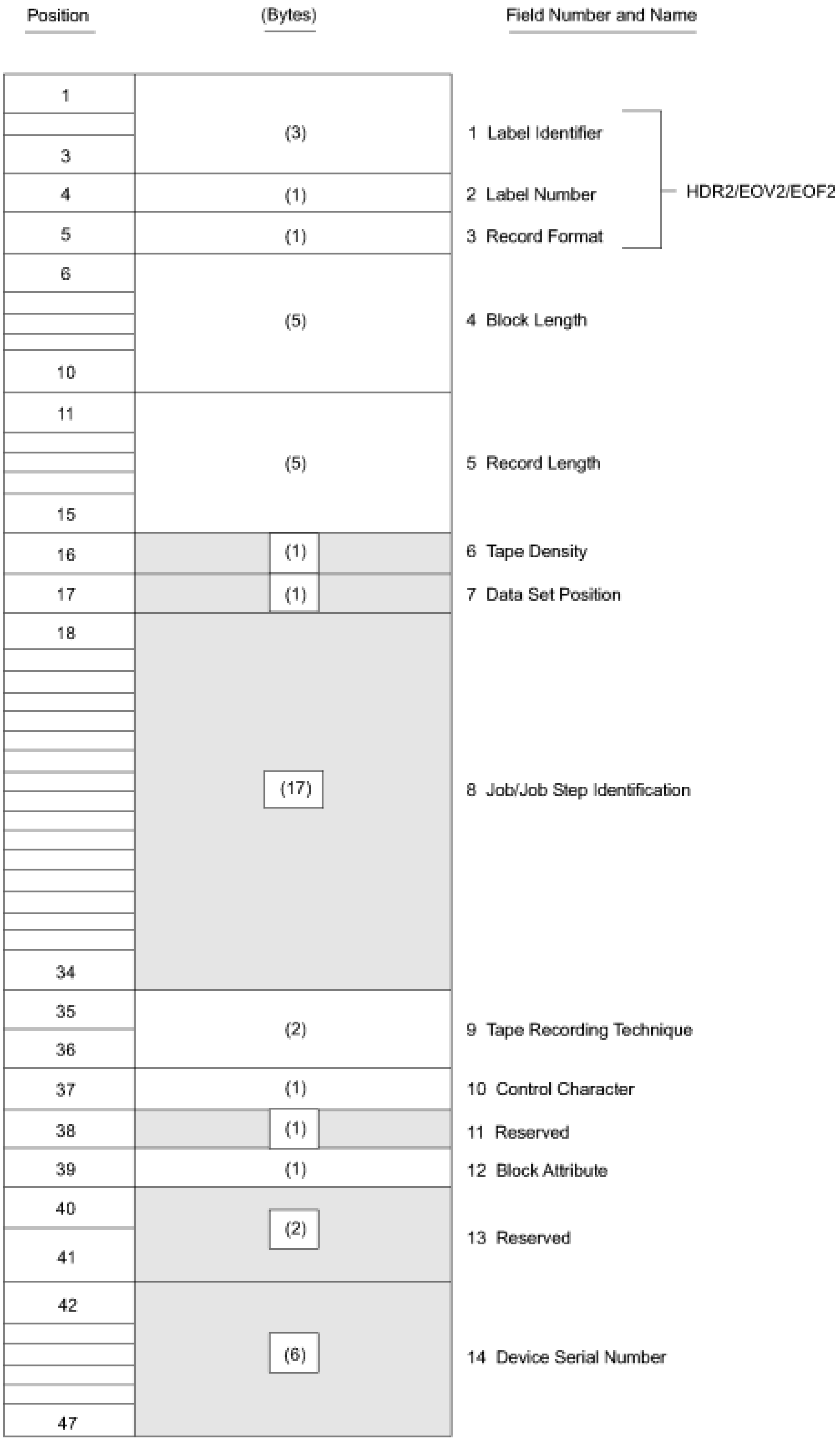
Format of IBM standard data set label 2 (HDR2/EOV2/EOF2)

IBM standard data set label 2 always follows data set label 1 and contains additional information about the associated data set. The format is used for header labels (HDR2), end-of-volume trailer labels (EOV2), and end-of-data-set trailer labels (EOF2). The label is 80 characters in length. It is recorded in EBCDIC on 9-track tape units and on cartridges, or in BCDIC on 7-track tape units. Tapes created on other operating systems such as VSE might not contain header label 2.

Figure 7 on page 27 shows the format of data set label 2. The shaded areas represent fields that the operating system writes in the label, but that are not used or verified during processing. The processing descriptions refer to the following system control blocks:

- Data control block (DCB)

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



Position	(Bytes)	Field Number and Name
1 3	(3)	1 Label Identifier
4	(1)	2 Label Number
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 Control Character
38	(1)	11 Reserved
39	(1)	12 Block Attribute
40 41	(2)	13 Reserved
42 47	(6)	14 Device Serial Number

Figure 7. Format of the IBM standard data set label 2

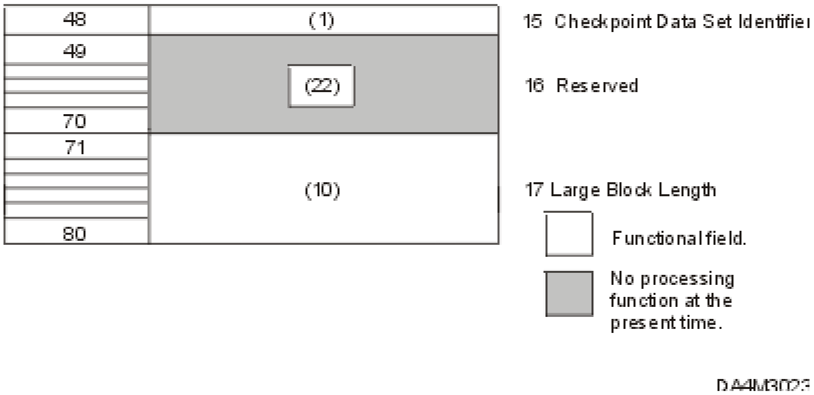


Figure 8. Format of the IBM standard data set label 2 continue

1—Label Identifier (3 bytes)

- *Contents*: Three characters that identify the label are as follows:
  - **HDR**
    Header label (at the beginning of a data set)

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

    EOF
    Trailer label (at the end of a data set).
- *Processing*: The system checks this field to verify that the record is an IBM standard data set label.

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

  If the DD statement specifies OPTCD=B for an input data set, the system accepts either EOV or EOF as the trailer label identifier, and the identifier is not used to determine whether a volume switch is necessary. If more volumes are available, the system performs the switching. If no volumes are available, the system passes control to the user's end-of-data routine or continues processing with a concatenated data set.

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

2—Label Number (1 byte)

- *Contents*: The relative position of this label within a set of labels of the same type; it is always a 2 for data set label 2.
- *Processing*: Verified and written in conjunction with Field 1 to identify this label as HDR2, 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 OPEN routine obtains the record format from this label, converts it to a bit code, and records it 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: This is a merging process in which existing specifications in the JFCB and DCB cannot be overridden.

  When creating labels, the system's OPEN and EOV routines do a reverse merge. 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 Figure 2 on page 4.

4—Block Length (5 bytes)

- *Contents*: A number up to 32 760 that indicates the block length, in bytes. Interpretation of the number depends on the associated record format in Field 3, as follows:

  – **Format F**
  Maximum block length (must be a multiple of the logical record length in Field 5). If field 12 contains a blank, then this field is also the minimum block length.

  Format V
  Maximum block length (including the 4-byte length field in the blocks)

  Format U
  Maximum block length.

  The system can determine the optimum block size when creating tape data sets. For more information, see *z/OS DFSMS Using Data Sets*.

  If the block length field contains zeros, then there is a large block interface block length. See field 17, Large Block Length.

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

5—Record Length (5 bytes)

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

  – **Format F**
  Logical record length

  Format V
  Maximum logical record length (including the 4-byte length field in the records)

  Format U
  Zeros

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

6—Tape Density (1 byte)

- *Contents*: A code indicating the recording density of the tape; the code is equivalent to the DEN parameter value on the DD statement. For the DEN parameter values, refer to "Tape characteristics" on page 11. For non-cartridge tapes, when density is not specified in the DCB or JCL DD statement, this field will be set to the highest density applicable to the drive (default).

  Note: Specifying DEN=0 for a 7-track 3420 results in 556 bits-per-inch recording, but corresponding messages and tape labels indicate 200 bits-per-inch recording density.

  For tape cartridges, this field contains a character 0.

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

7—Data Set Position (1 byte)

- *Contents*: A code indicating a volume switch is as follows:
  - **0**

    No volume switch has occurred

    1

    A volume switch previously occurred.
- *Processing*: Not used or verified. When creating labels, the open routine writes 0 in this field, and the EOV routine writes 1. The close routine determines which code to write by comparing the first volume serial number in the JFCB to the number in the UCB. It writes 0 if the numbers are equal, and 1 if they are not equal.

8—Job/Job Step Identification (17 bytes)

- *Contents*: Identification of the job and job step that created the data set. If your program extends the data set by using the EXTEND or OUTINX option of the OPEN routine, or by using the OUTPUT or OUTIN option of the OPEN routine with DISP=MOD, EOF2 contains the name of the job and job step that extended it. The first 8 bytes contain the name of the job, the ninth byte is a slash (/), and the final 8 bytes contain the name of the job step.
- *Processing*: Not used or verified. When the system creates labels, it sets this field as described above.

9—Tape Recording Technique (2 bytes)

- *Contents*: A code or blanks indicating the tape recording technique used to create the data set.

  For 7-track tapes the values are:

  - **Tƀ**

    Odd parity with translation

    Cƀ

    Odd parity with conversion

    Eƀ

    Even parity with no translation

    ET

    Even parity with translation

    ƀƀ

    Odd parity with no translation or conversion.

  The only technique available for 9-track tape is odd parity with no translation.

  For a magnetic tape subsystem with Improved Data Recording Capability, the values are:

  - **Pƀ**

    Record data in compacted format

    ƀƀ

    Record data in standard, uncompacted format

  For other tapes, this field is recorded as blanks.
- *Processing*: For 7-track tapes and cartridges written by a magnetic tape subsystem with Improved Data Recording Capability, the specification in the label is converted to a bit code and merged with the appropriate fields of the JFCB and DCB. The merging process is the same as that for the record format code in Field 3 of this label.

10—Control Character (1 byte)

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

    Contains ANSI control characters

M
Contains machine control characters

(blank)
Contains no control characters.

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

11—Reserved (1 byte)

- *Contents*: Reserved for possible future use (recorded as blanks).
- *Processing*: Not used or verified. When creating labels, the system writes blanks in this field.

12—Block Attribute (1 byte)

- *Contents*: A code indicating the block attribute used to create the data set:

  – **B**
  Blocked records

  S
  Spanned records, if the record format byte = 'V'

  S
  Standard records, if the record format byte = 'F'

  R
  Blocked and spanned records, if the record format byte = 'V'

  R
  Blocked and standard records, if the record format byte = 'F'

  (blank)
  Records that are not blocked and not spanned, or records that are not blocked and not standard.

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

13—Reserved (2 bytes)

- *Contents*: Reserved for possible future use (recorded as blanks).
- *Processing*: Not used or verified. When creating labels, the system writes blanks in this field.

14—Device Serial Number (6 bytes)

- *Contents*: For tape reels used by the 3420, byte 43 contains the model number, and bytes 44 through 47 contain the last 4 digits of the serial number of the creating tape unit.

  For tape cartridges written by IBM 3480 and 3490 magnetic tape subsystems, bytes 43 through 46 contain the last 4 digits of the serial number of the control unit, and byte 47 contains the device address.

  For tape cartridges written by IBM 3590 and 3592

  magnetic tape subsystems, bytes 42 through 47 contain the 6 digit device serial number.

  Note: The serial numbers in the header and trailer labels may not be the same if the data set was extended, or if the system swapped tape units while the data set was being created.

- *Processing*: A unique number to identify the recording unit is read off the tape during open processing, converted into binary, and inserted into the UCBCTD field in the UCB tape extension.

15—Checkpoint Data Set Identifier (1 byte)

- *Contents*: This byte contains the character C if the data set is a secure checkpoint data set; the byte is blank if the data set is not a secure checkpoint data set.

- *Processing*: This field is examined by open/close/EOV. If it finds the data set is a checkpoint data set, it performs the following security operations:
  - Verifies (by way of messages to the operator) that the data set is a secure checkpoint data set.
  - Determines whether the user is authorized. If the user is unauthorized, open/close/EOV does not allow access to the checkpoint data set directly, although it will allow the taking of checkpoints and performing restarts.

16—Reserved (22 bytes)

- *Contents*: Reserved for possible future use (recorded as blanks).
- *Processing*: Not used or verified. When creating labels, the system writes blanks in this field.

17—Large Block Length (10 bytes)

- *Contents*: A number that can be greater than 32 760 that indicates the block length, in bytes. Interpretation of the number depends on the associated record format in Field 3, as follows:
  - **Format F**
    Maximum block length (must be a multiple of the logical record length in Field 5). If field 12 contains a blank, then this field is also the minimum block length.

    Format V
    Maximum block length (including the 4-byte length field in the blocks)

    Format U
    Maximum block length

  The system can determine the optimum block size when creating tape data sets. For more information see *z/OS DFSMS Using Data Sets*.
- *Processing*: Used when field 4, the Block Length field, contains zeros. The number in the label is converted to binary and merged with appropriate fields in the JFCB, SWA, DCB, and DCBE. The merging process is the same as that for the record format code in field 3 of this label. If the maximum block length is 32 760 or less, the OPEN and EOV routines write it in field 4 and not this field. On input, the OPEN and EOV routines accept small or large values in the Large Block Length field.

Format of IBM standard user label (UHL1 - UHL8 or UTL1 - UTL8)

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

Figure 9 on page 33 shows the format of user labels. The format is used for user header labels (UHL1-UHL8) and user trailer labels (UTL1-UTL8). The labels are 80 characters in length. They are recorded in the density specified using JCL. Seven-track tape records are in BCDIC. The contents and processing of each field of the label are described below.

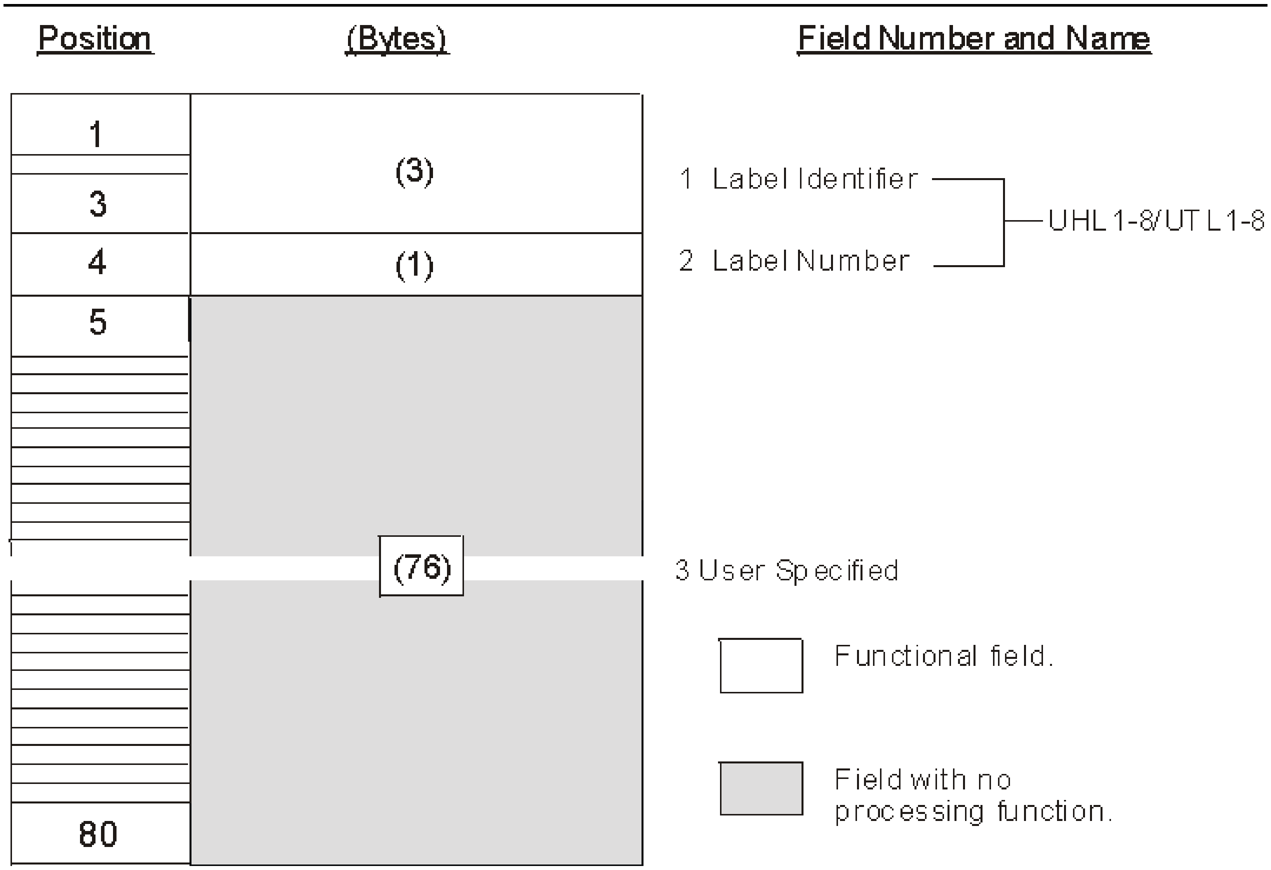


Figure 9. Format of user label

1—Label Identifier (3 bytes)

- *Contents*: Three characters that identify the label are as follows:
  - **UHL**
    User header label (at the beginning of a data set)

    UTL
    User trailer label (at the end-of-volume or end-of-data-set).
- *Processing*: This field is read to verify that the record is a user label. The system accepts either UHL or UTL.

2—Label Number (1 byte)

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

3—User Specified (76 bytes)

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

IBM standard label processing

Label processing is handled by the open/close/end-of-volume routines. Label processing consists of the following 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
- For volumes in an automated tape library, comparing the external volume serial number to the magnetically recorded internal label

These processing functions are summarized in Table 4 on page 34. The table shows the specific labels that are processed for each function and which routines perform the functions.

Table 4. IBM standard label processing by Open/Close/EOV routines

Processing	Volume Label VOL1	Header Labels"1" on page 35			Trailer Labels"1" on page 35		
		HDR1	**HDR2**	**UHL1-8**	**EOF1 or EOV1**	**EOF2 or EOV2**	**UTL1-8**
First or only volume"2" on page 35							
Checks labels on input tape	Open	Open	Open	Open	EOV	Bypassed	EOV
Checks existing labels on output tape before overwriting	Open	Open	Open	Not read	Not read	Not read	Open"3" on page 35
Writes new labels on output tape	Open or user"4" on page 35	Open	Open	Open	Close or EOV	Close or EOV	Close or EOV
Compares the volume serial number on the external volume label to the serial number in the tape volume label"6" on page 35	Open	Open	N/A"7" on page 35	N/A"7" on page 35	N/A"7" on page 35	N/A"7" on page 35	N/A"7" on page 35
Second or subsequent volumes"5" on page 35							
Checks labels on input tape.	EOV	EOV	Bypassed	EOV	EOV	Bypassed	EOV
Checks existing labels on output tape before overwriting.	EOV	EOV	Not read	Not read	Not read	Not read	Not read

Table 4. IBM standard label processing by Open/Close/EOV routines (continued)

Processing	**Volume Label VOL1**	**Header Labels“1” on page 35**			**Trailer Labels“1” on page 35**		
		HDR1	**HDR2**	**UHL1-8**	**EOF1 or EOV1**	**EOF2 or EOV2**	**UTL1-8**
Writes new labels on output tape.	EOV or user“4” on page 35	EOV	EOV	EOV	Close or EOV	Close or EOV	Close or EOV
Compares the volume serial number on the external volume label to the serial number in the tape volume label“6” on page 35	EOV	N/A“7” on page 35	N/A“7” on page 35	N/A“7” on page 35	N/A“7” on page 35	N/A“7” on page 35	N/A“7” on page 35

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 when the user program has turned on the DCB unlike attributes bit (DCBOFPPC). See *z/OS DFSMS Using Data Sets*.
3. If DISP=MOD is specified on the DD statement, or the OPEN macro specifies EXTEND or OUTINX, then 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).
4. User can create the label with the EDGINERS or IEHINITT utility programs, or a user program. Subsequently, the label may be rewritten by the open and EOV routines.
5. Includes the first volume of concatenated data sets when the user has not turned on the DCB unlike attributes bit.
6. Applies only to tape library.
7. Not Available.

When a data set is opened for input, the volume label and HDR1 are processed. HDR2 is processed if it exists. For an input end-of-data condition, the trailer labels are processed, unless deferred user input trailer label processing is specified in the DCB exit list. For an input end-of-volume condition, the trailer labels on the current volume are processed, and then the volume label and header labels on the next volume are processed. When the FEOV macro is issued for an input tape, the trailer labels on the current volume are not processed, but the volume labels and header labels on the next volume are processed. No label processing is performed when an input data set is closed, unless deferred user input trailer label processing is specified in the DCB. If deferred user input trailer label processing is specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

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

Creating a volume label

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

You can use the EDGINERS or IEHINITT utility programs to create IBM standard volume labels. EDGINERS and IEHINITT initialize the tape by writing in the following order:

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

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 existed on the tape. Therefore, IEHINITT does not check for password or RACF security protection. It does not create, modify, or delete RACF profiles of RACF-defined volumes. Detailed procedures for using the program are described in *z/OS DFSMSdfp Utilities*.

The EDGINERS utility program verifies that the volume is mounted before writing a volume label on a labeled, unlabeled, or blank tape. EDGINERS does not check password or RACF security protection, but it verifies whether the volume is defined to DFSMSrmm. DFSMSrmm must know that the volume needs to be labelled. If the labelled volume is undefined, then DFSMSrmm defines it to DFSMSrmm and can create RACF volume security protection. Detailed procedures for using the program are described in *z/OS DFSMSrmm Implementation and Customization Guide*.

Methods other than the EDGINERS or IEHINITT utility programs are available to write volume labels. You can use a card-to-tape program, or the system programmer can replace the IBM-supplied volume label editor routine with one that writes volume labels. On some old models of tape drives with reels, some data or a tape mark should already exist on the tape. See "IBM standard volume label editor routines" on page 37 for more information. Otherwise, the system might read through the entire reel of blank tape looking for a record or a tape mark.

The VOL1 label is rewritten by the open or EOV routine if *all* the following conditions are met. It is not necessary for a conflict to occur in order for the VOL1 label to be rewritten:

- If a density, track, or label type conflict occurred, then permission must be granted by the installation's label anomaly exit, the installation's label editor routine, DFSMSrmm, or the system operator.
- OUTPUT, OUTIN, or OUTINX is specified in the OPEN macro.
- The tape is positioned to the first data set on the volume.
- Either of the following two conditions are true:
  - The data set is not password protected, or
  - The volume is RACF protected, the system-wide RACF tape protection option has been specified, and the user is ALTER authorized

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

If you request a standard labeled (SL, SUL) output volume and the tape that you are allocated is recorded in the wrong density and cannot be read, the VOL1 label is rewritten in the density that you specify. This facility allows you to make nonspecific requests, you do not need to specify a volume serial number in your DD statement for output tapes. The operator can mount any scratch volume to answer your request. However, if the system-wide RACF tape protection option has been specified, the volume is rejected, because it cannot be verified that it is not a RACF-protected volume.

If you make a nonspecific output volume request for the first data set for a standard labeled (SL, SUL) tape and the mounted volume is an NL, NSL, AL or AUL labeled tape, the open or the EOV routine creates a volume label (VOL1). A message is sent to the console operator requesting serial number and owner information. The label anomaly exit volume label editor exit and DFSMSrmm can optionally prevent the creation or the destruction of standard labels.

IBM standard volume label editor routines

If you specify that an input or output tape has a standard label, the operating system checks for the standard volume label at the beginning of the tape.

Because of conflicting label types or tape characteristics, various error conditions can occur during the verification of the first record. Under some error conditions, the tape is accepted for use. Under other 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 label anomaly exit routine and to a volume label editor routine; your installation can use IBM-supplied routines, or it can supply its own routines.

The IBM-supplied volume label editor routines determine the discrepancies between the requested tape and the mounted tape, and if necessary, pass control to the appropriate system routine to create or destroy labels. If DFSMSrmm is controlling the volume, the DFSMSrmm label editor routine controls the creation or destruction of standard labels using RACF profiles in the FACILITIES class. Installation-supplied routines can perform other functions.

IBM recommends you use tape management exit routines instead of the volume label editor routines.

For detailed information about these and other available exit routines, see *z/OS DFSMS Installation Exits*.

Protecting data

Data on IBM standard volumes can be protected by either RACF or data set password protection. In an IBM system-managed tape library, data set password protection is not supported.

IBM recommends RACF instead of data set password protection. For more information see *z/OS DFSMSdfp Advanced Services*.

Note: All checking for authorization will be bypassed if security processing is suppressed. This can occur, for example, when the program properties table entry for the job step program is marked to suppress security checking. Only the system programmer can update the program properties table. For information about the program properties table, see *z/OS MVS Initialization and Tuning Reference*.

RACF allows you to control access to either the tape volumes or the individual data sets on the tape. RACF protection at the volume level overrides RACF protection at the data set level. See *z/OS Security Server RACF Security Administrator's Guide* for information on how to activate these levels of RACF protection, and how they interact with each other and with your own tape management system, if applicable.

DFSMSrmm supports RACF protection, but not password protection. For more information about DFSMSrmm and RACF, see *z/OS DFSMSrmm Implementation and Customization Guide*.

The following principles apply to RACF protection at the volume level:

- ALTER access authority is required to create or destroy the tape volume label.
- READ access authority is required to open the volume for input (open options INPUT or RDBACK). Note that if your program uses the INOUT option of OPEN and the DD statement has LABEL=(,,,IN), the system treats it as the INPUT option.
- UPDATE access authority is required to open the volume for output (open options OUTPUT, EXTEND, INOUT, OUTIN, or OUTINX).

If the tape volume is defined to RACF, the user has UPDATE access authority, and PROTECT=YES has not been specified in the JCL, the user can open the volume to read or write.

If the tape volume is defined to RACF and the user has UPDATE authority, and PROTECT=YES has been specified in the JCL, and the tape is not a RACF scratch volume, the request fails.

If the tape volume is defined to RACF and the user has READ but not UPDATE access authority, or if the user has UPDATE access but PROTECT=YES has been specified in the JCL and the volume is a RACF scratch tape volume, the system does not grant the user access to read until it has ensured that the user will not be able to write on the tape. The user cannot access the volume until one of the following conditions is met:

1. **Hardware Protection.** If the write-enable ring has been removed from the tape reel or the write-protect tab has been set to disable writing on the tape cartridge, the tape volume cannot be written on by any user, so the system safely permits the user to access the tape to read. This hardware protection cannot be circumvented by software.
2. **Logical write-protection.** If the write-protect tab on an IBM magnetic tape cartridge is set to enable writing, the system issues a hardware command to prevent writing on that cartridge. If the command succeeds, the system safely allows the user to access the tape to read. An unauthorized program cannot bypass this combination of hardware and software protection.
3. **IEC.TAPERING.** Your installation may choose to depend on a tape management system to prevent overwriting unexpired data on tapes. Typically, a tape management system only allows volumes with no unexpired data to be opened for output. DFSMSrmm provides facilities to prevent accidental overwriting of non-scratch volumes. The IEC.TAPERING support facilitates the operation of tape management systems because it allows all volumes to remain write-enabled (by the ring in the volume or the switch on the cartridge), eliminating the need for further operator intervention.

   If the write ring or cartridge tab is set to enable writing, the system checks if the user is authorized for read to the IEC.TAPERING profile in the RACF FACILITY class. If the user does have this authority, the system grants the user access.

   ⚠ **Attention:** If you use the IEC.TAPERING support to allow users to read from tapes that are enabled for writing (when the users are only authorized to read), the system software cannot prevent knowledgeable users from also writing on any files on the tapes.
4. **Operator Intervention.** If none of the preceding conditions are met, the system requires the operator to intervene and prevent writing on the volume. The system demounts the tape and issues a message asking the operator to remove the write-enable ring from the tape reel or change the switch on the tape cartridge. After the operator remounts the tape, the system continues to protect the volume from unauthorized writing by repeating the preceding checks, beginning with the check for hardware protection, until one of the conditions is met.

If the tape volume is not defined to RACF, access is granted and processing continues. For an overview of RACF protection for tape volumes, see *z/OS Security Server RACF Security Administrator's Guide*. For information on how DFSMSrmm can help you manage RACF security for your tape volumes, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Data set password protection is described in *z/OS DFSMSdfp Advanced Services*.

Opening an input data set

If you specify or default to IBM standard labels, the first record on the input tape must be an IBM standard volume label (VOL1). At the time the data set is opened, the system checks the first record on the tape to determine whether the record is 80 bytes long and contains the identifier VOL1 in the first 4 bytes. The various error conditions that can occur during verification of the first record are explained in "IBM standard volume label editor routines" on page 37.

Before OPEN verifies the volume, OPEN will abnormally terminate if the system-wide RACF tape protection option is specified and the DD statement specifies PROTECT=YES without previously opening the DD statement for output processing.

Volume serial number

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

Serial numbers are also required when the processing option is OUTPUT or OUTIN with DISP=MOD.

If the volume serial number is correct, the volume is considered to be mounted and verified. If the serial number is not correct, the system rejects the tape and issues another mount message.

Positioning the volume to the data set

After the volume is mounted and verified, the system 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 volume, and the header label group immediately follows the volume label. DFSMSrmm provides support for high speed positioning to other than the first data set on the volume. The OPEN provides the block identifier, and DFSMSrmm saves it in its control data set and supplies it back to OPEN on any subsequent request for the data set.

To retrieve a data set on a single volume of tape that contains multiple data sets, you specify a data set sequence number in the LABEL parameter of the DD statement, unless the data set is cataloged. For a cataloged data set, you need not specify a data set sequence number because the system obtains the number from the catalog along with the volume serial number.

- The sequence number can be from 1 to 65535, 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, your task will be abnormally terminated or, if the volume ends with EOV labels, the open routine will switch to the next volume.
- If the data set is not cataloged and you do not specify a sequence number, or you specify 0, the system assumes that the data set is the first in sequence on the volume.

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

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened, the open routine sets the data set sequence number in the UCB to 1. The exceptions are:

   - If the tape is still positioned from previous processing, such as for a LEAVE request, the open routine does not reset the number in the UCB.
   - If the data set sequence number in the JFCB and the data set sequence number in the first HDR1 label on the tape are both greater than one, the open routine sets the data set sequence number in the UCB to the value of the number in the first HDR1 label. (The data set sequence number in the first HDR1 label may be greater than one when the volume is part of a multiple-data-set/multiple-volume aggregate.)
   - When the processing method is INPUT, INOUT, OUTPUT or OUTIN to the start of a data set on a multiple file tape, the open routine starts with the first volume, unless a volume sequence number is specified. If the open routine finds that the volume ends with EOV labels before the desired file sequence number, the open routine switches to the next volume and permanently update the volume sequence number so that the next open to this data set will start with the correct volume.
   - When the processing method is RDBACK, and a volume sequence number was not specified, the open routine speeds up finding the end of the data set by starting with the last volume specified.

   Note: The use of the DCB parameter in the JCL causes a specific volume sequence number of 1. If the data set is not yet present on the last volume specified, the open routine can recover, if the file sequence number is 1, by backing up volumes. It detects that the data set is not present if the *dsname* is invalid, the tape starts at a file sequence number greater than 1, or the VOL label is followed by a tape mark.

3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the open routine adjusts the data set sequence number in the UCB as the tape is positioned 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.
5. If the data set is not open or has been closed, the data set sequence number field of the UCB will be set to X'0000' if:

- The data set was never opened.
- CLOSE (,REWIND) was specified.
- CLOSE (,REREAD) and LABEL=1 was specified.
- CLOSE (,DISP) was specified or defaulted, and DISP=(,PASS) was not specified on the JCL.

Otherwise, the data set sequence number in the UCB will have a value one greater than the value specified on the LABEL parameter of the JCL.

6. If the job terminates abnormally while a tape data set is open, the data set will be closed and the tape will be positioned as when CLOSE (,LEAVE) is specified. That is, the data set sequence number in the UCB will have a value one greater than that specified on the LABEL= parameter of the JCL.

There are several instances in which a volume is repositioned to the next (or previous) tape mark during the open routine. This is usually done by reading data but suppressing data transmission to storage until a tape mark is found, but can be done by I/O spacing commands (for example, BACKSPACE FILE). To reduce the chance of an unexpected record condition (613-0C), the first method is preferred over the spacing commands. In the event of a 613-08 or 613-0C abend, a data management abend installation exit (IFG0199I) is given control to try further recovery. For more information about the data management abend installation exit, see *z/OS DFSMS Installation Exits*.

Only one data set on a tape volume may be open at any given time. An attempt to begin processing 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, the system checks the label identification. Processing is abnormally terminated if the HDR1 identifier is not found.

Processing the HDR1 label

This topic covers the following subjects:

- "Verifying data set name" on page 40
- "Open/EOV user exit for security verification" on page 40
- "Password protection" on page 40
- "Expiration date" on page 41
- "Block count" on page 41

Verifying data set name

To ensure that the correct data set is being opened, the system compares the data set name in the HDR1 label to the data set name specified by the user in the DD statement. This comparison is made on only the 17 least significant (rightmost) nonblank 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). DFSMSrmm checks the full 44-character data set name if the previous data set name information is known to DFSMSrmm.

Processing is abnormally terminated if the comparison shows an incorrect data set name.

Open/EOV user exit for security verification

For authorized program facility (APF) programs for which the program property bypass password (and RACF) checking is active, a user exit is provided for verifying that a tape selected by Open or EOV should be used, and whether certain security checks may be bypassed. For more information on the Open/EOV volume security and verification exit, see *z/OS DFSMS Using Data Sets*.

Password protection

Password protection is not as strong as RACF protection. In each of the following situations, the system ignores any password protection for a tape and uses RACF protection:

- The volume is RACF protected

- The volume is in an automated tape library
- The volume is in a manual tape library

If RACF or an equivalent is not running, then the volume has no security protection

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, the system notifies the operator and asks for confirmation of the 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. Automatic handling of expiration date protected tapes is provided by DFSMSrmm functions.

Block count

The block count shown in the HDR1 label is always 0 (EBCDIC or BCDIC). This 0 is recorded in the data control block (in binary) and increased 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 decreased during processing for comparison to the 0 block count in the HDR1 label.

The block count is verified at end-of-data or end-of-volume. For more information, see “Verifying block count” on page 10 and “Verifying block count in label when reading” on page 42.

Processing the HDR2 label

This topic covers the following subjects:

- “Data set characteristics” on page 41
- “Positioning after header label 2” on page 41

Data set characteristics

The HDR2 label immediately follows the HDR1 label. The system 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
- Type of control characters

The information is obtained from the label and recorded in the job file control block (JFCB) and the data control block, provided that 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. For an explanation and illustration of this merging process, see “Completing the data control block” on page 3.

Positioning after header label 2

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

Note: If the HDR2 label is missing, processing will continue with the assumption that it is not needed. If the JFCB/DCB merge function needs it because of missing data set characteristics, the OPEN will abnormally terminate.

User header labels

Up to eight user header labels (UHL1 to 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 SUL or the user header label entry in the DCB exit list, the system positions the tape past the tape mark immediately after processing the HDR2 label, or the HDR1 label if the HDR2 label is not present. User label processing and the DCB exit list are described in *z/OS DFSMS Using Data Sets*.

Read backward

For the read backward (RDBACK) processing method, the system 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 is read before EOF2, and so forth. The data records, however, are read in reverse sequence.

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

End-of-Data or end-of-volume on input

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

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

After encountering a tape mark, the system checks the first 4 bytes of the first trailer label for the EOV1 or EOF1 identifier. If neither identifier is found, processing is abnormally terminated. When the FEOV macro is executed, the identifier of the first trailer label is not checked. The trailer labels on the current volume are not processed but the volume labels and header labels on the next volume are processed.

For an end-of-data condition, the first trailer label is processed, unless deferred user input trailer label processing is specified in the DCB exit list. For an end-of-volume condition, the first trailer label on the current volume is processed, and then the volume label and header labels on the next volume are processed.

Except when it is used as a header label for a RDBACK operation, the system ignores the second trailer label (EOV2 or EOF2) of an input data set.

If UTLs are present on input, the system 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.

Verifying block count in label when reading

To verify that all records on the input data set on the current volume have been read, the system 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, the system compares the 0 block count shown in the HDR1 label against the block count in the data control block.

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

Register
    Contents

0
    The block count shown in the label

1
    The address of the data control block.

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

Return Code
Meaning

0
Abnormally terminate with completion code of X'237'

4
Continue processing

For more information, see *z/OS DFSMS Using Data Sets* and “Verifying block count” on page 10.

If the appropriate user exit entry is not provided in the DCB exit list, or a return code of 0 is received, then a block count discrepancy causes the system to call the optional DCB ABEND exit and issue ABEND 237. When the FEOV macro is executed, the block count is not verified. For more information, see *z/OS DFSMS Using Data Sets*.

If the data set was created with an EXCP DCB with no device-dependent section, then the block count is written as zero and is not verified.

The EOV2/EOF2 label

The system ignores the EOV2 or EOF2 trailer label 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, the system can make them available to your program. To make them available, SUL must be coded on the DD statement and the address of an input user trailer label routine must be specified in the DCB exit list.

Determining volume switch

For a multivolume input data set, you must specify the serial numbers of all the volumes to be processed. Specify the serial numbers either directly in the DD statement or indirectly through the catalog procedure. 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. The system processes the group of volumes in whatever order you specify and processes only the volumes you specify. If you specify volumes out of order or omit volumes, you will get warning messages IEC709I or IEC710I.
- For cataloged data sets, the system processes the group of volumes in sequential order. You can begin processing at any volume of the group by specifying a volume sequence number in the VOLUME parameter of the DD statement.

For input, the label identifier of the trailer labels determines whether the system continues processing the data set. When the system finds an EOV label, it performs volume switching. When the system finds an EOF label, it passes control to the user's end-of-data routine, with these exceptions:

- If the DD statement specifies OPTCD=B, and if an additional volume is available, then the system performs volume switching. If OPTCD=B, no volumes are available for switching, and no data set is concatenated after the current one, then the system passes control to the user's end-of-data routine.

  Guidelines: Know these guidelines when you specify the OPTCD=B parameter:

  - If you specify the OPTCD=B parameter on the DD statement for a multi-volume tape data set, the system will generate the equivalent of individual concatenated DD statements for each volume serial number. This means that the system allocates one tape drive for each volume.
  - Be aware that you might receive an ABEND002 when the following conditions occur:

  - If you are using unlike attributes concatenation to process a multi-volume variable blocked spanned (VBS) tape data set and each volume serial number is specified on individual concatenated DD statements because you specified them that way, or because OPTCD=B was specified, and
  - If the data set spans volumes in such a way that one segment (for example, the first segment) is at the end of the current volume and the next segment (for example, the middle segment) is at the beginning of the next volume.
  - If you are using unlike attributes concatenation path for tape with OPTCD=B and VBS, and
  - If you have a VBS data set whose records span from one volume to the next, and
  - If you attempt to treat these volumes as separate data sets, the integrity of the data cannot be guaranteed. QSAM will abend. QSAM is designed to ensure that all of the segments can be put together, but will be unable to do so. This restriction will also be based on the data and whether or not the segments are split between volumes.

- If another data set is concatenated after this one, the system processes it. If the user has not set on the unlike-attributes bit (the DCBOFPPC bit), then the system treats the concatenated data set as if it began another volume of the current data set. If the user defined an EOV exit, the system calls it (not a DCB open exit). If the unlike-attributes bit is on, then the system closes the current data set and opens the concatenated data set. If the user defined a DCB open exit, the system calls it (not an EOV exit). For more information on using concatenated data sets, see *z/OS DFSMS Using Data Sets*.

To determine whether additional volumes are required, the system maintains a volume sequence number in the DEB in storage.

- For read forward operations, the volume sequence number in the DEB is increased 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 decreased as each volume is processed until the count reaches 0.

If another volume is not required (end-of-data-set condition), control is given to the user's end-of-data routine that is specified in the data control block (DCB) or DCBE. 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), the system 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 checks to see if a next-plus-one volume is specified, and if the volume just completed can be rewound and unloaded. If so, the EOV routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid; the next-plus-one volume label is not verified at this time. The next-plus-one volume mount is referred to as a lookahead mount. This lookahead or parallel mounting will result in an IEC501E mount message.

Lookahead mount processing is disabled when the volume sequence number that EOV is currently processing equals either the number of volsers in the JFCB or the JCL volume count. To ensure that lookahead mount processing stays enabled up to the maximum allowed volume sequence, specify a JCL volume count equal to this maximum allowed volume sequence. The smallest maximum allowed volume sequence is 5 to which you can add multiples of 15. For example, you can specify the JCL volume count as 5, 20, 35, and so on.

Note: The IEC501E message has a type 2 descriptor code that will cause the message to be retained on the screen.

If you do not want message retention, turn off the message retention attribute. You can accomplish this by either:

- Specifying RETAIN(NO) for the message id in an MPFLSTxx record
- Using a WTO message exit (an MPF specific exit or IEAVMXIT) to alter the message descriptor code or to request "do not retain this message"

For tape mounts, the IEC501E message will remain on the screen until the mount is satisfied or until EOV has occurred on the previous volume. EOV will delete the IEC501E message and will issue the IEC501A message.

Checking the next volume

When volume switching is performed, the EOV routine checks the volume and header labels on the new volume to ensure that the correct volume is mounted.

RACF access authorization is required for system-wide RACF tape protection. READ authorization is required if the volume is opened for INPUT or RDBACK. UPDATE authorization is required if the volume is opened for INOUT, OUTIN, or OUTINX. If a multivolume data set is RACF protected, then each volume data set must have the same profile for INOUT, OUTIN, or OUTINX (not INPUT or RDBACK) access.

No authorization is required if the new volume is not RACF protected and the user is authorized.

A specific volume request fails if the new volume is RACF protected and the user is unauthorized.

For a nonspecific volume request, the operator mounts a volume.

If the user is READ authorized but not UPDATE authorized, and the data set is open for INPUT or RDBACK, the system verifies that the volume is set to prevent writing. If the volume is not protected, the operator is notified to prevent the tape volume from writing by removing the write-enable ring from the reel or changing the write-protect tab on the cartridge. "Protecting data" on page 37 describes when operator intervention is necessary, and when other mechanisms of protection eliminate the need for it.

The method of locating and checking the HDR1 label varies according to the situation. The processing depends on whether the data set is a continuation of a multivolume data set or is a concatenated data set. In either case, the user exit for security verification may be taken, as described in "Open/EOV user exit for security verification" on page 40.

- *Multivolume data set*: The data set sequence number is irrelevant for the second and subsequent volumes of a multivolume data set. The EOV routine assumes that the data set continues at the beginning of the new volume. Therefore, the first header label group on the tape is checked. The HDR1 label is checked in the same manner as when the data set was opened on the first volume. Processing is abnormally terminated if:
  - The data set name is not the same
  - The volume is password protected, no previous volume in the data set was password protected, and the system operator or TSO user is not able to enter a correct password

  Note: For volumes residing in an IBM automated tape library, or manual tape library, password protection processing of the data tape set is bypassed. These tapes can be protected with RACF.
- *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.

  Note: For volumes residing in an IBM automated tape library, or manual tape library, password protection processing of the data tape set is bypassed. These tapes can be protected with RACF.

The HDR2 label on each volume after the first volume within a multivolume data set 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, unless deferred user input trailer label processing was specified in DCB exit list. If deferred user input trailer label processing was specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

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

Opening an output data set

If IBM standard labels are specified, the first existing record on the output tape must be an IBM standard volume label (VOL1). At the time the data set is opened, the system checks the first existing record on the tape to determine whether the record is 80 bytes long and contains the identifier VOL1 in the first 4 bytes. The various error conditions that can occur during verification of the first record are explained in "IBM standard volume label editor routines" on page 37.

If the system-wide RACF tape protection option has been specified and the DD statement has specified PROTECT=YES, and the DD statement has not been previously opened for output processing, open ensures the PROTECT=YES specification is valid. Both the volume sequence number and the file sequence number must be set to 1, and a private volume must be requested. The protection indicator in the JFCB is reset so that subsequent OPENs of that DD statement for output processing do not attempt validity checking (of the PROTECT=YES specification) and defining the volume to RACF.

Attention: Doing multiple opens and closes without writing any user data in the end-of-tape reflective marker area can result in the writing of header and trailer labels past the reflective marker. Access methods detect the reflective markers; however, since the creation of empty data sets does not involve access methods, the end-of-tape marker is not detected. This can cause the reel or cartridge to run out of tape.

Volume serial number

You are not required to specify volume serial numbers for output tapes, unless you are extending an existing tape data set. Serial numbers are required when the processing method is EXTEND or OUTINX or it is OUTPUT with DISP=MOD. If no number is specified, the mount message directs the operator to mount a scratch tape. The system obtains the volume serial number from the VOL1 label and records it in the JFCB and the UCB.

An optional user exit can identify a specific tape volume in place of a nonspecific volume. The exit is invoked when an open or EOV routine issues a mount request for a volume with no volume serial number. For more information about the Open/EOV nonspecific tape volume mount exit, see *z/OS DFSMS Using Data Sets*.

If you choose to specify the volume serial number, the system compares it with the volume serial number shown in the VOL1 label. When the serial numbers match, the system resets a mount switch in the unit control block to indicate that the mounted volume is verified. The mount switch is initially set when the mount message is issued to the operator. When the volume serial numbers do not match, the system may give the operator the option of having the label rewritten with the serial number of the volume requested. This occurs if the tape is not password protected, is not date protected, and is not a checkpoint/restart volume. Otherwise, the system rejects the tape and issues another mount message. Volumes are requested for mounting in the specified order. DFSMSrmm prevents an incorrect tape from being used and the label being rewritten.

If the system-wide RACF tape protection option has been specified, RACF authorization at the UPDATE level is checked. If the tape volume is not defined to RACF, processing continues. If the tape volume is defined RACF, and the user is UPDATE authorized and PROTECT=YES has not been specified, processing also continues.

If the tape volume is defined to RACF, the user is UPDATE authorized, and PROTECT=YES has been specified, processing will be abnormally terminated. If the tape volume is defined to RACF and the user is not UPDATE authorized, processing will also be abnormally terminated. When processing is abnormally terminated, the volume will be rejected if a nonspecific request is made, or the program will be abnormally terminated if a specific request is made.

Note: Beginning with RACF 1.7, if you specify PROTECT=YES in a DD statement for a volume or data set that you previously protected in this manner, you do not abnormally terminate due to this latter PROTECT=YES specification.

Positioning the volume to the data set

After the volume is mounted and verified, the system positions the tape to receive the new data set. Usually the new data set is the first and only data set on the tape, so the tape remains positioned immediately following the VOL1 label. DFSMSrmm provides support for high-speed positioning for data sets other than the first data set. The OPEN routine provides the block identifier, and DFSMSrmm saves it in its control data set and supplies it back to OPEN on any subsequent request to the data set.

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 65535, with 1 representing the first data set on the volume. If the volume ends with EOV labels before the specified sequence number, the open routine switches to the next volume. If you specify a sequence number that is greater than the number of data sets existing on the volume, plus one, your task is abnormally terminated.
- If you do not specify a sequence number, or specify zero, the system assumes that the data set is to be written as the first on the volume.

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

Only one data set on a tape volume can be open at any given time. Processing is abnormally terminated if you attempt to open another data set on the same volume.

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

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

- When a tape mark immediately follows the VOL1 label. This can occur when the tape is initialized by means other than the EDGINERS or IEHINITT utility programs (EDGINERS and IEHINITT write a dummy HDR1 label following the VOL1 label). The tape mark is overwritten by the new HDR1 label.
- When the new data set is to be written after the last existing data set on the volume (for multiple data set organizations). In this case, the system encounters the second tape mark following the existing EOF trailer label group. The tape mark is overwritten by the new HDR1 label.

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

High-Speed search on an IBM tape cartridge

Positioning to the End of a data set

When you want to extend a data set written on a tape cartridge by an IBM magnetic tape subsystem using OPEN for OUTPUT or OUTIN with DISP=MOD, or OPEN for OUTINX or EXTEND, you can use the *high-speed search* function. This function quickly positions the volume to the end of the requested data set. After the tape is positioned, the open module processes the trailer labels of the data set to be extended.

To invoke the high-speed search function when extending a data set, use the high-speed search interface as follows:

1. Obtain the JFCB for the requested data set by using the read JFCB-RDJFCB macro.
2. Set the fast positioning indicator in the JFCB (JFCPOSID in JFCBFLG3).
3. Set the block identifier in the JFCB (JFCRBIDO=blk-id).
4. Execute OPEN TYPE=J with the modified JFCB.

Note: DFSMSrmm automatically invokes the high speed search function when extending a data set.

The block identifier for high speed positioning is for the tape mark immediately following the last block of user data. You can obtain the block identifier when the data set is created by issuing the NOTE macro with the ABS parameter before issuing CLOSE. This can be done only with BSAM or EXCP, not with QSAM.

Positioning to the beginning of a data set

A similar method is used when you want to use the high speed search function to quickly position the volume to the beginning of the data set. The difference is that the block identifier is for the first standard header label of the requested data set when positioning to the beginning of the data set.

For more information about the high speed header label search function, see *3480 Magnetic Tape Subsystem: User's Reference*.

If the *fast positioning* indicator is set to on in a JFCB used by an open routine that is not TYPE=J, the fast positioning indicator will be reset.

When fast positioning is indicated, but a block identifier is not specified in JFCRBIDO, OPEN TYPE=J positions the tape normally and inserts a block identifier in JFCRBIDO. The block identifier is either for the first header label when opening to the beginning of a data set or for the tape mark immediately following the last block of user data when opening to extend a data set. You can use this technique with BSAM, QSAM, and EXCP.

OPEN does not update your copy of the JFCB. To retrieve the new value in the system's copy of the JFCB, issue RDJFCB after OPEN.

Issue a second RDJFCB request after the OPEN to update the JFCB JFCPOSID flag when the fast positioning is indicated.

Once you have turned on the fast positioning indicator in a JFCB for an OPEN with TYPE=J in a job step, make sure the fast positioning indicator and the block identifier in JFCRBIDO reflect your intentions before any subsequent OPEN with TYPE=J for the same data set.

After OPEN with TYPE=J uses JFCRBIDO for a high speed search, it clears JFCRBIDO in the system copy of the JFCB to prevent misinterpretation in a subsequent OPEN.

Open/EOV user exit for security verification

For APF-authorized programs for which the program property "bypass password (and RACF) checking" is active, a user exit is provided for verifying that a tape selected by open or EOV should be used, and whether certain security checks may be bypassed. For more information on the "Open/EOV volume security and verification" exit, see *z/OS DFSMS Installation Exits*.

Expiration date on existing label

The existing HDR1 label is inspected for the expiration date. If it is an SMS-managed scratch volume, then the system ignores the expiration date in the label. Otherwise, if the expiration date has not been reached, the system asks the operator to confirm use of the tape or to mount another tape. DFSMSrmm provides facilities that automate the handling of expiration date protection. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

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

Password protection and data set name on existing label

After checking the expiration date, the system 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, or if protection is indicated but the volume resides in an IBM automated tape library, or a manual tape library, the tape is accepted for output. There is no password protection for IBM system-managed tape libraries.

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

Two additional restraints are placed on creation of password-protected data sets:

- If you want to create a password-protected data set following an existing password-protected data set, you must supply the password of the existing data set. The security indicator must be the same in both the existing and the new data set. This consistency test is made even if the volume is RACF defined, or resides in an IBM automated tape library or manual tape library.
- When creating a multivolume, password-protected data set, the second and successive volumes will also be verified. Verification consists of ensuring that the data set name in the JFCB is the same as the data set name in the password record and that the protection-mode indicator allows writing to the data set.

If the data set name is correct, and if the tape volume has not been found to be RACF defined, the system requests the operator or TSO terminal user to key in the required password. The password is verified in the system's password data set. This password data set contains the data set name, the password, and a protection-mode indicator. The protection-mode indicator is set to permit either read/write or read-only operations. The read/write mode is necessary for output data sets. Processing is terminated if:

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

z/OS DFSMSdfp Advanced Services describes data set protection in detail and contains the information the system programmer needs 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 tape marks following the HDR1 label are irrelevant and may be overlaid by the new output.

Writing data set header labels

When the tape is accepted by the system for output, the system 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 the label fields is explained in the description of the label formats. The process of updating the JFCB is explained in Figure 2 on page 4.

The security indicator is set in the HDR1 label even if the volume is RACF defined, unless the volume resides in an IBM automated tape library in which case there is no password protection.

If no user header labels are to be written, the system writes a tape mark 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 LABEL parameter on the DD statement and the address of an output user header label exit is specified in the DCB exit list, the system can write as many as eight user header labels (UHL1 to UHL8). The DCB exit list and user label exits are described in *z/OS DFSMS Using Data Sets*.

Permanent I/O error

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

End-of-Volume on output

The system's EOV routine automatically switches volumes when one of the following EOV conditions occur:

- The reflective strip on a reel is encountered
- The logical end of a tape cartridge is reached
- The FEOV macro is executed

This volume switching includes:

1. Checking whether the expected number of blocks were read or written (see "Verifying block count" on page 10)
2. Writing trailer labels and a tape mark on the current volume
3. Ensuring that the next volume is mounted
4. Checking existing volume and header labels on the new volume
5. Writing a volume label on the new volume
6. Writing header labels and a tape mark on the new volume

When multiple tape units are used, the EOV routine also checks to see if a next-plus-one volume is needed, and if the volume just written can be rewound and unloaded. If so, the EOV routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid; the next-plus-one volume label is not verified at this time. This lookahead or parallel mounting results in an IEC501E mount message.

Note: The IEC501E message has a type 2 descriptor code that causes the message to be retained on the screen.

If you do not want message retention, turn off the message retention attribute. You can accomplish this by either:

- Specifying RETAIN(NO) for the message id in an MPFLSTxx record
- Using a WTO message exit (an MPF specific exit or IEAVMXIT) to alter the message descriptor code or to request "do not retain this message"

For tape mounts, the IEC501E message remains on the screen until the mount is satisfied or until EOV has occurred on the previous volume. EOV deletes the IEC501E message and issues the IEC501A message.

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 tape mark is written following the data set trailer labels. If user trailer labels are to be written, the tape mark 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 exit is specified in the DCB exit list, the system can write as many as eight user trailer labels (UTL1 to UTL8).

Writing 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. The user exit for security verification may be taken, as described in “Open/EOV user exit for security verification” on page 48.

When positioning the second and subsequent volumes, the EOV routine does not use the data set sequence number. The EOV routine assumes that the data set continues at the beginning of the new volume.

RACF processing on the new volume

If the system-wide RACF tape protection option has been specified, RACF authorization checking will occur for the new volume. If the previous volume is RACF protected, the new volume must either be not defined to RACF (in which case EOV will define it to RACF as part of the previous volume's volume set profile), or the new volume must be defined as part of the same volume set profile as the previous volume. If the previous volume is not RACF protected, the new volume must not be RACF protected. If the above conditions are not met, the new volume is rejected when a nonspecific request is made, or the program is abnormally terminated when a specific request is made.

Special end-of-volume conditions

The following conditions signal the end of the volume:

- The drive signals to the system that it is nearing the end of the tape.
  - On a reel, the system detects the end-of-tape reflective marker.
  - On a cartridge, internal mechanisms sense the end of the tape.
- The application program issues an FEOV macro.

The EOV routine writes the trailer labels as described above. If the drive signal occurs while the system is 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.

If the system is writing header labels, and the drive signals that it is nearing the end of the tape, the system:

- Writes the new header label group
- Allows the user to write one record
- Writes the new trailer label group
- Performs volume switching

Closing an output data set

The close routine handles end-of-data-set processing on output tapes. When a write operation is the last operation that occurs before closing a data set (for OUTPUT, EXTEND, OUTIN, OUTINX, or INOUT) or when no output is written before closing (for OUTPUT, OUTIN, OUTINX or EXTEND), the close routine creates data set trailer labels. If the tape is a cartridge (not a reel), the close routine can check whether the expected number of blocks were written. See “Verifying block count” on page 10.

Writing data set Trailer labels

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

The close routine writes two tape marks following the trailer labels. If user labels are to be written, the tape marks follow the user trailer labels. If another data set is added to the tape (multiple data set organization), its HDR1 label overlays the second tape mark. The close routine passes the block identifier on a cartridge tape to the installation's tape end-on-volume exit. DFSMSrmm saves this information. See *z/OS DFSMS Installation Exits*.

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 does the following:

- Restores applicable control blocks to the conditions that existed when the checkpoint was taken.
- 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. If the volume is a reel:
  - 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 explained in "Positioning the volume to the data set" on page 39. The data set labels are not reprocessed.
  - Uses the block count shown in the DCB to reposition the tape to the proper record within the data set. This positioning is always performed in a forward direction. If the block count is 0 or a negative number, the tape remains positioned at the interrecord gap preceding the first record.
- If the volume is a cartridge, then the block identifier saved in the checkpoint is used to position the tape to the proper record.

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

Chapter 3. ISO/ANSI labels

This chapter describes the organization, format, and processing of labels designed according to the specifications of the following industry standards as understood and interpreted by IBM:

- Level 4 of ANSI X3.27-1978, *American National Standard Magnetic Tape Labels and File Structure for Information Interchange*
- Level 4 of ISO 1001-1979, *Information Processing: Magnetic Tape Labeling and File Structure for Information Interchange*
- Level 4 of FIPS 79
- Level 4 of ANSI X3.27-1987, *American National Standard Magnetic Tape Labels and File Structure for Information Interchange*
- Level 4 of ISO 1001-1986(E), *Information Processing: Magnetic Tape Labeling and File Structure for Information Interchange*

In this book, the term *Version 4* is used when referring to ANSI X3.27-1987 level 4 and ISO 1001-1986(E) level 4 standards.

In this book, the term *Version 3* is used when referring to ANSI X3.27-1978, ISO 1001-1979 and FIPS 79 standards.

The term *Version 1* is used when referring to the previous ISO 1001-1967 and ANSI X3.27-1969 standards.

Only volumes with Version 3 or Version 4 labels will be created by the system. Volumes with Version 1 labels will be accepted for input, but Version 3 or Version 4 checking will not occur.

ISO/ANSI support executes on a processor with a magnetic tape device that supports the specifications in:

- *Information Processing: 9 - Track, 12.7mm (0.5in) Wide Magnetic Tape for Information Interchange, 32rpmm (800rpi)*, ISO 1863.
- *Information Processing: 9 - Track, 12.7mm (0.5in) Wide Magnetic Tape for Information Interchange, 8 and 32rpmm (200 and 800rpi) NRZI, and 63rpmm (1600rpi), Phase Encoded*, ISO 1864.
- *Information Processing: 9 - Track, 12.7mm (0.5in) Wide Magnetic Tape for Information Interchange, 63rpmm (1600rpi), Phase Encoded*, ISO 3788.
- *American National Standard Recorded Magnetic Tape for Information Interchange, 800cpi, NRZI*, X3.22-1973.
- *American National Standard Recorded Magnetic Tape for Information Interchange, 1600cpi, PE*, X3.39-1973.
- *American National Standard Recorded Magnetic Tape for Information Interchange, 6250cpi, Group-Coding*, X3.54-1976.

Both *Information Processing: 9 -Track, 12.7mm (0.5in) Wide Magnetic Tape for Information Interchange, 8rpmm (200rpi)*, ISO 1862, and *American National Standard Recorded Magnetic Tape for Information Interchange, 200cpi, NRZI*, X3.13-1973, require a magnetic tape device not supported by z/OS. Tapes written according to those standards are not supported.

All data on a tape with Version 3 labels must be recorded in the 128 characters of basic ASCII. Also, all the data in a Version 4 tape label must be recorded in the 128 characters of basic ASCII. Do not use Version 3 or Version 4 volumes to contain raw binary, floating point, or packed decimal data. See "ASCII conversion" on page 82 for more information. Unlabeled tapes recorded in ASCII can be processed, as explained in Chapter 5, "Unlabeled tapes," on page 103.

Label definition and organization

ISO/ANSI labels are similar to IBM standard labels. The principal differences between ISO/ANSI labels and IBM standard labels are:

- ISO/ANSI labels are written in ASCII characters. IBM standard labels are written in EBCDIC, except on 7-track tape. IBM standard labels on 7-track tape are written in BCDIC.
- IBM standard labels are 80 bytes long. ISO/ANSI labels have a length of 80 bytes or more.
- A maximum of 9 user volume labels can appear in the beginning-of-volume group. Version 4 labels can also have VOL2–VOL9 appearing after VOL1 label in the beginning-of-volume group.
- An unlimited number of ISO/ANSI user labels can be placed at the beginning and end of a file, and they need not be sequentially numbered or lettered.
- The formats of the ISO/ANSI labels VOL1, HDR2, EOF2, and EOV2 are slightly different from the formats of the corresponding IBM labels.
- A maximum of 9 user EOF or EOV labels can appear in the file section label group.

The labels must be recorded in the subset of ASCII characters allowed by ISO/ANSI standards.

For Version 3 the subset of ASCII characters allowed by ISO/ANSI standards are:

- Uppercase alphabetic
- Numeric
- Space
- Special (specifically, ! " % & ' ( ) * + , - . / : ; < = > ?)

For Version 4 the subset of ASCII characters allowed by ISO/ANSI standards are the same as for Version 3, except for the addition of the underscore:

- Uppercase alphabetic
- Numeric
- Space
- Special (specifically, ! " % & ' ( ) * + , - . / : ; < = > ? _)

All fields in Version 3 or Version 4 system labels (VOL1, HDR1, HDR2, EOV1, EOV2, EOF1, and EOF2) are treated as containing meaningful data. This means alphanumeric fields (except Reserved for Operating System fields) are left-justified, with unused positions filled with space characters; Reserved for Future Standardization fields are filled with space characters; numeric fields are right-justified, with unused positions filled with zeros. Date fields may have a leading space character.

The first 4 characters of an ISO/ANSI tape label always identify the type of label:

Label Identifier
Label Definition

VOL1 - VOL9
Volume label set (optional; VOL2 - VOL9 not produced by z/OS)

UVL1 - UVL9
User volume labels (optional; *not* produced by z/OS)

HDR1
Data set header label 1

HDR2
Data set header label 2 (produced by z/OS, but optional for input)

HDR3 - HDR9
Optional (not produced by z/OS)

UHLa
User header labels (optional; unlimited number permitted)

EOV1
End-of-volume trailer label 1 (produced by z/OS, but optional for input)

EOV2
End-of-volume trailer label 2 (produced by z/OS, but optional for input)

EOV3 - EOV9
Optional (not produced by z/OS)

EOF1
End of data set trailer label 1 (produced by z/OS, but optional for input)

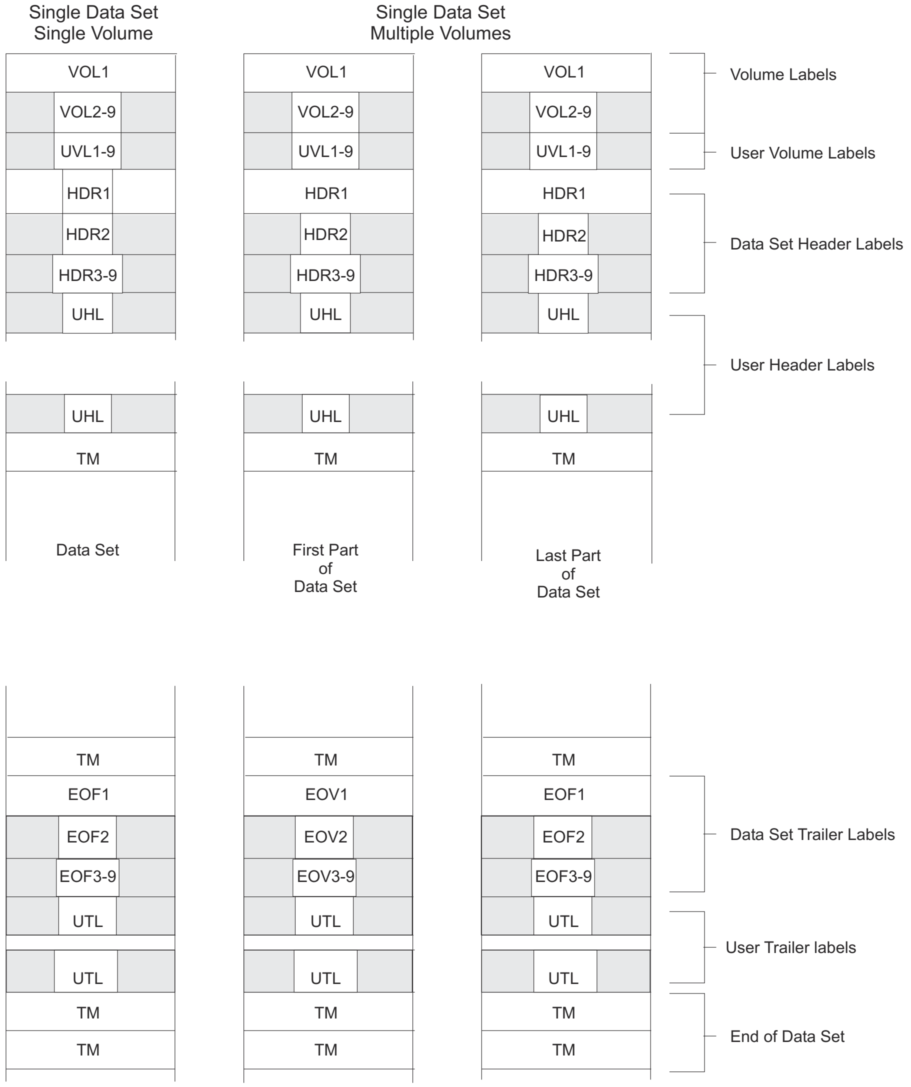
EOF2
End of data set trailer label 2 (produced by z/OS, but optional for input)

EOF3 - EOF9
Optional (not produced by z/OS)

UTLa
User trailer labels (optional; unlimited number permitted)

Note: The fourth character of the user header and trailer labels may be any valid ASCII character as defined above.

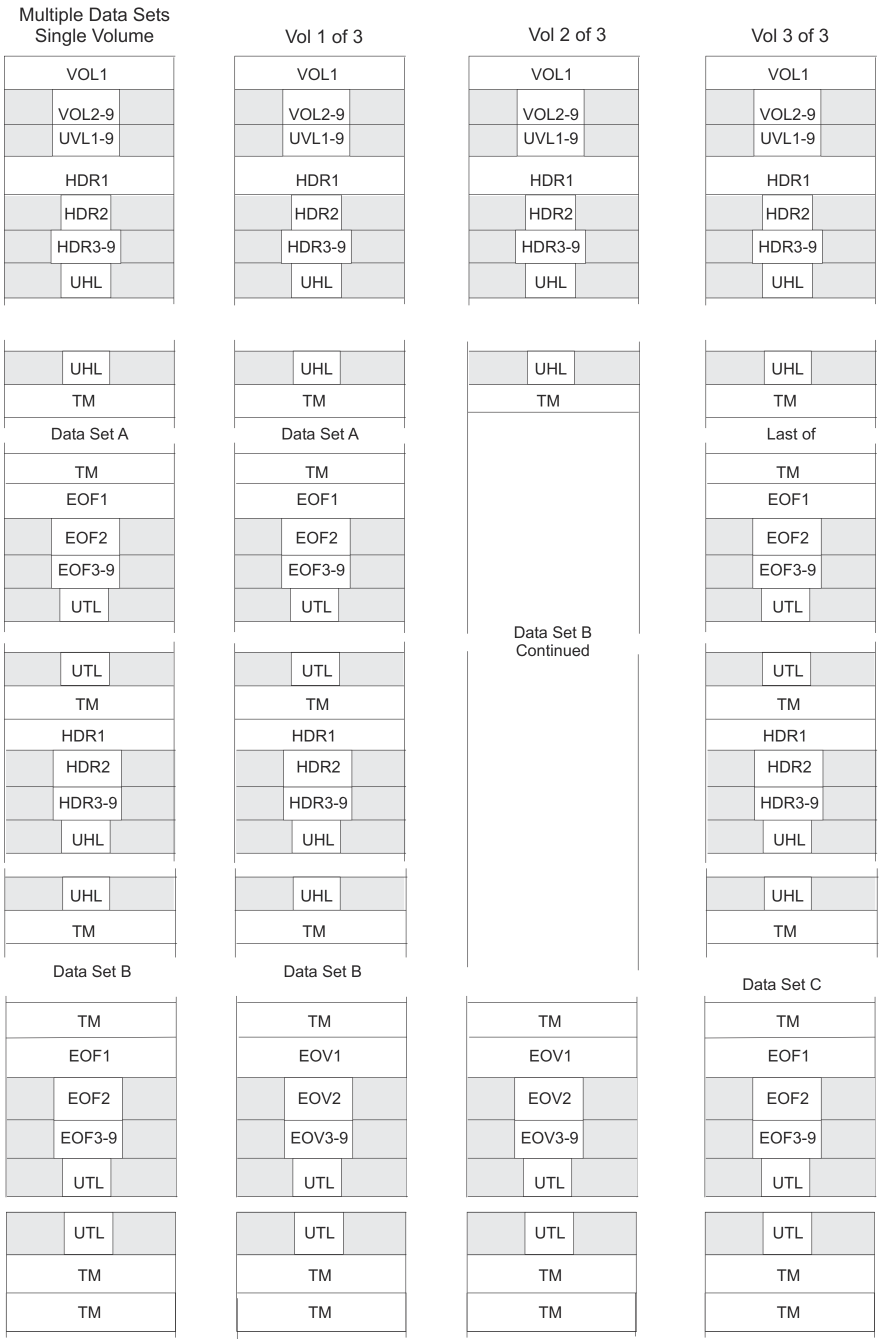
Figure 10 on page 56 and Figure 11 on page 57 show the position of the labels with various tape volume organizations. User labels (UHL, UTL) and volume labels (VOL2 - 9) are optional.



Single Data Set / Single Volume: The volume label is followed by the data set header labels and optional user header labels. The data set is preceded and followed by a tapemark. The data set trailer labels are identified as EOF and followed by 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.

Note: Shading indicates optional labels for ISO/ANSI labeled tapes.

Figure 10. Volume organization with ISO/ANSI 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. Each volume begins with a volume label. The last volume is organized the same as a multiple dataset / single volume layout except that the first data set begins on an earlier 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 two tapemarks.

Note: Gray or shading indicates optional labels for ISO/ANSI Labels (Multiple Data sets)

Figure 11. Volume organizations with ISO/ANSI labels (multiple data sets)

Volume label

The volume label (VOL1) is the first record on an ISO/ANSI labeled tape. It is at least 80 characters long. Although an ISO/ANSI label can exceed a length of 80 bytes, the excess is not within the scope of the standards and will be truncated by the operating system. All Version 3 and Version 4 labels written by the operating system routines will be 80 bytes in length, including those in which an original label greater than 80 bytes is rewritten (as in a density conflict).

The volume label identifies the volume and its owner and is used by the system to verify that the correct volume is mounted. It also protects the volume from unauthorized use.

Volume labels are created by the EDGINERS or IEHINITT utility programs, or by input from the operator during the Open or EOV routine when:

- A label type conflict is detected. A label type conflict occurs when the type of label requested differs from the type of label read at load point from the mounted volume.
- The first block of a volume cannot be read (except a RACF-protected volume is rejected if failure to read was because the tape format, such as a nonsupported density, was not recognizable by the control unit).

For additional information about the EDGINERS and the IEHINITT utility programs, see the *z/OS DFSMSrmm Implementation and Customization Guide* and the *z/OS DFSMSdfp Utilities*.

Volume labels (VOL2 - VOL9) are allowed but not created or processed by the operating system. They are checked only for valid label identification and correct sequencing.

User volume labels (UVL1 - UVL9) are allowed but not created or processed by the operating system. They are checked only for valid label identification and correct sequencing.

Data set header labels

ISO/ANSI header labels consist of a data set label 1 (HDR1) and, generally, a data set label 2 (HDR2). The HDR1 label is created by the system routines each time a data set is recorded on tape. A HDR1 label identifies and describes the data set and protects it from unauthorized use.

The HDR1 label is 80 characters long. Any characters after position 80 are not used for label checking or verification.

A HDR2 label is optional for ISO/ANSI labeled tapes. When present, it immediately follows the HDR1 label. The z/OS operating system produces a HDR2 label for output tapes; such a label on an input tape is treated similarly to an IBM HDR2 label. If the HDR2 label is produced by another system, the "reserved for operating system" fields are not processed.

Labels identified as HDR3 - HDR9 are allowed but only checked for valid label identification and correct sequencing (and only if they are among the header labels of a requested data set). z/OS does not create optional data set header labels beyond the second label.

User header labels

User header labels (UHLa) are optional, and there is no limit to the number that can be associated with a data set. They must contain only valid ASCII characters, but need not be sequentially numbered or lettered. These labels contain user-specified data that can be made available to your program for processing.

If you want the system to write user header labels or to make user header labels available to your program, you must specify AUL on the DD statement and the address of a user header label routine in the DCB exit list. (The exit list can address several user header label routines, that is, routines that process input user header labels and create output user header labels.) The DCB exit list is described in *z/OS DFSMS Using Data Sets*.

Data set Trailer labels

The data set trailer label 1 (EOV1, EOF1) is required for ISO/ANSI labeled tapes. The standard EOV1 and EOF1 labels duplicate the standard HDR1 label so that the tape can be read backward. The EOV1 and EOF1 labels are identical to a HDR1 label except that:

- The identifier is EOV or EOF instead of HDR.

- A block count is recorded in the first EOV1 or EOF1 label and is used on input to verify that all blocks of the data set are processed. The block count field in the HDR1 label contains zeros (which is what the EOF1/EOV1 block count is reduced to when the data set is read backward).

A data set can be extended by a job different from the one that started writing it. You might do that with DISP=MOD on the DD statement or with the OPEN macro options EXTEND, OUTINX or INOUT. On a Version 3 tape these are allowed only if the installation's label validation exit allows it. Extending a data set might result in fields such as these differing between headers and trailer labels: job name, step name, creation date, expiration date.

Eighty-character Version 3 or Version 4 EOV1 and EOF1 labels are created by the system when the data set is recorded on tape. EOV1 and EOF1 labels created by non-MVS systems must be up to 80 characters long; any characters after position 80 are not used for checking or verification.

Although they are optional, the EOV2 and EOF2 labels are produced by the operating system. These labels are treated similarly to IBM EOV2 and EOF2 labels. If the EOV2 or EOF2 labels are produced by another system, then the "reserved for operating system" fields are not processed.

Labels identified as EOV3-EOV9 or EOF3-EOF9 are allowed but checked for valid label identification and correct sequencing if they are among the trailer labels of a requested data set. The system does not create optional trailer labels beyond the second label.

User Trailer labels

User trailer labels (UTLa) are optional, and there is no limit to the number that can be associated with a data set. They must contain only certain ASCII characters, but need not be sequentially numbered or lettered. These labels contain user-specified data that can be made available to your program for processing.

If you want the system to write user trailer labels or to make user trailer labels available to your program, you must specify AUL on the LABEL parameter of the DD statement and the address of a user trailer label routine in the DCB exit list. The exit list can address several user trailer label routines, that is, routines that process input user trailer labels and create output user trailer labels. The DCB exit list (EXLST) is described in *z/OS DFSMS Using Data Sets*.

System-Managed mountable tape external labels

An external label is required for every tape cartridge in the tape library. This label contains both machine-readable and operator-readable identification characters that are used by the tape library's vision system to identify the volume serial number of the cartridge. In a manual tape library, a machine readable external label is not required, but is advisable for migration purposes.

Each label can have a maximum of six characters for identification purposes:

Uppercase A - Z
Numerics 0 - 9
Trailing blank spaces (non-service volumes only)

The external label must be applied within the indentation on the side of the cartridge; however, a larger information label, supplied by the customer, can be used on the top of the cartridge.

Tape marks

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

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

- A tape mark marks the end of the trailer label group. A second tape mark follows the trailer label group of the last data set on the volume, provided that the data set does not continue on another volume.

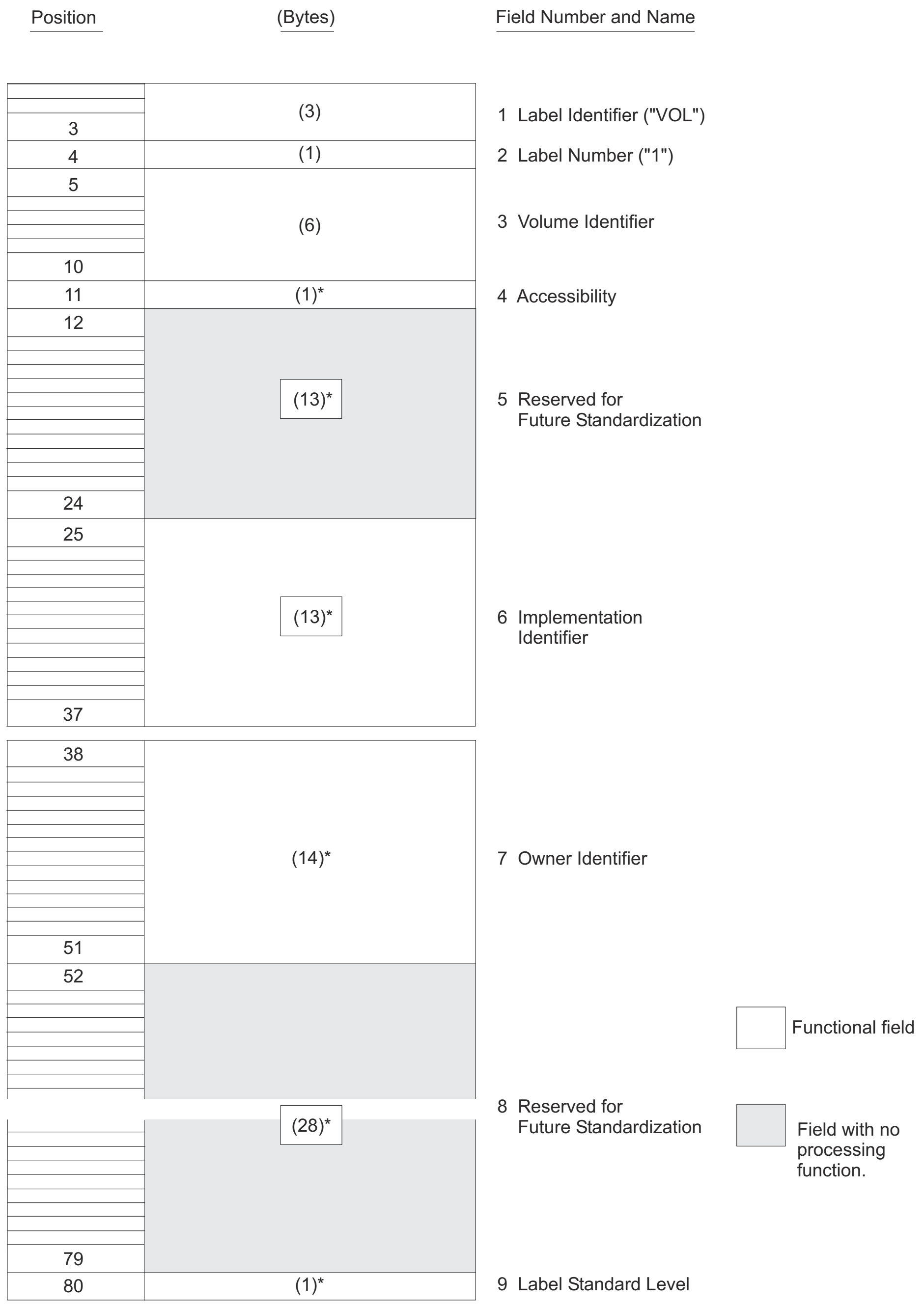
When the operating system is used to create a data set with IBM ISO/ANSI labels, it writes the necessary tape marks.

ISO/ANSI labels format

The ISO/ANSI standard volume label is 80 bytes or longer and is always used to identify the tape volume and its owner. It is always the first record on ISO/ANSI standard labelled tape. It is recorded in ASCII.

Format of the ISO/ANSI volume label (VOL1)

Figure 12 on page 61 shows the format of the ISO/ANSI volume label. The shaded areas represent fields that are recorded in the label, but not used or verified during processing. The contents and processing of each field of the label are described, as are differences between the ISO/ANSI volume label and the IBM volume label.



Position	(Bytes)	Field Number and Name
3	(3)	1 Label Identifier ("VOL")
4	(1)	2 Label Number ("1")
5 10	(6)	3 Volume Identifier
11	(1)*	4 Accessibility
12 24	(13)*	5 Reserved for Future Standardization
25 37	(13)*	6 Implementation Identifier
38 51	(14)*	7 Owner Identifier
52 79	(28)*	8 Reserved for Future Standardization
80	(1)*	9 Label Standard Level

* ISO/ANSI field differs from corresponding IBM field.

Figure 12. Format of the ISO/ANSI 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.

  The labels are created by the EDGINERS or IEHINITT utility programs.

  In the following situations, the Open or EOV routines create a Version 3 or Version 4 volume label based on specifications in your DD statement:

  - If you request a Version 3 or Version 4 output volume (AL, AUL) without specifying a volume serial number, and an NL, NSL, or SL labeled volume is mounted to satisfy your request
  - If you request a Version 3 or Version 4 output volume (AL, AUL) without specifying a volume serial number, and a volume that can be overwritten and recorded in the wrong density is mounted to satisfy your request (only if output is for the first data set on the volume).

For ISO/ANSI labeled tape volume requests, DFSMSrmm or the label anomaly installation exit routine or label editor installation exit routine can give permission for OPEN or EOV to write or rewrite the VOL1 label. See "ISO/ANSI volume label verification and editor routines" on page 79 for more information.

2—Label Number (1 byte)

- *Contents*: The relative position of this label within a set of labels of the same type; it is always 1 for the Version 3 or Version 4 volume label.
- *Processing*: Verified in conjunction with Field 1 to identify this label as VOL1.

3—Volume Identifier (6 bytes)

- *Contents*: A unique identification code, known in the system as the volume serial number, assigned through JCL, DFSMSrmm, or IEHINITT to the volume when it enters the system, or assigned by the operator when Open or EOV routines label the volume. This code might also appear on the external surface of the volume for visual identification. The code is typically numeric (000000 to 999999), but can be from 1 to 6 characters long. If the code is less than 6 characters, it is left-justified and padded with blanks.

  All national characters and some special characters in this field are rejected during open as invalid ASCII characters. For a list of the valid ASCII characters, see "Label definition and organization" on page 54.
- *Processing*: When the user specifies a volume serial number, it is in the JFCB. The system copies it to the UCB. When the system processes the labels, it compares the number in the UCB 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.
- *Difference from IBM Field*: The corresponding field in an IBM standard label is called the Volume Serial Number.

4—Accessibility (1 byte)

- *Contents*: A valid Version 3 or Version 4 character indicates that the RACHECK installation exits are entered and receive accessibility parameters, or the volume access exit will be entered in order to accept or reject the volume. A space indicates that the volume is authorized for access, unless RACF rejects the volume. All other characters cause the volume to be rejected if the volume is not defined to RACF.
- *Processing*: If this field contains any invalid Version 3 or Version 4 characters, then the volume is rejected by the system, unless the volume has been defined to RACF.
- *Difference from IBM Field*: The corresponding field in an IBM standard label is reserved and currently unused.

5—Reserved for Future Standardization (13 bytes)

- *Contents*: Reserved for possible future use and recorded as blanks.
- *Processing*: Not used or verified, except to check for all blanks. The EDGINERS and IEHINITT utility programs write blanks in this field. (Blanks are translated to ASCII space characters on output.)

6—Implementation Identifier (13 bytes)

- *Contents*: Reserved for possible future use and recorded as blanks.
- *Processing*: Not used or verified, except to check for all blanks. The EDGINERS and IEHINITT utility programs write blanks in this field. (Blanks are translated to ASCII space characters on output.)

7—Owner Identifier (14 bytes)

- *Contents*: Indicates a specific customer, person, installation, department, and so forth, to which the volume belongs. Any code or name is acceptable.
- *Processing*: Not used or verified, except to check for valid Version 3 or Version 4 characters. The EDGINERS and IEHINITT utility programs write the text specified by the user, and the open and EOV routines write the text specified by the operator. If the code is less than 14 bytes long, it is left-justified and the remainder of the field is padded with blanks. (Blanks are translated to ASCII space characters on output.)
- *Difference from IBM Field*: The corresponding field on an IBM standard label is 10 bytes long.

8—Reserved for Future Standardization (28 bytes)

- *Contents*: Reserved for possible future use and recorded as blanks.
- *Processing*: Not used or verified, except to check for all blanks. The EDGINERS and IEHINITT utility programs write blanks in this field. (Blanks are translated to ISCII/ASCII space characters on output.)

9—Label Standard Level (1 byte)

- *Contents*: 3 signifies that the tape is formatted according to Version 3 interchange standards. 4 signifies that the tape is formatted according to Version 4 interchange standards.
- *Processing*: The operating system always places 3 or 4 in this field on output. Version 1 tapes contain 1 in this field and are accepted for input processing. The system rejects any character other than 1, 3, or 4 in this field.
- *Difference from IBM Field*: This field is blank in IBM standard labels.

Format of ISO/ANSI data set label 1 (HDR1/EOV1/EOF1)

Figure 13 on page 64 shows the format of HDR1, EOV1, and EOF1. The shaded areas represent fields that the operating system writes in the label, but are not used or verified during processing. The contents and processing of each field of the label are described, as are differences between the ISO/ANSI labels and the IBM labels.

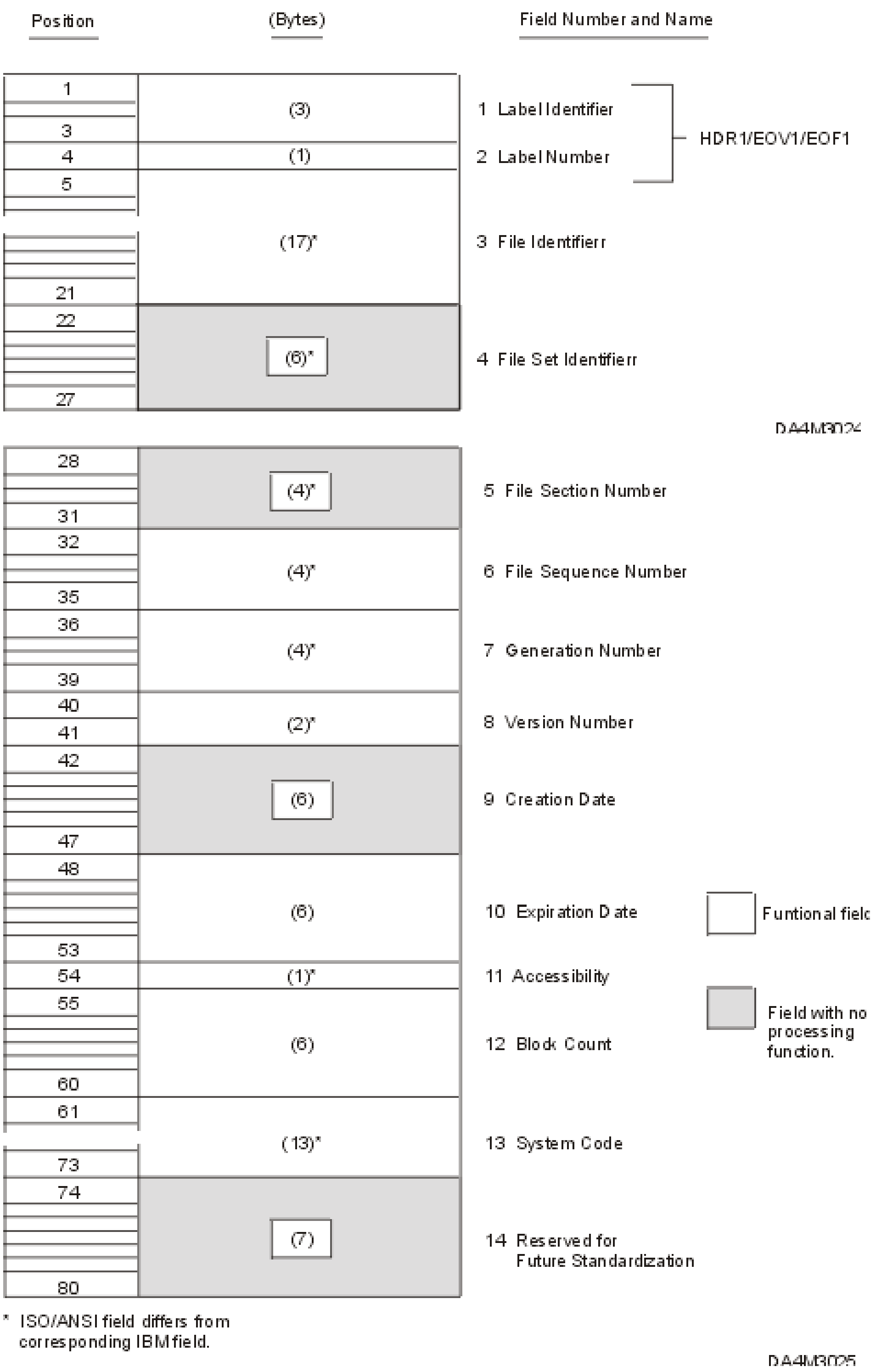


Figure 13. Format of ISO/ANSI header 1 and trailer 1 labels

1—Label Identifier (3 bytes)

- *Contents*: Three characters that identify the label are as follows:
  - **HDR**
    Header label (at the beginning of a data set)

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

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

- *Processing*: The system checks this field to verify that the record is an ISO/ANSI data set label.

  For input data sets, the system checks the label identifier to determine whether data set processing should continue. When the system finds an EOV label, it performs volume switching. When the system finds an EOF label, it passes control to the user's end-of-data routine or continues processing with a concatenated data set.

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

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

2—Label Number (1 byte)

- *Contents*: The relative position of this label within a set of labels of the same type; it is always 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—File Identifier (17 bytes)

- *Contents*: The rightmost 17 bytes of the data set name, *not* including the suffix .G*xxxx*V*yy* for generation data sets. Version 1 tapes will continue to be accepted for input with the suffix included as part of the file identifier. If the data set name is fewer than 17 bytes, it is left-justified and the remainder of this field is padded with ASCII space characters.

  If the name contains embedded spaces or other special characters, you must enclose the name in apostrophes on the DD statement that requests this data set. *z/OS MVS JCL User's Guide* 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 comparison ensures that the correct data set is being processed. For Version 3 only, the file identifier is also compared to the names of other data sets on the volume during open positioning to ensure against duplicate names. See “Processing the HDR1 label” on page 88 for more information.

  For output, the data set name in the existing label is verified in conjunction with password protection to determine whether the existing data set can be overwritten. If password protection is not specified, then the data set name is checked for valid Version 3 or Version 4 characters.

  For input and output the first file data set identifier is compared with the last 17 characters of the first file data set name recorded in the DFSMSrmm control data set. If there is no match, the volume is rejected. This check ensures that the correct volume is mounted.

  When creating labels for a new data set, the user-specified data set name is obtained from the JFCB and recorded in this field.
- *Difference from IBM Field*: The corresponding field in an IBM standard label is called the Data Set Identifier.

4—File Set Identifier (6 bytes)

- *Contents*: The volume serial number of the tape volume containing the data set. For multivolume data sets, this field contains the serial number of the first volume of the aggregate created at the same time.

  The content requirements are the same as for the volume serial number field in the volume label.
- *Processing*: If this field has inconsistent values in the volumes in a volume set as described in Figure 3 on page 16 and Figure 4 on page 17, the system writes a warning message. The warning indicates that one of the volumes might be the wrong volume. The message number is IEC709I. When creating labels, the serial number is obtained from the UCB and recorded in this field.

- *Difference from IBM field*: The corresponding field on an IBM standard label is called the Data Set Serial Number.

5—File Section Number (4 bytes)

- *Contents*: A number (0001 to 9999) that indicates the order of the volume within the multivolume aggregate. This number is always 0001 for a single volume data set.
- *Processing*: If this field does not have a value one greater than the previous volume in a volume set as described in Figure 3 on page 16 and Figure 4 on page 17, the system writes a warning message IEC709I. This message warns you that the volumes might be out of order. When creating labels, the open routine writes 0001 in this field if the data set sequence number is 1; otherwise the open routine copies the corresponding value from an earlier label on the volume. The EOV and close routines write a number one higher than on the previous volume.
- *Difference from IBM Field*: The corresponding field on an IBM standard label is called "Volume Sequence Number".

6—File Sequence Number (4 bytes)

- *Contents*: A number (0001 to 65535) that indicates the relative position of the data set within a multiple data set group (aggregate). This number is always 0001 for a single data set organization.
- *Processing*: This number in the first HDR1 label on the tape is referred to when the open routine positions the tape. If this number in the first HDR1 label and the requested data set sequence number in the JFCB are both greater than 1, then the logical data set sequence number in the UCB is set to the number in the label. Otherwise, the logical data set sequence number in the UCB is set to 1.

  When creating labels, the open and close routines obtain the user-specified data set sequence number from the JFCB (0 is changed to 1). The EOV routine propagates the number.

7—Generation Number (4 bytes)

- *Contents*: If the data set is part of a generation data group, this field contains a number from 0001 to 9999 indicating the absolute generation number (the first generation is recorded as 0001). If the data set is not part of a generation data group, this field contains 0001.
- *Processing*: A nonnumeric or all zero value is not valid, and can cause the label validation exit to be entered unless validation has been suppressed.

  When creating labels, the system checks the JFCB to determine whether the data set is part of a generation data group. If so, the generation number is obtained from the last part of the data set name in the JFCB. Otherwise, this field is recorded as 0001.

8—Version Number (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 ASCII zeros.
- *Processing*: The system always records this field as zeros. For a version level other than zero, you must specify the absolute generation and version numbers as part of the data set name when creating or retrieving a data set.

9—Creation Date (6 bytes)

- *Contents*: Date when allocation begins for creating the data set. The date is format cyyddd, where:

  ```
    c = century (blank=19; 0=20; 1=21; etc.)
   yy = year (00-99)
  ddd = day (001-366)
  ```

  Note: The `century` code gives the first two digits of the year, not the actual century. For example, a blank that translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- *Processing*: Not used or verified, except to check for proper format. When the system creates labels, the date is obtained from the JFCB. If a data set is allocated via JCL, then the creation date is the date

when allocation begins for the job step responsible for creating the data set. If a data set is allocated dynamically, then the creation date is the date when allocation begins for the data set.

10—Expiration Date (6 bytes)

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

  ```
    c = century (blank=19; 0=20; 1=21; etc.)
   yy = year (00-99)
  ddd = day (001-366)
  ```

  Note: The `century` code gives the first two digits of the year, not the actual century. For example, a blank that translates into 19 indicates a year in the 1900s, not in the nineteenth century.

- *Processing*: For input, not used or verified, except to check for proper format. For output, the expiration date in the existing label is compared to the current date. If the date in the label is later than the current date, then the operator receives a message and has the option of using the tape or mounting another. If other data sets are on the same volume, then the system checks the expiration date of the immediately preceding data set. If the previous data set's expiration date is earlier than that of the output data set, then the label validation exit is entered (unless label validation has been suppressed). If any other data sets follow on the same volume, then they are considered expired on the same day. The expiration dates of scratch SMS-managed tape volumes, both automatic and manual tape libraries, are ignored. DFSMSrmm can override the expiration date or reject the volume without operator intervention.

  If you enter 99365 or 99366, the system retains your data sets permanently. Do not use those dates as expiration dates. Use them as no-scratch dates only.

  When creating labels, the system obtains the expiration date from the JFCB. If you did not specify a retention period or expiration date, then the expiration date is recorded as zeros, and the data set expires. DFSMSrmm uses the expiration date to update information its control data set. For data sets without an expiration date, DFSMSrmm assigns one using the DFSMSrmm default retention period, but the label field is left as zeros.

11—Accessibility (1 byte)

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

  Valid Version 3 or Version 4 characters:
  If the volume is not RACF protected, the file access exit is entered.

  Space
  No data set access protection.

  1
  Password protection. Additional identification of the data set is required before it can be read, written, or deleted. (Ignored if volume is RACF defined.) Password protection can be specified in the PASSWORD subparameter of the LABEL keyword of JCL.

  3
  Password protection. Additional identification of the data set is required before it can be written or deleted. (Ignored if volume is RACF defined.) This can be specified in the NOPWREAD subparameter of the LABEL keyword of JCL.

  Other character
  Protected volume. No access is possible under the operating system, unless the volume has been defined to RACF and is authorized for use by RACF.

- *Processing*: For input, the system inspects this field on a single volume data set, on each concatenated data set, and on each volume of a multivolume data set. If password protection is specified in this field, the system verifies the password furnished by the operator or TSO terminal user and sets a security indicator in the JFCB. If a valid Version 3 or Version 4 character is specified and the volume is not defined to RACF, the file access exit is entered (the IBM-supplied exit routine will reject the volume). If a character other than an ASCII character is encountered, the DCB will not be opened and no further processing will take place.

For output, the system inspects this field in the existing HDR1 label. If 1 or 3 is specified with the system code "IBMZLA", the existing data set cannot be overwritten until the system verifies the password and the data set name in Field 3 of this label. Password checking is bypassed if the volume is defined to RACF. If you specify a data set name different from the one in Field 3, and the data set is the first one on the first or only volume, the operator must remove the tape and mount a scratch tape, even though you requested a specific volume. If the data set is not the first one on the volume or this is not the first volume of a multivolume data set, the job is abnormally terminated. If a valid Version 3 or Version 4 character is specified and the volume is not defined to RACF, the file access exit will be entered (the IBM-supplied exit routine will reject the volume).

When the system creates labels, the user's request for security is determined from the indicator in the JFCB for password processing or from a JCL ACCODE value. Password codes override ACCODE values if they are both specified.

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 000000.
- *Processing*: The DCB block count (in the device-dependent area of the DCB) is increased as the data set is read. The final DCB count is compared with the count in the trailer label at end of data or end of volume. If the counts do not agree, a user exit entry in the DCB exit list determines whether processing continues or abnormally terminates. A block count discrepancy causes processing to abnormally terminate if the appropriate user exit entry is not provided.

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

  When the system creates labels, the block count in the header label is set to 000000. The block count in the trailer label is obtained from the DCB during close and EOV label creation.

  If the data set was created with an EXCP DCB that had no device-dependent area, and the volume was not rejected, the block count is written as zero and is not verified. For tapes with Version 3 labels, a data set opened without at least a 4-word device-dependent area in the DCB will cause the label validation exit to be entered with a symmetry error, unless validation has been suppressed. The default of the exit is to reject the volume.

13—System Code (13 bytes)

- *Contents*: A unique code, IBMZLA, that identifies the system.

  Note: Version 1 tapes produced by MVS contain OS/360 or OS/370 as a system code.
- *Processing*: On input, the field is checked to determine how succeeding labels are to be processed (the field determines whether Field 3,"Record Format", and Field 6,"Reserved for Operating System", in the second header label will be processed). On output, the operating system supplies the IBMZLA code.

14—Reserved for Future Standardization (7 bytes)

- *Contents*: Reserved for possible future use; contains blanks.
- *Processing*: Not used or verified, except to check for blanks. When creating labels, the system writes blanks in this field. Blanks are translated to ASCII space characters on output.

Format of ISO/ANSI data set label 2 (HDR2/EOV2/EOF2)

Figure 14 on page 69 shows the format of HDR2, EOV2, and EOF2. The shaded areas represent fields that the operating system writes in the label, but are not used or verified during processing. The contents and processing of each field of the label are described, as are differences between ISO/ANSI labels and IBM labels.

If the labels are produced by MVS, they are treated like IBM standard header 2 and trailer 2 labels. If the labels are produced by another system, the "Reserved for Operating System" and "Record Format" fields are not used.

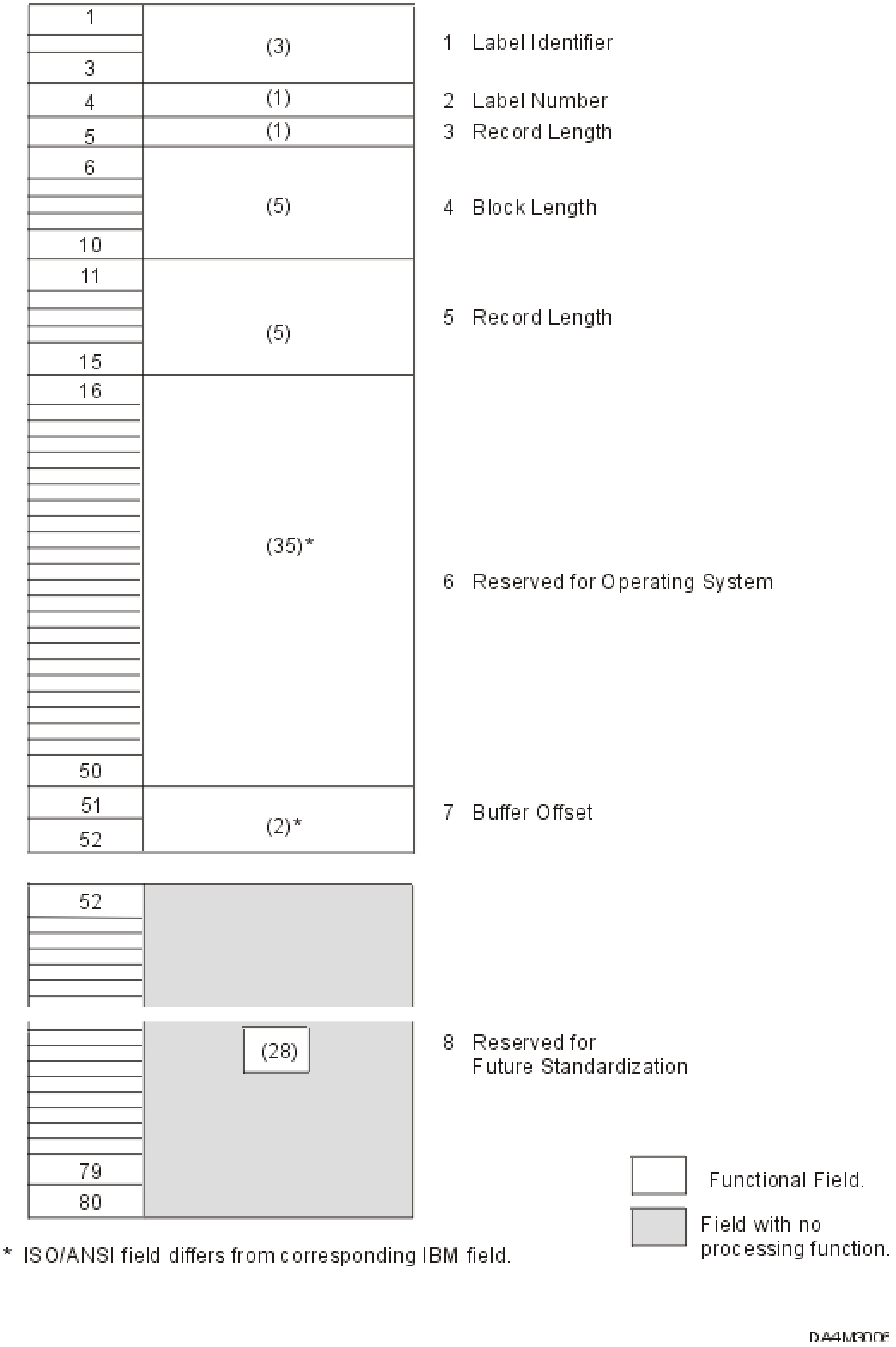


Figure 14. Format of ISO/ANSI header 2 and trailer 2 labels

1—Label Identifier (3 bytes)

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

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

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

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

- *Processing*: The system checks this field to verify that the record is an ISO/ANSI data set label.

  For input data sets, the system checks the label identifier to determine whether data set processing should continue. When the system finds an EOV label, it performs volume switching. When the system finds an EOF label, it passes control to the user's end-of-data routine or continues processing with a concatenated data set.

  If the DD statement specifies OPTCD=B for an input data set, the system accepts either EOV or EOF as the trailer label identifier, and the identifier is not used to determine whether a volume switch is necessary. If more volumes are available, the system performs the switching. If no volumes are available, the system passes control to the user's end-of-data routine or continues processing with a concatenated data set.

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

2—Label Number (1 byte)

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

3—Record Format (1 byte)

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

  - **F**
    Fixed length

    D
    Variable length

    S
    Spanned

  Note: RECFM=U (undefined length) is accepted for *input* from a Version 1 tape. If specified for input or output for Version 3 or Version 4 tapes, the label validation exit is entered.

  For detailed information about record formats, see *z/OS DFSMS Using Data Sets*.

- *Processing*: For input, the open routine obtains the record format from this label, converts it to a bit code, and records it in the JFCB (if the JFCB field is 0). Then the record format in the JFCB is recorded in the DCB (if the DCB field is 0). Note that this is a merging process in which existing specifications in the JFCB and DCB cannot be overridden.

  Record format will not be accepted from the label if the block attribute (in Field 6) is not applicable to MVS; in this case, record format must be specified in JCL or in the DCB.

  When creating labels, the system's OPEN and EOV routines do a reverse merge. 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 Figure 2 on page 4.

4—Block Length (5 bytes)

- *Contents*: A number from 18 to 2 048 for Version 3 or from 18 to 32 760 for Version 4 indicates the block length (including buffer offset and padding) in bytes.

Note: The 2 048-byte limit on block length is part of the Version 3 standard. Larger blocks (up to 9 999 bytes for format D and format S records and 32 760 bytes for format F records) may be specified with the agreement of the interchange parties. However, for tapes with Version 3 labels, exceeding the 2 048-byte limit causes the label validation exit to be entered. The maximum block length for Version 4 is 32 760 bytes.

The system can determine the optimum block size when creating tape data sets. For more information, see *z/OS DFSMS Using Data Sets*.

Interpretation of the number depends on the associated record format in Field 3, as follows:

- **Format F**
  Maximum block length. If the block attribute in field 6 contains a blank, this field is also the minimum block length.

  Format D
  Maximum block length that includes the 4-byte length indicator in the records and the optional block prefix.

  Format S
  Maximum block length that including the optional block prefix, plus one or more pairs of 5-byte segment control words and segments.

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

5—Record Length (5 bytes)

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

    Format D
    Maximum record length that includes the 4-byte length indicator in the records.

    Format S
    Maximum record length that excludes all the 5-byte segment control words that describe the record. If the record length is larger than 99 999, this field is 0.
- *Processing*: The number in the label is converted to binary and merged with the appropriate fields in the JFCB and DCB. The merging process is the same as for the record format code in Field 3 of this label.

6—Reserved for Operating System (35 bytes)

- *Contents*: If the label is produced by this operating system (the system code in the first header label is IBMZLA), the content and format of this field are similar to the content and format of the ten fields that follow the record length field in an IBM standard label. If the label is produced by another system, the content and format of this field are optional, and the field is not processed by the operating system. The fields produced by the operating system are:
  - Tape Density (1 byte)
    - *Contents*: A code indicating the recording density of the tape; the code is equivalent to the DEN parameter value on the DD statement (for the DEN parameter values, refer to "Tape characteristics" on page 11). If DCB=DEN was not specified in JCL, this field is set to blanks.
    - *Processing*: If DCB=DEN was not specified in JCL, this data is merged in the JFCB for input. When the system creates labels, the information for this field is obtained from the JFCB.
  - Data Set Position (1 byte)
    - *Contents*: A code indicating a volume switch is as follows:
      - **0**
        No volume switch has occurred.

1
A volume switch previously occurred.

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

– Job/Job Step Identification (17 bytes)

- *Contents*: Identification of the job and job step that created the data set. The first 8 bytes contain the name of the job; the ninth byte is a slash (/), and the last 8 bytes contain the name of the job step.
- *Processing*: Not used or verified. When the system creates labels, it sets this field as described above.

– Tape Recording Technique (2 bytes)

- *Contents*: Recorded as blanks for all tapes except 7-track tapes. A 9-track tape can only be recorded in odd parity with no translation.
- *Processing*: Not used or verified.

– Control Characters (1 byte)

- *Contents*: A code indicating whether a control character set was used to create the data set, and the type of control characters used:
  - **A**
    Contains Version 3 or Version 4 control characters.

    ƀ
    Contains no control characters.
- *Processing*: The specification in the label is converted to a bit code and merged with the Record Format field in the JFCB if ANSI carriage control was not specified as part of DCB=RECFM in JCL.

– Buffer Alignment Block (1 byte)

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

– Block Attribute (1 byte)

- *Contents*: A code indicating the block attribute used to create the data set:
  - **B**
    Blocked records

    ƀ
    Records not blocked.
- *Processing*: The specification in the label is converted to a bit code and merged with the record format field in the JFCB. The merging process is the same as for the record format code in Field 3 of this label.

– Coded Character Set ID (CCSID) (5 bytes)

- *Contents*: A code indicating the CCSID used to create the user data on the Version 4 tape or blank.
- *Processing*: The CCSID used to convert data to and from the tape. If blank, the data is written in ASCII characters. See Appendix D, “Equivalent ASCII and EBCDIC codes,” on page 121, and “CCSID conversion” on page 83.

– Reserved (6 bytes)

- *Contents*: Recorded as blanks.
- *Processing*: Reserved for definition by IBM. Currently not used or verified.

7—Buffer Offset (2 bytes)

- *Contents*: The length of the block prefix (from 0 to 99).
- *Processing*: Used to determine the length of an optional prefix that may be a part of a physical block on tape. The version of the prefix for variable and spanned record formats is known as a block descriptor word (BDW). A BDW is always 4 bytes long and contains the block length of the physical record it describes, including the BDW. The BUFOFF=L operand informs the system that the prefix is an MVS BDW. The prefix is not made available as part of the data read into storage by the queued access method. For more information about BDWs, see *z/OS DFSMS Using Data Sets*.
- *Difference from IBM Field*: This field is not present in IBM standard labels.

8—Reserved for Future Standardization (28 bytes)

- *Contents*: Reserved for possible future use; recorded as blanks. The blanks are translated to ASCII space characters on output.
- *Processing*: Not used or verified. When creating labels, the system writes blanks in this field.

Format of ISO/ANSI user header and Trailer labels (UHL/UTL)

Figure 15 on page 73 shows the format of UHL and UTL labels.

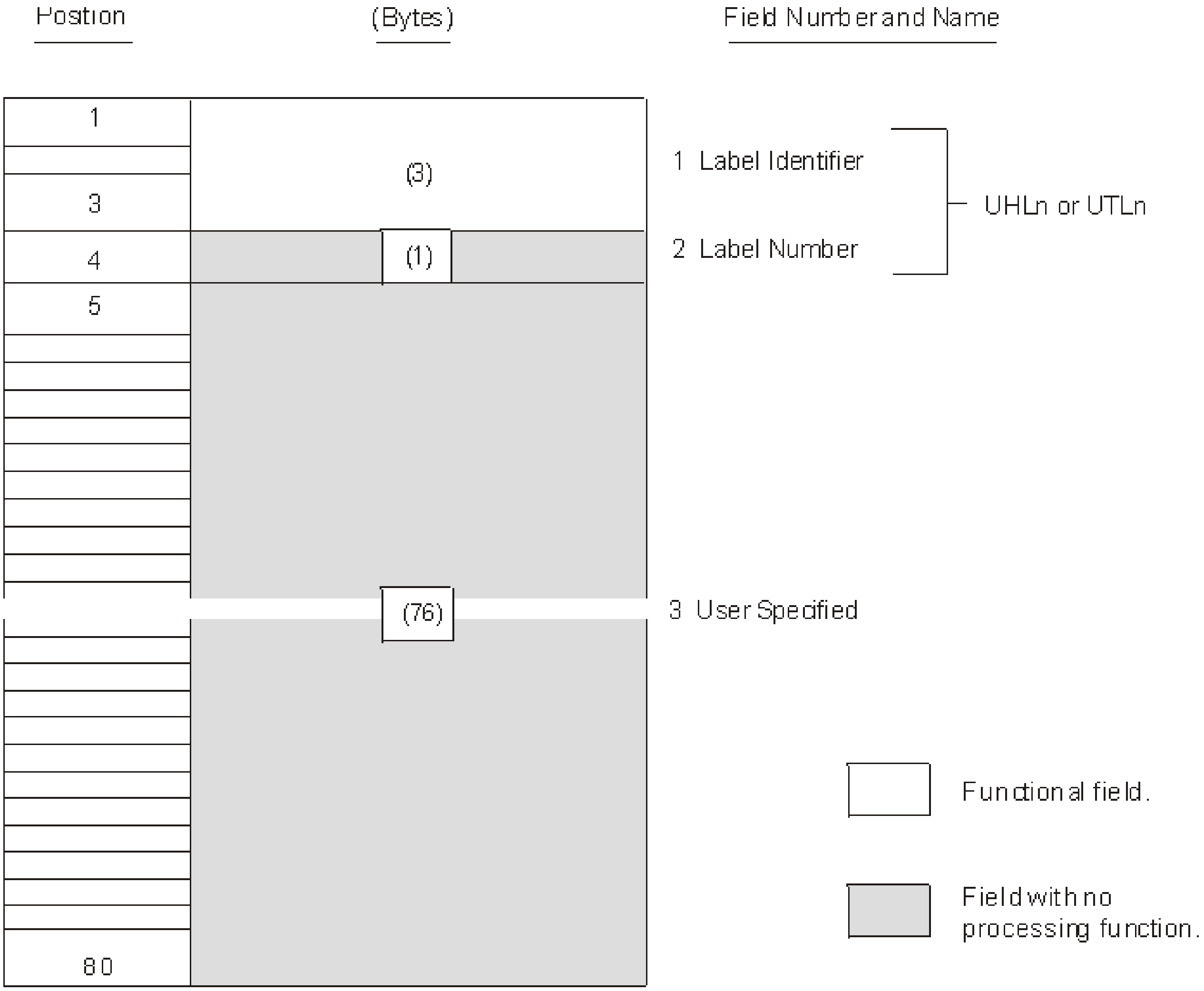


Figure 15. Format of ISO/ANSI user labels

1—Label Identifier (3 bytes)

- *Contents*: Three characters that identify the label are as follows:
  - **UHL**
    User header label (at the beginning of a data set)

    UTL
    User trailer label (at the end-of-volume or end-of-data set).
- *Processing*: This field is read to verify that the record is a user label. The system accepts either UHL or UTL.

2—Label Number (1 byte)

- *Contents*: Any valid Version 3 or Version 4 character.
- *Processing*: This field is checked by the operating system for a valid Version 3 or Version 4 character during creation of a user header label (UHL) during open/EOV if validation has not been suppressed. If an invalid character is detected, the label validation exit is entered.
- *Difference from IBM Field*: This field can contain only numeric 1 to 8 for IBM standard user labels. A maximum of 8 user header or trailer labels is supported for conventional IBM standard user labels, but any number of user labels can be written for Version 3 or Version 4 tapes, and they may be lettered or numbered in any order.

3—User Specified (76 bytes)

- *Contents*: Specified by the user, but must be valid Version 3 or Version 4 characters.
- *Processing*: Specified in the DCB exit list. This field is checked as explained for Field 2.

Other file header labels

Other file header labels (HDR3 - HDR9, EOF3 - EOF9, EOV3 - EOV9) and user volume labels (UVL1 - UVL9) and volume labels (VOL2 - VOL9) are not created by the operating system. Because these labels may appear on magnetic tapes created by other systems, the operating system will accept them as input. Those labels are ignored during label processing and are not placed on output tapes. Ignored labels are checked only for a valid label identifier and proper label sequence; no checks are made for ignored labels encountered during positioning to the requested data set.

Figure 16 on page 75 shows a hypothetical input tape from a non-IBM system and a corresponding output tape produced by the operating system. The user volume label, the additional header label, and the additional trailer label are not placed on the output tape.

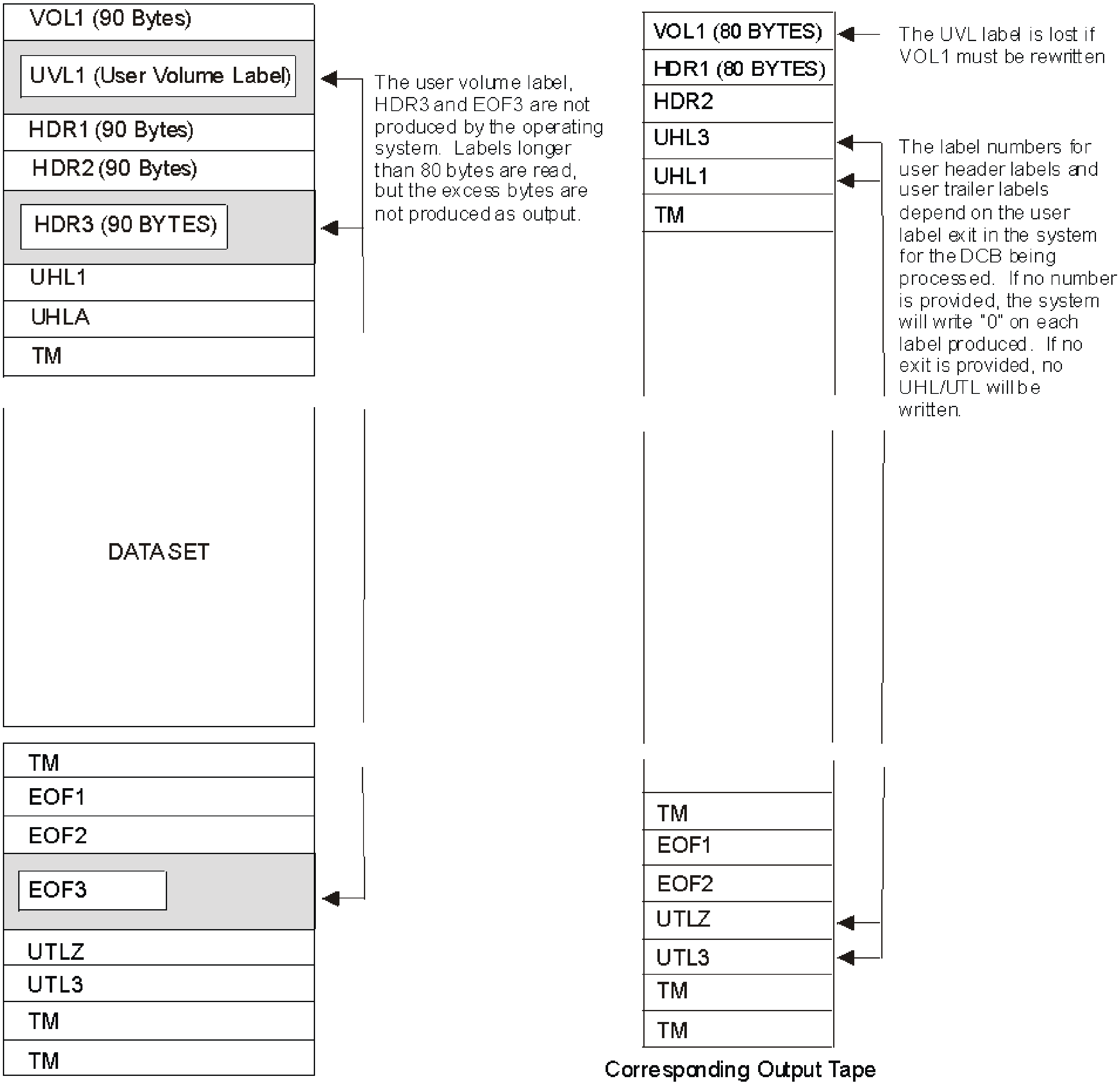

VOL1 (90 Bytes)
UVL1 (User Volume Label)
HDR1 (90 Bytes)
HDR2 (90 Bytes)
HDR3 (90 BYTES)
UHL1
UHLA
TM
DATA SET
TM
EOF1
EOF2
EOF3
UTLZ
UTL3
TM
TM
The user volume label, HDR3 and EOF3 are not produced by the operating system. Labels longer than 80 bytes are read, but the excess bytes are not produced as output.
VOL1 (80 BYTES)
HDR1 (80 BYTES)
HDR2
UHL3
UHL1
TM
TM
EOF1
EOF2
UTLZ
UTL3
TM
TM
The UVL label is lost if VOL1 must be rewritten
The label numbers for user header labels and user trailer labels depend on the user label exit in the system for the DCB being processed. If no number is provided, the system will write "0" on each label produced. If no exit is provided, no UHL/UTL will be written.
Corresponding Output Tape

Figure 16. Example of a ISO/ANSI tape from a non-IBM system and a corresponding output tape

Overview of ISO/ANSI label processing

Label processing is handled by the I/O support routines of data management (open, EOV, and close). Label processing consists of the following basic functions:

- Converting input labels from ASCII to EBCDIC and converting output labels from EBCDIC to ASCII
- Checking the labels on input tapes to
  - Ensure that the correct volume is mounted
  - Identify, describe, and protect the data set being processed
  - Attempt to ensure maximum correctness and consistency of data sets and their labels
  - Identify compatibility conflicts with Version 3 or Version 4 standards
- Checking the existing labels on output tapes to

  - Ensure that the correct volume is mounted
  - Prevent overwriting of vital data
  - Identify compatibility conflicts with Version 3 or Version 4 standards
- Creating and writing new labels on output tape
- For volumes in an automated tape library, comparing the external volume serial number to the magnetically recorded internal label.

These processing functions are summarized in Table 5 on page 76. The table shows the specific labels that are processed for each function and which routines perform the functions.

Although the default of each IBM-supplied installation exit is to reject the volume, the exits can be modified to do additional label processing. See “ISO/ANSI installation exits” on page 83 for more information.

All volumes will be written with Version 3 or Version 4 labels. For information about using tapes with other than Version 3 or Version 4 labels, see “Compatibility with non-version 3 or non-version 4 volumes” on page 85.

Table 5. ISO/ANSI label processing by Open/Close/EOV routines

	Volume Label			**Header Labels“1” on page 77**				**Trailer Labels“1” on page 77**			
Processing	**VOL1**	**VOL 2–9**	**User Volume Labels**	**HDR1**	**HDR2**	**HDR3-9**	**UHL**	**EOF1 or EOV1**	**EOF2 or EOV2**	**EOF3-9 or EOV3-9**	**UTL**
First or only volume“2” on page 77											
Checks labels on input tape	Open	Ignored“3” on page 77	Ignored“3” on page 77	Open	Open	Ignored“3” on page 77	Open	EOV	Bypassed	Ignored“3” on page 77	EOV
Checks existing labels on output tape before overwriting	Open	Ignored“4” on page 77	Ignored“4” on page 77	Open	Not read	Not read	Not read	Not read	Not read	Open	
Writes new labels on output tape	Open or user“5” on page 77	Not written	Not written	Open	Open	Not written	Open	Close or EOV	Close or EOV	Not written	Close or EOV
Compares the volume serial number on the external volume label to the serial number in the tape volume label“7” on page 77	Open	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77
Second or subsequent volumes“6” on page 77											
Checks labels on input tape.	EOV	Ignored“3” on page 77	Ignored“3” on page 77	EOV	Bypassed	Ignored“3” on page 77	EOV	EOV	Bypassed	Ignored“3” on page 77	EOV
Checks existing labels on output tape before overwriting.	EOV	Ignored“4” on page 77	Ignored“4” on page 77	EOV	Not read	Not read	Not read	Not read	Not read	Not read	Not read
Writes new labels on output tape.	EOV or user“5” on page 77	Not written	Not written	EOV	EOV	Not written	EOV	Close or EOV	Close or EOV	Not written	Not written
Compares the volume serial number on the external volume label to the serial number in the tape volume label“7” on page 77	EOV	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77	N/A“8” on page 77

Table 5. ISO/ANSI label processing by Open/Close/EOV routines (continued)

	Volume Label			Header Labels“1” on page 77				Trailer Labels“1” on page 77			
Processing	**VOL1**	**VOL 2–9**	**User Volume Labels**	**HDR1**	**HDR2**	**HDR3-9**	**UHL**	**EOF1 or EOV1**	**EOF2 or EOV2**	**EOF3-9 or EOV3-9**	**UTL**
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 when the user program has turned on the DCB unlike attributes bit (DCBOFPPC). See *z/OS DFSMS Using Data Sets*. 3. Label sequence is checked but contents are ignored. 4. Operator must give Open/EOV permission to overwrite VOL2–9 or ULV labels. 5. User can create the label with the EDGINERS or IEHINITT utility programs, or a user program. 6. Includes the first volume of concatenated data sets when the user has not turned on the DCB unlike attributes bit. 7. Applies only to tape library. 8. Not Available.											

Creating a volume label

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

The IBM-supplied utility programs EDGINERS or IEHINITT write the following labels in ASCII code:

1. A VOL1 label with the volume serial number, accessibility code, and owner identification that you specify. You cannot specify any other fields of the VOL1 label.
2. A dummy HDR1 label with '0001' in the file section number, file sequence number, and generation number fields; "IBMZLA" in the system code field; and a space in the accessibility field (allowing unlimited access). Reserved fields contain zeros, with a leading space in the date fields.
3. A tape mark.

A tape initialized with these labels should not be confused with an Version 3 or Version 4 tape, which requires one data set (the data set can be empty).

Version 3 or Version 4 standards require label symmetry around an empty data set when a volume is initialized. The labels written by EDGINERS and IEHINITT are accepted by the system routines that produce symmetrical labels. The HDR1 label is updated with system data, the single tape mark is overwritten with a HDR2 label containing data set characteristics, and a tape mark is written to complete the beginning-of-volume and beginning-of-file label groups. A set of trailer labels are written when the data set is closed, even if no data was written in the data set.

The IEHINITT utility program can write a VOL1 label on a labeled, unlabeled, or blank tape; it makes no checks to see what data, if any, exists on the tape. Therefore, IEHINITT does not check for password or RACF security protection; it does not create, modify, or delete RACF profiles of RACF-defined volumes. However, the IEHINITT utility does call Security Authorization Facility (SAF), which may result in a RACF call for tape volumes in an ATLDS. The IEHINITT utility can create either Version 3 or Version 4 tape labels. Detailed procedures for using the program are described in *z/OS DFSMSdfp Utilities*.

The EDGINERS utility program validates the mounted volume before writing a VOL1 label on a labeled, unlabeled, or blank tape. DFSMSrmm uses RACF to protect its resources with RACF profiles. DFSMSrmm also uses RACF to control actions that are taken on RACF tape profiles through the RMM parmlib option TPRACF. DFSMSrmm can also provide control of RACF at the volume pool level. It can also be used to create Version 3 or Version 4 tape labels. For details on how to do this, see the*z/OS DFSMSrmm Implementation and Customization Guide*.

Methods other than the EDGINERS or IEHINITT utility programs can be used to write VOL1 labels. You can use a card-to-tape program, or the system programmer can replace the IBM-supplied volume label editor routine with one that writes VOL1 labels. See “ISO/ANSI volume label verification and editor routines” on page 79 for more information. On some old models of tape drives that use reels, some data or a tape mark should already exist on the tape; otherwise, the tape control unit may read through the entire reel of blank tape looking for a record or tape mark.

For all IBM magnetic tape subsystems, the VOL1 label is rewritten by the open or EOV routines if *all* of the following conditions are met:

- If density conflict has occurred or an ISO/ANSI routine conflict has occurred, then permission must be granted by the installation's label anomaly exit, the installation's label editor routine, DFSMSrmm or the system operator.
- OUTPUT, OUTIN, or OUTINX is specified in the OPEN macro.
- The tape is positioned to the first data set on the volume.
- Either of these conditions are true:
  - The data set is not password protected.
  - The volume is RACF protected, the system-wide RACF tape protection option has been specified, and the user has ALTER authorization.
- The volume does not have VOL2 - 9 or UVL labels. If it does, the operator must give permission to rewrite the VOL1 label because the VOL2 - 9 and UVL labels will be overwritten.

If you request output to the first data set on a Version 3 or Version 4 output volume (AL, AUL), and the tape that you are allocated is recorded in the wrong density and cannot be read or in the wrong ISO/ANSI version level, then the open or EOV routine can rewrite the VOL1 label in the density or version you specify.

This allows you to make nonspecific requests for output tapes and allows the operator to mount any available scratch tape. However, if the system-wide RACF tape protection option has been specified, then the volume is rejected because it cannot be verified that it is not a RACF-protected volume.

If you request output to any data set other than the first data set on the volume and a density or a version conflict occurs, then all further processing is done in the density or version of the existing VOL1 label (unless the IBM-supplied volume label editor routines are replaced with installation-written routines that do otherwise).

Note: If the densities are mixed, the tape will not meet the specifications of Version 3 or Version 4 standards.

If you request an output volume for an Version 3 or Version 4 labeled (AL, AUL) tape and the mounted volume is an NL, NSL, SL, or SUL tape, then the open or EOV routine creates a VOL1 label after checking authorization for access and the expiration date. The routine sends a message to the console operator requesting serial number and owner information (see Figure 12 on page 61, Fields 3 and 7). The label anomaly exit, volume label editor exit and DFSMSrmm can optionally prevent the creation or the destruction of standard labels.

The system programmer can indicate whether to create a Version 3 or Version 4 label by specifying a parameter in the DEVSUP*xx* PARMLIB member. For details see *z/OS MVS Initialization and Tuning Reference*.

The system uses the following hierarchy to determine which version of labels will be written when opening the first file of the first volume of an ISO/ANSI tape data set for output:

- Maintain version level in VOL1 label if output DISP=MOD, or for EXTEND, or if not the first data set on the volume.
- Maintain version level on VOL1 label if extending data set to a new volume.
- Use version level specified in the volume mount installation exit. For details, see *z/OS DFSMS Installation Exits*.
- Use the specified 'Forced' version level in the DEVSUPxx member. For details, see *z/OS MVS Initialization and Tuning Reference*.
- Use the existing version level as specified in the VOL1 label of Version 3 or Version 4.
- Use the specified version level in the DEVSUPxx member. For details, see *z/OS MVS Initialization and Tuning Reference*.

- Use Version 3 as the default.

ISO/ANSI volume label verification and editor routines

If you specify that an input or output tape has a Version 3 or Version 4 label, the operating system checks for the Version 3 or Version 4 volume label at the beginning of the tape.

Because of conflicting label types or conflicting tape characteristics, various error conditions can occur during the verification of the first record. Under some error conditions, the tape is accepted for use. Under other 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 label anomaly exit routine and to a *volume label editor routine*; your installation can use IBM-supplied routines or it can supply its own routines.

The IBM-supplied volume label editor routines determine the discrepancies between the requested tape and the mounted tape and, if necessary, pass control to the appropriate system routine to create or destroy labels. If DFSMSrmm is controlling the volume, the DFSMSrmm label editor routine controls the creation or destruction or standard labels using RACF profiles in the FACILITIES class. Installation-supplied routines can perform other functions.

IBM recommends using tape management exit routines instead of the volume label editor routines.

For detailed information about these and other available exit routines, see *z/OS DFSMS Installation Exits*.

Protecting data

The accessibility fields in the VOL1 and HDR1 labels indicate whether a volume and data set are protected against unauthorized use. Version 3 or Version 4 volumes can be protected by means of one of the following:

- RACF.
- IBM-supplied Version 3 or Version 4 installation exits described in "ISO/ANSI installation exits" on page 83. These exits can be replaced by installation-written exit routines.
- Data set password protection.

  Note: In a IBM system-managed library, data set password protection is not supported.
- A combination of the installation exits and password protection.

Version 1 input volumes are protected by either RACF or data set password protection.

Note: All checking for authorization is bypassed if security processing is suppressed. This can occur, for example, when the program properties table entry for the job step program is marked to suppress security checking. Only the system programmer can update the program properties table. For information about the program properties table, see *z/OS MVS Initialization and Tuning Reference*.

RACF allows you to establish access requirements for both tape data sets and tape volumes. To protect data on tape, you can do either or both of the following:

- Control access to the tape volumes.
- Control access to individual tape data sets on the tape volumes.

RACF protection at the volume level overrides RACF protection at the data set level. For more information on how to activate these levels of RACF protection and how they interact with each other and with your own tape management system, see the *z/OS Security Server RACF Security Administrator's Guide*. DFSMSrmm supports RACF protection, but not password protection. For information about DFSMSrmm and RACF, see *z/OS DFSMSrmm Implementation and Customization Guide*.

The following principles apply to RACF protection at the volume level:

- ALTER access authority is required to create or destroy the VOL1 label.
- READ access authority is required to open the volume for input (open options INPUT or RDBACK). Note that if your program uses the INOUT option of OPEN and the DD statement has LABEL=(,,,IN), the system treats it as the INPUT option.

- UPDATE access authority is required to open the volume for output (open options OUTPUT, EXTEND, INOUT, OUTIN, or OUTINX).

The user can open the volume to read or write if the tape volume is defined to RACF, the user has UPDATE access authority, and PROTECT=YES has not been specified in the JCL.

The request fails if the tape volume is defined to RACF, the user has UPDATE authority, PROTECT=YES has been specified in the JCL, and the tape is not a RACF scratch volume.

If the tape volume is defined to RACF and the user has READ but not UPDATE access authority, or if the user has UPDATE access but PROTECT=YES has been specified in the JCL and the volume is a RACF scratch tape volume, the system does not grant the user access to read until it has ensured that the user can not write on the tape. The user cannot access the volume until one of the following conditions is met:

- **Hardware Protection.** If the write-enable ring has been removed from the tape reel or the write-protect tab has been set to disable writing on the tape cartridge, the tape volume cannot be written on. The system safely permits the user to access the tape to read. This hardware protection cannot be circumvented by software.
- **Logical write-protection.** If the write-protect tab on an IBM magnetic tape cartridge is set to enable writing, the system issues a hardware command to prevent writing on that cartridge. If the command succeeds, the system safely allows the user to access the tape to read. An unauthorized program cannot bypass this combination of hardware and software protection.
- **IEC.TAPERING.** Your installation may choose to depend on a tape management system to prevent accidentally overwriting tapes with unexpired data. Typically, a tape management system only allows volumes with no unexpired data to be opened for output. DFSMSrmm provides facilities to prevent accidental overwriting of non-scratch tape volumes. The IEC.TAPERING support facilitates the operation of tape management systems because it allows all volumes to remain write-enabled (by the ring in the volume or the switch on the cartridge). This eliminates the need for operator intervention.

  If the write ring or the cartridge tab is set to enable writing, the system checks if the user is authorized for read to the IEC.TAPERING profile in the RACF FACILITY class. If the user does have this authority, the system grants the user access.

  ⚠ **Attention:** If you use the IEC.TAPERING support to allow users to read from tapes that are enabled for writing, when the users are only authorized to read, the system software cannot prevent knowledgeable users from writing on any files on the tapes.

- **Operator Intervention.** If none of the preceding conditions are met, the system requires the operator to intervene and prevent writing on the volume. The system demounts the tape and issues a message asking the operator to remove the write-enable ring from the tape reel or change the switch on the tape cartridge. After the operator remounts the tape, the system continues to protect the volume from unauthorized writing by repeating the preceding checks, beginning with the check for hardware protection. This check continues until one of the previously mentioned conditions is met.

If the tape volume is not defined to RACF, access is granted and processing continues. For an overview of RACF protection for tape volumes, see *z/OS Security Server RACF Security Administrator's Guide*. For information on how DFSMSrmm can help you manage RACF security for your tape volumes, see *z/OS DFSMSrmm Implementation and Customization Guide*.

For more information on data set password protection, see *z/OS DFSMSdfp Advanced Services*.

Volume accessibility

Version 3 or Version 4 Volumes: If the volume is RACF protected, and the volume security processing has not been suppressed, the RACHECK installation exits are entered to accept or reject the volume (by a return code from RACHECK). If the VOL1 accessibility field contains a valid Version 3 or Version 4 character, the RACHECK parameter list is initialized to address the ISO/ANSI exit parameter list (IECIEPRM) and the accessibility code. Otherwise, the RACHECK installation exits are passed zeroed pointers for the ISO/ANSI parameters. If RACF accepts the volume but the ISO/ANSI parameters to the exit are zero, the validation suppression exit is entered.

If RACF is not available, or the volume is not defined to RACF, OPEN/EOV checks the accessibility field in the VOL1 label. If the field contains a valid Version 3 or Version 4 character, the volume access exit is entered to accept or reject the volume (by a return code in the exit parameter list). If the field contains a space, the validation suppression exit is entered. All other characters cause the volume to be rejected.

The VOL1 accessibility code can be changed when the VOL1 label is changed by the EDGINERS or IEHINITT utility programs or by the operator during open. See "Volume label" on page 57.

Version 1 Volumes: First, OPEN/EOV checks the accessibility field in the VOL1 label. If the field contains a space character, the volume is defined to RACF, and volume security processing has not been suppressed, then the RACHECK installation exits are entered to accept or reject the volume (by a return code from RACHECK). Access to the volume is allowed if the field contains a space character, but the volume is not defined to RACF, or the volume security processing has been suppressed. Each data set on the volume can be individually protected. The volume is rejected if the field contains anything other than a space character.

The VOL1 accessibility code can be changed when the VOL1 label is changed by the EDGINERS or IEHINITT utility programs, or by the operator during open. See "Volume label" on page 57.

Data set accessibility

If a volume is RACF protected, and access to the volume was checked when the volume was verified (by RACHECK processing), then data set accessibility fields are not checked; therefore, any data set on the volume can be accessed. If a volume is not RACF protected, then the system inspects the accessibility field in the HDR1 label of the requested data set. For a multivolume data set, the system inspects the HDR1 label on each volume. The accessibility codes of any concatenated data sets are also inspected.

Version 3 or Version 4 Volumes: 1 or 3 in the data set accessibility field with a system code of "IBMZLA" indicates an MVS password-protected data set; a space character in the accessibility field allows unlimited access; and a valid Version 3 or Version 4 character causes the file access exit to be entered for authorization processing. The file access exit is also entered when a new data set is added to an output volume and if the first character of the JCL ACCODE keyword is a valid character. Any other character causes the volume to be rejected.

If the accessibility field of a data set being opened contains a valid character or a space, no attempt is made to check the accessibility field of any other data sets that may exist on the volume; therefore, the other data sets can be overlaid by new output. If accessibility checking for multiple data sets is required, it must be done by an installation-written file access exit. Never place an access-protected data set on the same volume with unprotected data sets because no exit is entered for unprotected data sets.

An HDR1 accessibility code of valid character can be specified by the ACCODE parameter of JCL. ACCODE can also be specified for dynamic allocation (SVC 99 key). Changes to an HDR1 accessibility code of a valid character can be monitored by an installation-written file access exit. Password codes override ACCODE values if both are simultaneously specified.

Version 1 Volumes: 1 or 3 in the data set accessibility field indicates a password-protected data set.

A space character in the accessibility field allows unlimited access, and no attempt is made to check the accessibility field of any other data sets that may exist on the volume; therefore, the other data sets may be overlaid by new output.

Input processing for password-protected data sets

The system asks the operator or TSO user to enter the correct password for input processing of a password-protected data set. The password is verified in the system's password data set. This password data set contains the data set name, the password, and a protection-mode indicator. The data set name for Version 3 or 4 generation data sets does not include the .G*xxxx*V*yy* suffix. The protection-mode indicator is set to permit either read/write or read-only operations. Processing is terminated if either of the following situations occur:

- The operator or TSO terminal user does not supply the correct password in two attempts.
- The password record for the data set to be opened does not exist in the password data set.

z/OS DFSMSdfp Advanced Services describes the protection feature and discusses how to create and maintain the password data set.

Note: Password protection for a tape data set in an IBM automated tape library, or manual tape library, is not supported. In such tape libraries, the only security protection for the tape data set is via RACF.

Output processing for password-protected data sets

The system compares the data set name shown in the HDR1 label with the data set name specified in the DD statement for output processing of a password-protected data set. For generation data sets on a Version 3 or 4 volume, the comparison of names does not include the generation and version numbers.

If the names are not the same, processing is terminated unless the data set is the first one on the first or only volume. In this case, even if you specify a specific volume, the operator must demount the tape and mount a new scratch tape. If the names are the same, the system requests the operator or TSO user to key in the required password. The password is verified in the system's password data set in the same manner as described above for an input data set. The read/write protection mode is necessary for output data sets.

Two restraints are placed on creation of password-protected data sets:

- When creating a password-protected data set following an existing password-protected data set, supply the password of the existing data set. The accessibility code of "1" or "3" must be the same in both the existing and the new data set. This is true even if the volume is RACF defined, or resides in an IBM automated tape library or a manual tape library.
- When creating a multivolume, password-protected data set, the second and successive volumes will also be verified. Verification consists of ensuring that the data set name in the JFCB (for Version 3 volumes, not including generation and version numbers, if present) is the same as the data set name in the password record. Verification also ensures that the protection-mode indicator allows writing to the data set. This verification is bypassed if the volume resides in an IBM automated tape library or a manual tape library.

Deleting a Password-Protected Data Set: If a password-protected data set is deleted, the HDR1 accessibility field must be set to a space character before the volume can be written on again. The HDR1 accessibility field can be set by using either the EDGINERS or IEHINITT utility programs, or a user program to relabel the volume.

ISO/ANSI installation and use characteristics

This section summarizes the main characteristics of installing support for and using ISO/ANSI Version 3 or Version 4 volumes.

ASCII conversion

The system performs ASCII conversion on all ISO/ANSI labels, all user data on Version 1 and 3 tapes and user data on Version 4 tapes when Coded Character Set Identifier (CCSID) conversion is not done. The system also performs ASCII conversion on unlabeled tapes if you specify OPTCD=Q in the DCB or DD statement. The system performs CCSID conversion if the application program or the data set has the CCSID option.

The IBM-supplied conversion routine converts ASCII 7-bit code and is designed to support Version 3 or Version 4 standards. The system forces OPTCD=Q during the open routine to cause ASCII conversion during processing of any volume mounted as LABEL=AL.

Note: If the input data includes raw binary, floating point, or packed decimal fields, then any bytes containing values that convert into more than 7 bits are converted to X'1A'. This results in a permanent loss of data.

Additional information on ASCII 7-bit code and tables showing the relationship of EBCDIC and ASCII 8-bit code is given in Appendix D, “Equivalent ASCII and EBCDIC codes,” on page 121.

CCSID conversion

Version 4 allows user data written in any national or user defined character standards. The character standards are specified by the CCSID parameters on the JOB/EXEC and DD statements. For details, see *z/OS DFSMS Using Data Sets*. If BUFOFF=L is in effect with CCSID conversion, the access method does ASCII conversion on the block prefix and CCSID conversion on the rest of the user data. The Version 4 standards require this distinction.

Volume initialization

Version 3 or Version 4 volumes are initialized with the EDGINERS or IEHINITT utility programs, or by the operating system in the case of certain label conflicts during mount verification of a volume. Volume initialization is discussed under "Creating a volume label" on page 77.

ISO/ANSI installation exits

Four ISO/ANSI *installation exits* are provided, as defaults, for a Version 3 or a Version 4 volume:

- Volume access
- File access
- Label validation
- Label validation suppression

A fifth installation exit, WTO/WTOR, can be written (or modified, if one has already been written) by your installation to convert ISO/ANSI non-Version 3 to Version 3 labels.

All the default installation exit routines are supplied in a module containing a single CSECT (IFG0193G, alias IFG0553G), in SYS1.LPALIB. A copy of the source code for the module is contained in member ANSIEXIT of SYS1.SAMPLIB.

The default routines, except the validation suppression exit, reject the volume. Your system programmer can replace any of the IBM-supplied exit routines with your own, installation-written, exit routines. The default routines can be modified or replaced to perform I/O (such as overwriting a label), change system control blocks, and mount or demount volumes. The return code from the exits can be modified to request continued processing.

For detailed information about these and other available exit routines, see *z/OS DFSMS Installation Exits*.

Volume access

Version 3 or Version 4 volumes can be protected by either RACF or the *volume access exit*. The volume access exit is entered during open or EOV if a volume is not RACF protected and the accessibility field in the volume label contains an ASCII valid Version 3 or Version 4 character. The exit is bypassed if volume security checking has been suppressed. See "Protecting data" on page 79 for more information.

File access

Version 3 or Version 4 data sets can be protected by the *file access exit*, data set password protection, or a RACF data set profile. The file access exit is entered after positioning to a requested data set if the accessibility field in the HDR1 label contains an ASCII valid Version 3 or Version 4 character and the volume is not RACF protected. Likewise, the exit is entered when a data set is written to an output volume if the first character of the JCL ACCODE is also a valid character. See "Protecting data" on page 79 for information on data set password protection and RACF.

Label validation

Labels and program specifications for labels and file structure are checked to ensure that a tape format does not violate Version 3 or Version 4 standards. The label validation exit is entered if any of the following requirements are not met:

- Labels must contain only certain ASCII characters. For a list of the characters, see "Label definition and organization" on page 54.
- Label fields must contain properly aligned data. For more information, see "Label definition and organization" on page 54.
- Labels that frame a data set must be symmetrical:
  - DISP=MOD is not allowed for extending an existing Version 3 data set or an existing non-IBM format Version 4 data set; however, it can be specified to create a new data set.
  - An open option for EXTEND, OUTINX, or INOUT is not allowed in Version 3 or non-IBM format Version 4.
  - An EXCP DCB used for output must contain at least a 4-word device dependent area.

  For more information, see "Data set Trailer labels" on page 58, "Creating a volume label" on page 77, and *z/OS DFSMS Installation Exits*.
- Block size must not exceed 2 048 bytes for Version 3 or 32 760 bytes for Version 4. See "Format of ISO/ANSI data set label 2 (HDR2/EOV2/EOF2)" on page 68.
- Variable length records must be specified as Format-D, not as Format-V. Undefined-length records (Format-U) are allowed only for Version 1 input volumes; they are not allowed for Version 3 or Version 4 volumes.
- Duplicate data set names, including names for generation data sets, are not allowed on the same Version 3 volume. See "Processing the HDR1 label" on page 88.
- Expiration dates for successive data sets on a Version 3 volume must not be in ascending sequence. See "Expiration date on existing label" on page 96.

Label validation suppression

The validation suppression exit allows the option of suppressing label validation. The validation suppression exit is entered during open or EOV if:

- Volume security checking has been suppressed
- The volume label accessibility field contains an ASCII space character
- RACF accepts a volume and the accessibility field does not contain a valid Version 3 or Version 4 character.

WTO/WTOR installation exit

For ISO/ANSI tape volumes, DFSMS V1R1 and above supports output only to Version 3 and input from either Version 1 or Version 3. DFSMS/MVS V1R5 supports output to both Version 3 and Version 4 and input from Version 1, Version 3, or Version 4. If a label version conflict is detected during an output request to the first data set on a volume, the system uses WTOR to prompt the installation operator to obtain information for rewriting the volume label as a Version 3 or a Version 4 label. If the installation does not want the operator to provide the necessary label information, the WTO/WTOR installation exit or message processing facility (MPF) can provide the information. See "Compatibility with non-version 3 or non-version 4 volumes" on page 85. The WTO/WTOR installation exit (IEAVMXIT) and MPF are described in *z/OS MVS Installation Exits*.

RACHECK installation exits

The RACHECK preprocessing and postprocessing installation exits can act on Version 3 or Version 4 volume accessibility and label validation. These exits are discussed under "Protecting data" on page 79.

Generation data groups

For Version 3 or Version 4, generation and version numbers are *not* part of the data set name, as contained in the file identifier field of the HDR1 label. See "Format of ISO/ANSI data set label 1 (HDR1/EOV1/EOF1)" on page 63.

Spanned records

Spanned records (Format-S) are allowed and can be up to 16 776 192 bytes long; however, the actual length of logical records to be processed depends upon the amount of virtual storage that can be acquired. See “Format of ISO/ANSI data set label 2 (HDR2/EOV2/EOF2)” on page 68 and *z/OS DFSMS Using Data Sets*.

Block padding

A fixed-length logical record (Format-F) containing all ASCII circumflex characters (X'5E') will signal an end-of-block condition. A variable-length record (Format-D) beginning with a circumflex character, even though the rest of the record does not contain circumflex characters, signals an end-of-block condition.

Checkpoints

The CHKPT macro service prevents a checkpoint from being taken on as ISO/ANSI tape.

Compatibility with non-version 3 or non-version 4 volumes

For input, volumes with Version 1 labels are accepted, but without Version 3 or Version 4 checking. No Version 3 or Version 4 installation exits will get control, and the volume label accessibility code field will not be passed to either the RACF pre-processing or post-processing exits, ICHRCX01 or ICHRCX02. All acceptable version types must have the labels written in 7-bit ASCII.

For output, all volumes will be written with Version 3 or Version 4 labels. A volume with Version 1 labels, or any other volume with an 80-character volume label containing the ASCII characters VOL1 in the label identifier field, might be mounted for output, but only if the first data set is being written. (However, extending an existing data set, such as by specifying DISP=MOD in the DD statement, is not allowed for Version 3 or for non-IBM format Version 4.) The volume label is rewritten and the new header and trailer labels are written in Version 3 or Version 4 format. This converts the volume to Version 3 or Version 4.

Note: This conversion requires intervention by the console operator (in response to message IEC704A during open/EOV). Therefore, with careful operational procedures, you can avoid creating an unexpected Version 3 or Version 4 volume.

Processing differences between Version 1 and Version 3 or Version 4

Version 3 or Version 4 processing for certain characteristics differs from Version 1 support. Because of these differences, job streams that run under Version 1 might fail or produce different results when running under Version 3 or Version 4. Therefore, Version 1 users should check their job streams and modify them, if necessary, before installing Version 3 or Version 4 support.

The characteristics that distinguish Version 3 or Version 4 processing from Version 1 are:

- Label validation
- Generation data groups
- Spanned records
- Block padding
- Checkpoints
- Compatibility with non-Version 3 or non-Version 4 volumes

Processing Version 3 or Version 4 volumes on earlier MVS systems

When a Version 3 or a Version 4 volume is mounted on an earlier MVS system, the following restrictions apply:

- Input generation data sets are not recognized because the .G*xxxx*V*yy* suffix is not part of a Version 3 or a Version 4 file identifier.
- Spanned record format is not supported.

- Data set characteristics recorded in the second header or trailer label of an input volume are not made known to the application program; instead, they must be specified either in JCL or in the DCB. (This is because the Version 3 or Version 4 MVS system code, IBMZLA, is not recognized by earlier systems.)
- A data set having a header label created by a non-MVS system with 1 or 3 in the accessibility field is treated as an MVS password-protected data set unless the volume is defined to RACF.
- A valid Version 3 or Version 4 character in the accessibility field of a header or volume label causes the volume to be rejected.
- Any output does not meet the specifications of Version 3 or Version 4 standards.

Tape marks

The tape mark requirements for ISO/ANSI interchange tapes are:

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

When creating a data set with Version 3 or Version 4 labels, the system routines write the necessary tape marks.

Opening an input data set

Before OPEN verifies the volume, OPEN will abnormally terminate the program if the system-wide RACF tape protection option is specified and the DD statement specifies PROTECT=YES without previously opening the DD statement for output processing.

If you specify ISO/ANSI labels, the first record on the tape must be at least 80 bytes and must contain the ASCII identifier VOL1 in the first 4 bytes. The volume label can be Version 1, Version 3, or Version 4. If the record is over 80 bytes long, the excess characters are ignored.

The system also checks for label type conflicts (the type of label requested differs from the type of label read at load point from the mounted volume), version compatibility conflicts (the label is not Version 1 or Version 3 or Version 4), and density conflicts (the density specified by the user for the data set is not compatible with the density of the VOL1 label). In the case of a label type conflict or version compatibility conflict, the volume is rejected. In the case of a density conflict, the density of the VOL1 label is used for all further processing of the data set.

If system security checking should be bypassed, the validation suppression exit is entered. Otherwise, if the system-wide RACF tape protection option has been specified, RACF access authorization to the volume is checked for READ authority. If a tape with Version 3 or Version 4 labels is not RACF protected and the VOL1 accessibility field contains a valid Version 3 or Version 4 character, the volume access exit is entered. For more details of RACF and accessibility checking, see "Protecting data" on page 79.

For a tape with Version 3 or Version 4 labels, (after access to the volume is authorized, the request is checked for possible symmetry violations). Because INOUT can result in asymmetrical labels, specifying this option will cause the label validation exit to be entered (unless it is Version 4 or validation has been suppressed). In addition, unless validation has been suppressed, the VOL1 label is checked for conditions not allowed by Version 3 or Version 4 standards. If an invalid condition is found, the label validation exit is entered.

Volume serial number

The system 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 you specify. 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, RDBACK, EXTEND or OUTINX.

Serial numbers also are required when the processing option is OUTPUT or OUTIN with DISP=MOD.

If the volume serial number is correct, the volume is considered to be mounted and verified. If the serial number is not correct, the system rejects the tape and issues another mount message.

Positioning the volume to the data set

After the volume is mounted and verified, the system positions the tape to the front of the header label group of the data set to be processed. Usually, there is only one data set on the volume, and the header label group immediately follows the volume label. DFSMSrmm provides high speed positioning to other than the first data set.

Unless the data set is cataloged, you specify a data set sequence number in the LABEL parameter of the DD statement to retrieve a data set when more than one data set is on a single volume of tape. You need not specify a data set sequence number for a cataloged data set, because the system obtains the number from the catalog along with the volume serial number.

The sequence number can be from 1 to 65535, 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, your task will be abnormally terminated or, if the volume ends with EOV labels, the open routine will switch to the next volume.

If the data set is not cataloged and you do not specify a sequence number, or you specify 0, the system assumes that the data set is the first in sequence on the volume.

To position the tape, the system 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 UCB. The number in the UCB represents the current position of the tape, and is maintained as follows:

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened, the open routine sets the data set sequence number in the UCB to 1. The exceptions are:
   - If the tape is still positioned from previous processing, such as for a LEAVE request, the open routine does not reset the number in the UCB.
   - If the data set sequence number in the JFCB and the data set sequence number in the first HDR1 label on the tape are both greater than 1, the open routine sets the data set sequence number in the UCB to the value of the number in the first HDR1 label. The data set sequence number in the first HDR1 label may be greater than 1 when the volume is part of a multiple data set/multiple volume aggregate.
   - When the processing method is INPUT, INOUT, OUTPUT, OUTIN or to the start of a data set on a multiple file tape, the open routine starts with the first volume, unless a volume sequence number is specified. If OPEN finds that the volume ends with EOV labels before the desired file sequence number, the open routine switches to the next volume and permanently updates the volume sequence number so that the next open routine to this data set starts with the correct volume.
   - When the processing method is RDBACK and a volume sequence number was not specified, the open routine speeds up the search for the end of the data set by starting with the last volume specified. If the data set is not yet present on the last volume specified, and if the file sequence number is 1, the open routine can recover by backing up volumes. It detects that the data set is not present if the *dsname* is invalid, the tape starts at a file sequence number greater than 1, or the VOL label is followed by a tape mark.
3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the open routine adjusts the data set sequence number in the UCB as the tape is positioned 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.
5. If the data set is not open or has been closed, the data set sequence number in the UCB is set to X'0000' if:

   - The data set was never opened
   - CLOSE (,REWIND) was specified
   - CLOSE (,REREAD) and LABEL=1 was specified
   - CLOSE (,DISP) was specified or defaulted, and DISP=(,PASS) was not specified on the JCL

   Otherwise, the data set sequence number field of the UCB has a value that is one greater than the value specified on the LABEL parameter of the JCL.
6. If the job abends while a tape data set is open, the data set will be closed and the tape is positioned as when CLOSE (,LEAVE) is specified. That is, the data set sequence number in the UCB has a value that is one greater than that specified on the LABEL= parameter of the JCL.

There are several instances in which a volume is repositioned to the next (or previous) tape mark during open. This is usually done by reading data but suppressing data transmission to storage until a tape mark is found, but can be done by I/O spacing commands (for example, BACKSPACE FILE). To reduce the chance of an unexpected record condition (613-0C), the first method is preferred over the spacing commands. In the event of a 613-08 or 613-0C abend, a data management abend installation exit (IFG0199I) is given control to try further recovery. For more information about the data management abend installation exit, see *z/OS DFSMS Installation Exits*.

Only one data set on a tape volume may be open at any given time. Abnormal termination occurs when you attempt to process a second data set on the same volume.

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

Processing the HDR1 label

This topic covers the following subjects:

- “Verifying data set name” on page 88
- “Processing label 1 accessibility code” on page 89
- “Processing expiration date” on page 89
- “Block count” on page 89

Verifying data set name

To ensure that the correct data set is being opened, the system compares the data set name in the HDR1 label to the data set name specified by the user in the DD statement. This comparison is made on only the 17 least significant (rightmost) nonblank 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). DFSMSrmm checks the full 44-character data set name if the previous data set name information is known to DFSMSrmm.

The comparison is made on only the last (rightmost) 17 nonblank characters of the data set name shown in the HDR1 label. It is a good practice, therefore, to limit tape data set names to 17 characters or fewer, when unique names are required, as for password-protected data sets.

For Version 1 HDR1 labels, the generation and version numbers of a generation data set are included in the file identifier and thus are included in the characters compared by the system; however, for Version 3 or Version 4 HDR1 labels, they are not included in the file identifier. For generation data sets with Version 3 or Version 4 labels, the generation and version numbers do not participate in password protection.

For tapes with Version 3 or Version 4 labels, during positioning, the file identifier in each HDR1 label encountered is also checked against duplicate names for the requested data set. The duplicate name check occurs only from the volume position at the time of OPEN until the destination position. For example, if a volume is positioned at the end of data set X at position 5 on the volume as a result of a previous CLOSE LEAVE request, and you request that a data set Y be opened at position 6, no duplicate name check will occur because the volume is already at the desired position. You can force a subsequent duplicate name check from the beginning of the volume by using the CLOSE REWIND option.

If a request is for the first data set on a volume, the volume is always rewound to load point, and no duplicate name search occurs.

You can force a duplicate name check from the first volume of a multivolume aggregate by specifying a volume sequence number of 1 (VOL=(PRIVATE,,1)) along with a cataloged data set request *or* along with serial numbers specified. During the search, the message `IEC140I START OF DATASET NOT ON VOLUME` occurs for each volume mounted on which the requested data set is not found.

Because the suffix of a generation data set is not included in the file identifier of a Version 3 or Version 4 HDR1 label, duplicate name checking prohibits maintaining members of the same generation data group on a volume unless label validation has been suppressed.

If a duplicate name is encountered for a Version 3 label, the label validation exit is entered.

Note: If you inadvertently specify an incorrect file sequence number for a data set, and the data set is encountered before the number specified, the exit is entered with a "duplicate name error" condition. If the exit accepts the error, an abend can result when the target data set does not match the requested data set name.

Processing label 1 accessibility code

For tapes with Version 3 or Version 4 labels, the accessibility field in the HDR1 label is also checked if the volume is not RACF protected. If it contains a valid Version 3 or Version 4 character, the file access exit is entered. For more details about accessibility, see “Protecting data” on page 79.

Processing 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, the system notifies the operator and asks for confirmation of the 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. Automatic handling of expiration date protected tapes is provided by DFSMSrmm functions.

Block count

The block count shown in the HDR1 label is always an ASCII 0. This 0 is recorded in the DCB and increased 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 DCB and decreased during processing for comparison to the 0 block count in the HDR1 label.

The block count is verified at the end of the data set or the end of volume. For more information, see “Verifying block count” on page 10 and “Verifying block count in label when reading” on page 91.

Processing the HDR2 label

This topic covers the following subjects:

- “Data set characteristics” on page 90
- “Positioning after header label 2” on page 90

Data set characteristics

The HDR2 label immediately follows the HDR1 label. The system 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
- Type of control characters
- Coded character set identifier (CCSID)

The above information is obtained from the label and recorded in the job file control block (JFCB) and the data control block, provided that 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. For an explanation and illustration of this merging process, see "Completing the data control block" on page 3.

Because the HDR2 label is optional with ISO/ANSI interchange tapes, and the HDR2 format may vary if produced by systems other than MVS, the manner of processing HDR2 labels depends on whether the HDR1 label specifies "IBMZLA" as the name of the system producing the tape. If "IBMZLA" is specified for Version 3, the HDR2 label "Reserved for Operating System" field contains fields that are similar to an IBM standard HDR2 label, and is processed accordingly. See "Processing the HDR2 label" on page 41 for more information. For Version 4, the "Reserved for Operating System" field also contains the CCSID that the data set was written in. If "IBMZLA" is not specified, the data in the HDR2 label "Reserved for Operating System" and "Record Format" fields cannot be used by the operating system because the contents are unknown. In that case you can specify the missing information in the DCB or DD statement.

The HDR2 label is merged into the vacant fields of the JFCB if a tape data set is being opened for INPUT, INOUT, DISP=MOD, OPEN for OUTPUT, or RDBACK.

Positioning after header label 2

Unless user header labels are to be processed, the system reads forward until a tape mark is found. All labels that follow the HDR2 label are checked for proper sequence until the tape is positioned at the first data set record (unless validation checking has been suppressed). This includes the optional HDR3-HDR9 labels.

Note: If the HDR2 label is missing, processing will continue with the assumption that it is not needed. If the JFCB/DCB merge function needs it because of missing data set characteristics, the OPEN will abnormally terminate.

Processing user header labels

User header labels (UHLx) may follow the HDR2 label. To make the user header labels available to your program, ANSI and user header or trailer labels (AUL) must be coded on the DD statement and the address of an input user header label routine must be specified in the DCB exit list. If you omit AUL or the user header label entry in the DCB exit list, the system positions the tape past the tape mark immediately after processing either the HDR2 label or the HDR1 label if the HDR2 label is not present. User label processing and the DCB exit list are described in *z/OS DFSMS Using Data Sets*.

Read backward

For the RDBACK processing method, the system 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, and so forth. The data records, however, are read in reverse sequence.

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

End-of-Data or end-of-volume on input

The system's EOV routine handles both end-of-data-set and end-of-volume conditions on input. These conditions occur when either a tape mark is read or an FEOV macro instruction is executed by the processing program.

After encountering a tape mark, the system checks the first 4 bytes of the first trailer label for the identifier EOV1 or EOF1. Processing is abnormally terminated if neither identifier is found.

When the FEOV macro is issued, the identifier of the first trailer label is not checked. The trailer labels on the current volume are not processed, but the volume labels and header labels on the next volume are processed.

For an end-of-data condition, the first trailer label is processed, unless deferred user input trailer label processing is specified in the DCB exit list. For an end-of-volume condition, the first trailer label on the current volume is processed, and then the volume label and header labels on the next volume are processed.

Except when it is used as a header label for a RDBACK operation, the system ignores the second trailer label (EOV2 or EOF2) of an input data set. Unless the tape is read backward and the trailer labels are used as header labels, trailer labels are not validated for Version 3 or Version 4 standards.

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

Verifying block count in label when reading

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

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

Register
  Contents

0
  The block count shown in the label

1
  The address of the DCB

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:

ReturnCode
  Meaning

0
  Abnormally terminate with completion code of X'237'

4
  Continue processing.

For more information, see *z/OS DFSMS Using Data Sets* and “Verifying block count” on page 10.

If the appropriate user exit entry is not provided in the DCB exit list, or a return code of 0 is received, then a block count discrepancy causes the system to call the optional DCB ABEND exit and issue ABEND 237.

When the FEOV macro is executed, the block count is not verified. For more information, see *z/OS DFSMS Using Data Sets*.

If the data set was created with an EXCP DCB with no device-dependent section, then the block count is written as zero and is not verified. The data set can be created that way only if the label validation installation exit allows it. This is a requirement of the Version 3 standard.

The EOV2/EOF2 label

The system ignores the EOV2 or EOF2 trailer label of an input data set, except when it is used as a header label for a read backward operation.

User Trailer labels

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

Determining volume switch

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

- For noncataloged data sets, you specify the volume serial numbers in the VOLUME parameter of the DD statement. The system processes the group of volumes in whatever order you specify and processes only the volumes you specify. If you specify volumes out of order or omit volumes, you will get warning messages IEC709I or IEC710I.
- For cataloged data sets, the system processes the group of volumes in sequential order.

However, you can begin processing at any volume of the group by specifying a volume sequence number in the VOLUME parameter of the DD statement.

For input, the label identifier of the trailer labels determines whether the system continues processing the data set. When the system finds an EOV label, it performs volume switching. When the system finds an EOF label, it passes control to the user's end-of-data routine, with these exceptions:

- If the data set is being turned into a concatenation, OPTCB=B generates the concatenation to allow this to occur. If the application is turning on the unlike concatenation bit causing a multivolume data set to be treated as threeseparate data sets, then there is a resrtiction. If you have a VBS data set that spans volumes in such a way that one segment (for example, the first segment) is at the end of the first volume and the next segment (for example, the middle segment) is at the beginning of the next volume, and you attempt to treat these volumes as separate data sets, the integrity of the data cannot be guaranteed. QSAM will abend. QSAM is designed to ensure that all of the segments can be put together, but will be unable to do so. This restriction will also be based on the data and whether or not the segments are split between volumes.
- If the DD statement specifies OPTCD=B, and if an additional volume is available, then the system performs volume switching. If you specify OPTCD=B, no volumes are available for switching, and no data set concatenated after the current one, then the system passes control to the user's end-of-data routine.
- If another data set is concatenated after this one, the system processes it. If the user has not set on the unlike-attributes bit (the DCBOFPPC bit), then the system treats the concatenated data set as if it began another volume of the current data set. If the user defined an EOV exit, the system calls it (not a DCB open exit). If the unlike-attributes bit is on, then the system closes the current data set and opens the concatenated data set. If the user defined a DCB open exit, the system calls it (not an EOV exit). For more information on using concatenated data sets, see *z/OS DFSMS Using Data Sets*.

To determine whether additional volumes are required, the system maintains a volume sequence number in the DEB in storage.

- For read forward operations, the volume sequence number in the DEB increases as each volume is processed. This count is compared to the total number of volumes requested, as shown in the JFCB.
- For RDBACK 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 decreased as each volume is processed, until the count reaches 0.

If another volume is not required (end-of-data-set condition), control is given to the user's end-of-data-set routine that is specified in the DCB or DCBE. Subsequently, the processing program or the operating system closes the data set. The user's end-of-data-set routine is not entered until the last specified volume or the last concatenated data set is processed. If an input data set is closed before the end of the data set is reached, the user's end-of-data-set routine is not entered.

If another volume is required (end-of-volume condition), the system 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 checks to see if there is a next-plus-one volume specified, and if the volume just completed can be rewound and unloaded. If so, the EOV routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid; the next-plus-one volume label is not verified at this time. This lookahead or parallel mounting results in an IEC501E mount message.

The next-plus-one volume mount is referred to as the lookahead mount. Lookahead mount processing is disabled when the volume sequence number that EOV is currently processing equals either the number of volsers in the JFCB or the JCL volume count. To ensure that lookahead mount processing stays enabled up to the maximum allowed volume sequence, specify a JCL volume count equal to this maximum volume sequence. You can add multiples of 15 to the smallest maximum JCL volume count, 5. For example, the JCL volume count should be 5, or 20, or 35, and so forth.)

Note: The IEC501E message has a type 2 descriptor code that causes the message to be retained on the screen.

If you do not want message retention, turn off the message retention attribute. You can stop message retention by either:

- Specifying RETAIN(NO) for the message id in an MPFLST*xx* record
- Using a WTO message exit (an MPF specific exit or IEAVMXIT) to alter the message descriptor code or to request "do not retain this message"

For tape mounts, the IEC501E message remains on the screen until the mount is satisfied or until EOV has occurred on the previous volume. EOV deletes the IEC501E message and issues the IEC501A message.

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, to ensure that the correct volume is mounted.

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

- *Multivolume data set*: The data set sequence number is irrelevant for the second and subsequent volumes of a multivolume data set. The 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. Processing is abnormally terminated if:
  - The data set name is not the same
  - The volume is password protected, no previous volume in the data set was password protected, and the system operator or TSO user is not able to enter a correct password.

Note: For volumes residing in an IBM automated tape library, or manual tape library, password protection processing of the data tape set is bypassed. These tapes can be protected with RACF.

- *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 as it was when the first data set was opened.

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. However, the HDR2 label is validated for Version 3 standards.

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 unless deferred user input trailer label processing was specified in the DCB exit list. If deferred user input trailer label processing was specified, the processing otherwise performed for an input end-of-data condition is performed when an input data set is closed.

If an input data set is closed before it reaches the end of data set or the end of volume, or if the FEOV macro is executed, processing of trailer labels is not performed.

Opening an output data set

When a data set is opened for output, processing is similar to opening for input, except that:

- Only Version 3 or Version 4 tapes are created for output processing.
- If the system-wide RACF tape protection option has been specified, and the DD statement has specified PROTECT=YES but has not previously been opened for output processing, OPEN ensures that the PROTECT=YES specification is valid. Both the volume sequence number and the file sequence number must be set to 1 and a private volume must be requested. The protection indicator in the JFCB is reset so that subsequent OPENs of that DD statement for output processing do not attempt validity checking (of the PROTECT=YES specification) and definition of the volume to RACF.
- The label anomaly exit and volume label editor are entered for label conflicts or for version conflicts (the label is not Version 3 or Version 4) if output is to the first data set. If output is to any data set other than the first data set, or if the first data set is allowed to be extended, a version compatibility conflict causes the volume to be rejected.
- An action message is issued to the operator if the tape is file protected (to allow writing).
- If the system-wide RACF tape protection option has been specified, RACF authorization at the UPDATE level is checked. If a Version 3 or Version 4 tape is not RACF protected, the volume accessibility code is checked as it is for input processing. For additional information, see "Protecting data" on page 79.
- Symmetry violations during output to a Version 3 volume occur if the open option is EXTEND or OUTINX. Open for OUTPUT or OUTIN with DISP=MOD during output also violates symmetry. Open for OUTPUT or OUTIN with DISP=MOD is checked after the volume has been positioned. An EXCP DCB is checked to ensure the presence of a device-dependent area large enough to contain a block count.

If a density conflict occurs, the label anomaly installation exit and possibly the volume label editor are entered. If the conflict occurs for the first data set on the volume, a new volume label (Version 3 or Version 4) is written in the density specified by the user. For other than the first data set, the data set header labels are written in the density of the volume label at the beginning of the tape, unless the volume label editor is modified. If the old volume label is longer than 80 characters, the excess characters are lost unless:

- The IBM-supplied volume label editor is replaced with a program that protects the extra data during a density conflict before returning to open/EOV for reverification of the volume, or

- The operator rejects a rewrite of the label by a response to message IEC704A L or IEC704A L UVL.

For compaction capable tape drives, the system enforces single mode recording. This results in the second and subsequent data sets on the volume being written in the same mode as the first data set. Since the existing mode is determined by information in the volume label, single mode recording is not enforced for NL volumes.

New header labels are written after the volume label is rewritten.

Attention: Doing multiple opens and closes without writing any user data in the end-of-tape reflective marker area can result in the writing of header and trailer labels past the marker. Access methods detect the reflective markers; however, since the creation of empty data sets does not involve access methods, the end-of-tape marker is not detected. This can cause the reel or cartridge to run out of tape.

Volume serial number

You are not required to specify volume serial numbers for output tapes, unless you are extending an existing tape data set. Serial numbers are required when the processing method is EXTEND or OUTINX or it is OUTPUT with DISP=MOD. If no number is specified, the mount message directs the operator to mount a scratch tape. The system obtains the volume serial number from the VOL1 label and records it in the JFCB and the UCB.

If you choose to specify the volume serial number, the system compares it with the volume serial number shown in the VOL1 label. If the serial numbers match, then the system resets a mount switch in the UCB to indicate that volume mounted is verified. The mount switch is initially set when the mount message is issued to the operator. If the serial numbers do not match, the system may give the operator the option of having the label rewritten with the serial number of the volume requested. This will occur if the tape is not password protected, is not date protected, and is not a checkpoint/restart volume. Otherwise, the system rejects the tape and issues another mount message. DFSMSrmm prevents an incorrect tape from being used and the label being rewritten.

Positioning the volume to the data set

After the volume is mounted and verified, the system positions the tape to receive the new data set. Usually, the new data set is the first and only data set on the tape, so the tape remains positioned immediately following the VOL1 label. DFSMSrmm provides support for high speed positioning for data sets other than the first data set. The OPEN routine provides the block identifier, and DFSMSrmm saves it in its control data set and supplies it back to OPEN on any subsequent request to the data set. For 3590 and higher tape technologies, OPEN routines perform High Speed Positioning by default. This positioning does not rely on block identifiers.

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 65535, with 1 representing the first data set on the volume. If the volume ends with EOV labels before the specified sequence number, the open routine will switch to the next volume. If you specify a sequence number that is greater than the number of data sets existing on the volume, plus one, your task will be abnormally terminated. For any label validation errors encountered beyond the 254th data set, the error message does not include the explicit sequence number; instead, it indicates "254+".
- If you do not specify a sequence number, or if you specify 0, the system assumes that the data set is to be written as the first one on the volume.

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

Only one data set on a tape volume can be open at any time. Processing is abnormally terminated if you attempt to open another data set on the same volume.

When the tape is positioned to receive the new data set, the system expects to find either an existing HDR1 label or a tape mark. If neither is present, the system 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 tape mark is found, it indicates that a HDR1 label does not exist at the position at which the new data set is to be written. The system bypasses all further label verification and accepts the tape for output. The conditions under which the system finds a tape mark instead of a HDR1 label are:

- When a tape mark immediately follows the VOL1 label. This can occur when the tape is initialized by means other than the EDGINERS or IEHINITT utility programs (both write a dummy HDR1 label following the VOL1 label). The tape mark is overwritten by the new HDR1 label.
- When, for multiple data set organizations, the new data set is to be written after the last existing data set on the volume. In this case, the system encounters the second tape mark following the existing EOF trailer label group. The tape mark is overwritten by the new HDR1 label.

If the system finds an existing HDR1 label, it checks the label to determine whether the existing data set can be overlaid. Duplicate data set names are checked for Version 3 volumes during positioning, as described under "Opening an input data set" on page 86. If a volume is not RACF protected, the accessibility code in the existing HDR1 label of a Version 3 or Version 4 volume is checked before it is overwritten with a new HDR1 label.

Expiration date on existing label

For a Version 3 volume with multiple data sets, the system checks the expiration date of the data set that immediately precedes the output data set. If the expiration date of the previous data set is later than the expiration date of the output data set, unless label validation has been suppressed, the label validation exit is entered.

If the previous data set's expiration date is equal to or greater than that of the output data set, then the expiration date of the output data set is compared with the current date. If the expiration date of the output data set has not been reached, then the operator confirms the use of the tape or mounts another tape. DFSMSrmm provides facilities that enable automation of handling expiration protection. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

Any data sets following the output data set are treated as if they expired on the same day as the output data set.

Writing data set header labels

When the tape is accepted by the system for output, the system 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 a label is explained in the description of label formats. The process of updating the JFCB is explained in Figure 2 on page 4.

A security indicator of 0, 1, or 3 is set in the HDR1 label even if the volume is RACF defined. However, when the volume resides in an IBM automated tape library, the security indicator is always set to 0, indicating no password protection.

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

Writing user header labels

For the system to write User Header Labels (UHLs), you must code AUL on the LABEL parameter on the DD statement and specify the address of an output user header label routine in the DCB exit list. The DCB exit list is described in *z/OS DFSMS Using Data Sets*.

Unless label validation has been suppressed, the user header labels are validated to ensure they contain valid ASCII characters. Violations cause the label validation exit to be entered.

User header labels do not have to be sequentially numbered or lettered. There is no limit to the number that can be associated with a data set.

Permanent I/O error

In some cases, if a permanent I/O error occurs during label processing, and the data set is the first one on the first or only volume, then the operator removes the tape and mounts a scratch tape, even if you request a specific volume. The job will be abnormally terminated if the data set is not the first one on the volume or is not the first volume of a multivolume data set.

EOV on output

The system's EOV routine automatically switches volumes when one of the following EOV conditions occur:

- The reflective strip on a reel is encountered
- The logical end of a tape cartridge is reached
- The FEOV macro is executed

This volume switching includes:

1. Checking whether the expected number of blocks were read or written (see "Verifying block count" on page 10)
2. Writing trailer labels and a tape mark on the current volume
3. Ensuring that the next volume is mounted
4. Checking existing volume and header labels on the new volume
5. Writing a volume label on the new volume (in the case of a density conflict or version conflict)
6. Writing header labels on the new volume

If a density or a version conflict occurs when the volume label is checked, a new label (Version 3 or Version 4) is written in the density or the version specified by the user. If any VOL2 - VOL9 or user volume labels (UVLs) exist, they are overwritten. If the original volume label is longer than 80 characters, the excess characters are truncated when the new volume label is written.

When multiple tape units are used, the EOV routine checks to see if a next-plus-one volume is needed, and if the volume just written can be rewound and unloaded. If so, the EOV routine issues a message directing the operator to mount the next-plus-one volume on the tape unit just used. This is a premounting aid; the next-plus-one volume label is not verified at this time. This lookahead or parallel mounting results in an IEC501E mount message.

Note: The IEC501E message has a type 2 descriptor code that causes the message to be retained on the screen.

If you do not want message retention, turn off the message retention attribute. You can accomplish this by either:

- Specifying RETAIN(NO) for the message id in an MPFLST*xx* record
- Using a WTO message exit (an MPF specific exit or IEAVMXIT) to alter the message descriptor code or to request "do not retain this message".

For tape mounts, the IEC501E message remains on the screen until the mount is satisfied or until EOV has occurred on the previous volume. EOV deletes the IEC501E message and issues the IEC501A message.

Writing data set Trailer labels

Trailer labels are always written at an EOV condition on output tapes and 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.

Two tape marks are written at EOV following the data set trailer labels. If user trailer labels are written, the tape marks follow the user labels.

Writing user Trailer labels

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

The contents of UTLs are not checked during EOV processing, but, if they contain any invalid ASCII characters, the condition is detected during a subsequent open for RDBACK.

Writing 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, except the version level of the new volume will be the same as the version level of the previous volume. It will not be allowed to create a mixed version multi-volume data set. All volumes in a multi-volume data set must be the same version, either Version 3 or Version 4.

When positioning the second and subsequent volumes, the EOV routine does not use the data set sequence number. The EOV routine assumes that the data set continues at the beginning of the new volume.

RACF processing on the new volume

If the system-wide RACF tape protection option has been specified, RACF authorization checking occurs for the new volume. If the previous volume is RACF protected, the new volume must either be not defined to RACF (in which case, EOV will define it to RACF as part of the previous volume's volume set profile), or the new volume must be defined as part of the same volume set profile as the previous volume. If the previous volume is not RACF protected, then the new volume is rejected if a nonspecific request is made, or the program will be abnormally terminated if a specific request is made.

For tapes with Version 3 or Version 4 labels, the volume access exit is entered if the new volume is not RACF protected and the accessibility field in the VOL1 label contains a valid Version 3 or Version 4 character.

Special EOV conditions

The following conditions signal the end of the volume:

- The drive signals to the system that it is nearing the end of the tape.
  - On a reel, the system detects the end-of-tape reflective marker.
  - On a cartridge, internal mechanisms sense the end of the tape.
- The application program issues an FEOV macro.

The EOV routine writes the trailer labels as described above. If the drive signal occurs while the system is 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.

If the system is writing header labels, and the drive signals that it is nearing the end of the tape, the system:

- Writes the new header label group
- Allows the user to write one record
- Writes the new trailer label group
- Performs volume switching

Closing an output data set

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

Writing data set Trailer labels

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

The close routine writes two tape marks following the trailer labels. If user labels are to be written, the tape marks follow the user trailer labels. If another data set is added to the tape (multiple data set organization), then that data set's HDR1 label overlays the second tape mark. The close routine passes the block identifier on a cartridge tape to the installation's tape end-on-volume exit. DFSMSrmm saves this information in its control data set. See *z/OS DFSMS Installation Exits*.

Writing user Trailer labels

When AUL is coded on the DD statement and the address of an output UTL routine is specified in the DCB exit list, the close routine can write as many UTLs as desired.

The contents of the UTL are not checked during close processing, but if they contain any invalid ASCII characters, the condition is detected during a subsequent open for RDBACK.

Checkpoint/Restart not allowed

Any ISO/ANSI tape that is open during a CHKPT macro service prevents a checkpoint from being taken.

Chapter 4. Nonstandard labels

Nonstandard labels do not conform to the IBM or ISO/ANSI standard label formats. They are labels that you design. Your installation also must provide routines to write and process them. There are no requirements for the length, format, content and number of nonstandard labels, except that the first record on a tape cannot be a standard volume label recorded in EBCDIC, BCDIC or ASCII.

DFSMSrmm does not support nonstandard labeled tapes.

Nonstandard label routines are inserted into the control program by link-editing them into SYS1.LPALIB. Only an authorized system programmer can update SYS1.LPALIB. IBM recommends that you use SMP/E to link-edit.

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

If you use nonstandard tape labels and you want to use the DDR option, your system program must provide an exit routine to perform volume verification. The routine must be able to perform your verification within the first 48 bytes of any record in your nonstandard label. An LPA routine named NSLREPOS must be provided.

For detailed information about these and other available exit routines, see *z/OS DFSMS Installation Exits*.

Nonstandard volume label verification and editor routines

If you specify that an input or output tape has a nonstandard label, the operating system checks for the nonstandard volume label at the beginning of the tape.

Because of conflicting label types or conflicting tape characteristics, various error conditions can occur during the verification of the first record. Under some error conditions, the tape is accepted for use. Under other 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 label anomaly installation exit and possibly to a *volume label editor routine.* Your installation can use IBM-supplied routines, or it can supply its own routines.

The IBM-supplied volume label editor routines determine the discrepancies between the requested tape and the mounted tape and, if necessary, pass control to the appropriate system routine to create or destroy labels. DFSMSrmm uses a label anomaly exit that prevents the creation and destruction of standard labels. Installation-supplied routines can perform other functions.

If your installation supplies its own volume label editor routines, the first (or only) module of each routine must be named as follows:

- OMODVOL1 (for the editor routine associated with open)
- EMODVOL1 (for the editor routine associated with EOV)

If either of your installation's editor routines consists of more than one load module, the 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.

For detailed information about these and other available exit routines, see *z/OS DFSMS Installation Exits*.

Verifying block count

At the end of the data set on each cartridge (not reel), EOV or CLOSE routines can check whether the expected number of blocks were read or written. See "Verifying block count" on page 10.

Chapter 5. Unlabeled tapes

To process or create a tape with no labels, specify NL or LTM (for VSE tapes) in the LABEL parameter of the DD statement. An unlabeled tape contains only data records and tape marks. The organization of data sets on one or more volumes is shown in Figure 17 on page 104. The system automatically writes the tape marks on output and expects to find a similar placement of tape marks on input.

- A tape mark does not precede the first data set on any volume if you specify NL. A tape mark can precede the first data set if you specify LTM. If you specify NL for a tape with a tape mark as the first record, it means the first data set contains no data. If you specify LTM for an unlabeled tape that does not contain a leading tape mark, the system treats the request the same as for NL. See "Unlabeled tapes" on page 111 for more information.
- A tape mark follows each data set.
- Two tape marks follow a data set if it is the last or only data set on the volume.

An unlabeled tape can be read backward even though there is no tape mark preceding the first data set. In this case, the end-of-data-set condition is signaled by the reflective strip or other indicator at the beginning of the tape.

Open/Close/EOV lookahead mounting does not accept any unlabeled volume that is mounted before the system requests it (UCBNRY=0 and UCBVOLI=zeros). A demount message with a blank *vol ser* is issued.

Note: AVR automatically recognizes mounts of labeled volumes. If JCL-specific requests are used and a volume that is not specified is mounted at allocation time, the system requests a demount of the incorrect volume and a mount of the specified volume. Avoid mounting unlabeled tapes before the system requests them.

Bypassing label processing (BLP) is identical to NL processing except that the check for an existing label is bypassed. For more information about BLP, see "Bypassing label processing" on page 10.

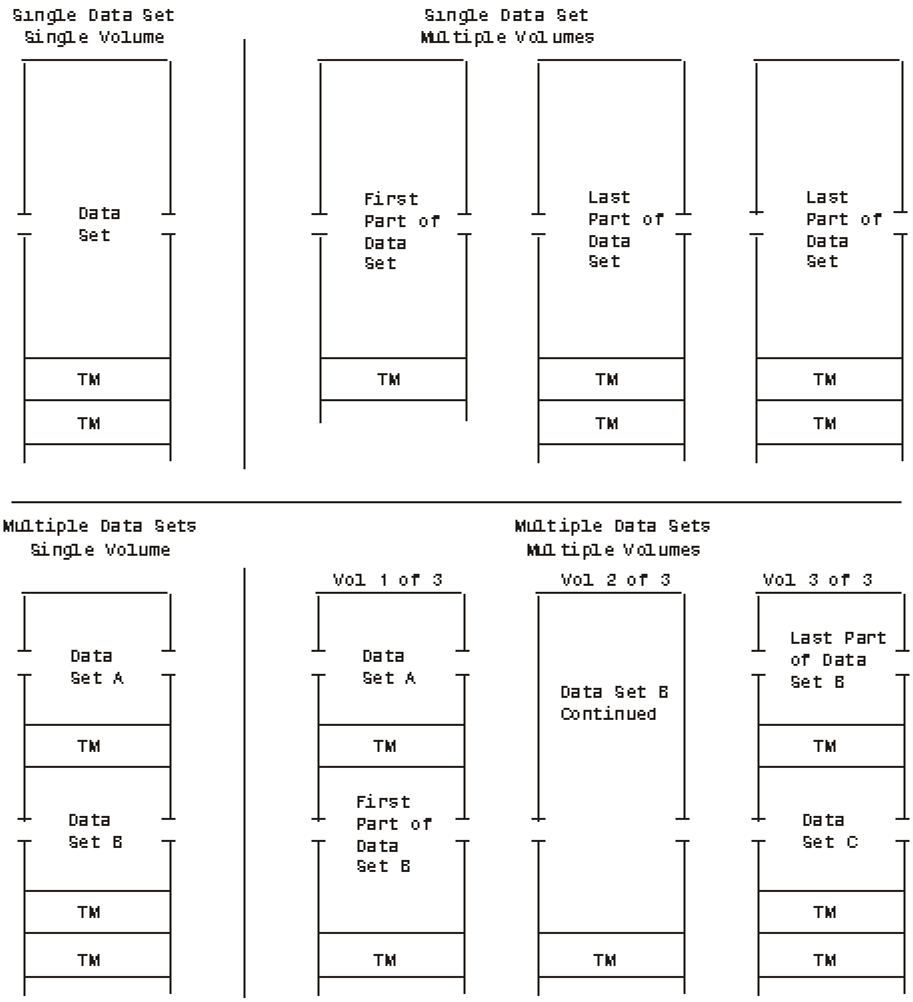


Figure 17. Organization for unlabeled tapes

Creating an unlabeled tape volume with DFSMSrmm

You can use the EDGINERS utility to create an unlabeled tape volume. DFSMSrmm verifies that the correct volume is mounted, either by reading any existing label or by asking the operator to confirm the external label. DFSMSrmm writes an 80-byte record that is not recognized as any valid label format, however, the DFSMSrmm utility can recognize it as an unlabeled tape once it is initiated.

Opening an input data set

When you specify no labels, the system checks the input tape to ensure that the first record is not a standard volume label (that is, the first 4 characters are not VOL1 in EBCDIC or ASCII). If the first record is a standard volume label, the tape is rejected with a message from the system directing the operator to mount the correct tape or a valid unlabeled tape. The various error conditions that can occur during label verification are explained under "IBM standard volume label editor routines" on page 37, "ISO/ANSI volume label verification and editor routines" on page 79, and " Nonstandard volume label verification and editor routines" on page 101.

The search for a standard label is the only mount verification performed by the system. Without labels, neither the volume nor the data set can be positively identified and the system assumes that they are correct. The operator checks the reel's or cartridge's external identification to ensure compliance with the mount message.

Positioning the volume to the data set

When the tape is accepted for input, the system positions the tape at the first record of the data set to be processed. Usually there is only one data set on the volume and positioning is set to the first record on the tape.

To retrieve a data set when there are more than one on a single volume of tape, 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 65535, 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, then OPEN spaces the tape through and removes it from the volume or receives an I/O error.
- If you do not specify a data set sequence number or specify a zero, and the data set is not cataloged, then the system assumes that the data set is first in sequence on the volume.
- The first data set on an unlabeled tape is not normally preceded by a tape mark. If a tape mark precedes the first data set, the sequence number of that data set is two. The effect is as if a data set containing no data preceded the tape mark. Alternately you can use the LTM option described in "Leading tape marks" on page 112.

To position the tape, the system uses the requested data set sequence number in the JFCB and maintains a logical data set sequence number in the UCB. The number in the UCB represents the current position of the tape and is maintained as follows:

1. When a tape is first mounted, the data set sequence number in the UCB is 0.
2. When a data set is opened, the open routine sets the data set sequence number in the UCB to 1. If the tape is still positioned from previous processing, such as for a LEAVE request, the open routine does not reset the number in the UCB. This means that multiple volume, multiple data set NL tape, unlike SL tape, requires a different JCL data set sequence number, unless the tape stays mounted.
3. The data set sequence number in the UCB is compared to the requested data set sequence number in the JFCB. If they are equal, the tape is already positioned at the requested data set. If they are not equal, the open routine adjusts the data set sequence number in the UCB as the tape is positioned past each data set until the number in the UCB equals the number in the JFCB.
4. To position to a data set when there are multiple data sets on multiple volumes, specify the volume serial numbers of the volumes on which the data set resides in the VOLUME parameter of the DD statement and the data set sequence number in the LABEL parameter. You are not required to specify the volume serial number or the data set sequence number for a cataloged data set.
5. 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 can be open at a time. Processing is abnormally terminated if you attempt to begin processing a second data set on the same volume.

Read backward

For the RDBACK operation, the data records are retrieved in reverse sequence. Multivolume data sets can be read backward. Concatenated data sets cannot be read backward. Format-V (variable-length) records cannot be read backward. Seven-track tape with data conversion cannot be read backward.

End-of-Data or end-of-volume on input

For input, the system's EOV routine handles both end-of-data-set and end-of-volume conditions. These conditions occur when either a tape mark is read or an FEOV macro instruction is executed by the processing program. If the tape is a cartridge, the EOV routine can check whether the expected number of blocks were read. See "Verifying block count" on page 10.

Determining volume switch

The volume serial numbers 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. 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. The system processes the group of volumes in whatever order you specify and processes only the volumes you specify.
- For cataloged data sets, the system processes the group of volumes 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. The system does not consider whether the data set on the current volume is followed by one or two tape marks.

If another volume is required, the system obtains the next volume serial number from the JFCB and switches volumes. The system checks the initial record of the new volume to ensure that it is not a standard volume label and positions the tape to the data set. For a multivolume data set, the tape is positioned to the first record on the new volume. If another data set is concatenated after this one, the system processes it. If the user has not set on the unlike-attributes bit (the DCBOFPPC bit), then the system treats the concatenated data set as if it began another volume of the current data set. If the user defined an EOV exit, the system calls it (not a DCB open exit). If the unlike-attributes bit is on, then the system closes the current data set and opens the concatenated data set. If the user defined a DCB open exit, the system calls it (not an EOV exit). For more information on using concatenated data sets, see *z/OS DFSMS Using Data Sets*. For a concatenated data set, the tape is positioned according to the specified data set sequence number.

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

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

Opening an output data set

When you specify unlabeled tape, the system checks the output tape to ensure that the existing first record is not a standard volume label. If the first record is 80 bytes in length and contains the identifier VOL1 in the first 4 bytes, the system checks for the following conditions in the order presented:

1. RACF authorization
2. Password protection
3. Expiration date

If the system-wide RACF tape protection option has been selected, the system checks the alter level authorization to the tape volume. If the tape volume is not defined to RACF, then password protection is checked. If the tape volume is defined and the user is not authorized for ALTER, then the tape is demounted. If the user is authorized for ALTER, then password protection is bypassed.

If the system determines that the volume is password protected, then a message to demount the tape is issued to the operator. Otherwise, the system continues processing by checking the expiration date. If the expiration date is earlier than the current date, then a message is issued to ask to destroy the labels, and the operator can either refuse or allow the use of the tape. If the expiration date is later than the current date, then another message is issued, and again the operator can refuse or allow the use of the tape. If in either situation the operator refuses the use of the tape, then the system requests that another volume be mounted. If the operator accepts the tape, then the system destroys the standard label by writing a tape mark over it, thus providing you with the unlabeled tape you requested.

The installation label anomaly exit, volume label editor exit and DFSMSrmm can prevent standard labelled tapes from being converted into unlabeled tapes.

If the tape volume has been found to be RACF defined and the user has ALTER authorization, then the definition is deleted when the label is destroyed. If the system programmer does not want the system to perform this checking, then the system programmer can replace the label anomaly exit or volume label editor exit.

For more information on the label anomaly exit, label editor routines and the various error conditions that can occur during verification of the first record, see "IBM standard volume label editor routines" on page 37, "ISO/ANSI volume label verification and editor routines" on page 79, and " Nonstandard volume label verification and editor routines" on page 101.

Attention: Performing multiple open and close routines without writing any user data past the end-of-tape marker can result in tape marks being written past the marker. Access methods detect the marker when writing; however, since the creation of empty data sets does not involve access methods, the end-of-tape marker is not detected. This can cause the reel or cartridge to run out of tape.

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.

DFSMSrmm prevents the use of unlabeled scratch tapes because there is no way to tell which volume is mounted or to manage them without operator intervention.

If you request a specific volume, the operating system uses the specified volume serial number for mounting messages, for cataloging, and for passing the volumes to other job steps.

If you do not request a specific volume, the system cannot obtain the actual serial number of the volume that is mounted. In this case, the system generates a volume serial number and assigns it to the volume. These volume serial numbers are generated in the form L*xxxyy*, where:

xxx

is a number the open routine increments (by one) each time an output data set is opened on a nonspecified unlabeled volume. If more than one data set is created on the same volume, this number is increased only when the first data set is opened.

yy

is set to 00 by the open routine. The EOV routine increments this number (by one) each time an end-of-volume condition occurs. In this way, each volume of a multivolume data set is assigned a different volume serial number.

If a data set is to be cataloged, you should specify the volume serial numbers for all the volumes required. This prevents different data sets residing on different volumes from being cataloged with identical volume serial numbers, which could result in the mounting of wrong volumes.

Positioning the volume to the data set

When the tape is accepted for output, it is positioned to receive the new data set. Usually, the new data set is the first or only data set on the volume, so the tape is positioned at load point.

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

- The sequence number can be from 1 to 65535, with 1 representing the first data set on the volume. If you specify a sequence number that is 2 or more greater than the number of data sets existing on the volume, one of three things may happen:

  - the tape will be spaced through and removed from its volume, or
  - the data set will be written but separated from the preceding data set by unusable (old) data, or
  - the cartridge probably will return a positioning error, causing OPEN to issue an ABEND.
- If you do not specify a sequence number, or if you specify 0, the system assumes that the data set is to be written as the first one on the volume.

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

No more than one data set on a tape volume can be open at a time. Processing is abnormally terminated if you attempt to open a second data set on the same volume.

End-of-Volume on output

The system's EOV routine automatically switches volumes when one of the following EOV conditions occur:

- The reflective strip on a reel is encountered
- The logical end of a tape cartridge is reached
- The FEOV macro is executed

The EOV routine:

- Checks whether the expected number of blocks were written (see "Verifying block count" on page 10)
- Writes one tape mark after the data set on the current volume
- Checks the new volume to ensure that the volume does not contain a standard volume label

The output is then continued on the new volume

Closing an output data set

The close routine handles end-of-data-set processing on output tapes. When a write operation is the last operation that occurs before closing a data set (for OUTPUT, OUTIN, OUTINX, EXTEND or INOUT) or when no output is written before closing (for OUTPUT, OUTIN, OUTINX or EXTEND), the close routine writes two tape marks. When you are writing on a cartridge (not a reel), the system can perform block count checking. See "Verifying block count" on page 10.

Restarting from a checkpoint

When a job step is restarted from a checkpoint, the restart routine mounts and positions any tape volumes containing data sets that were open when the checkpoint was taken. Specifically, the restart routine performs the following functions:

- Restores applicable control blocks to the conditions that existed when the checkpoint was taken.
- Ensures that the first existing record on the tape is not a standard volume label (VOL1 in EBCDIC or ASCII).
- If the volume is a reel:
  - Uses the data set sequence number shown in the JFCB to position the tape to the required data set. The method of positioning is the same as previously explained for opening an input data set.
  - Uses the block count shown in the DCB to reposition the tape to the proper record within the data set. For forward read operations, this positioning is performed in a forward direction. If the block count is 0 or negative, the tape remains positioned at the interrecord gap preceding the first record. For backward read operations, this positioning is performed in a backward direction. If the block count is 0 or a positive number, the tape is positioned at the interrecord gap following the last record of the data set.

- If the volume is a cartridge, then the block identifier saved in the checkpoint is used to position the tape to the proper record.

Chapter 6. Using tape volumes created by other systems

Occasionally, is necessary to process a tape volume created by another system. There is no exact procedure: many of the factors vary according to the situation and the user's options at the time the volume was created. The volume can be slightly or extremely different in its organization, label formats, or label contents. Analysis in this chapter 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, use the operating system to create new labels and volume organizations.

Standard labels

All IBM programming systems create tape labels with the same standard label formats. However, the actual contents of each label field may vary from system to system. To see which fields of each label are functional for the operating system, refer to Figure 5 on page 20, Figure 6 on page 22, Figure 7 on page 27, Figure 12 on page 61, Figure 13 on page 64, and Figure 14 on page 69. 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 data set identifier field of data set label 1 (HDR1, EOV1, EOF1). The data set name in the label created by another system may contain embedded blanks or special characters. This name is compared to the data set name that you specify in the DD statement; therefore, you must enclose the name in apostrophes on the DD statement that requests this data set. *z/OS MVS JCL User's Guide* 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 JFCB after the DD statement information is used to update the JFCB.

The operating system can obtain certain data set characteristics from the standard data set label 2 (HDR2/EOV2/EOF2). Some IBM 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 when 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 or the DD statement.

If the first block on the tape is 80 bytes long and begins with VOL1 in EBCDIC or if it is at least 80 bytes long and begins with VOL1 in ASCII, then the tape must meet the requirements described in Chapter 2, “IBM standard labels,” on page 15 or Chapter 3, “ISO/ANSI labels,” on page 53.

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 under Chapter 4, “Nonstandard labels,” on page 101.

If you want to ignore the nonstandard labels, you can retrieve the data set by treating the volume as an unlabeled tape. 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 if the data sets are followed by recognizable tape marks.

Leading tape marks

To position a tape to the desired data set, you must specify the correct data set sequence number in the DD statement. If a tape mark precedes the first data set and the LABEL subparameter LTM is specified, then the system tests for and bypasses, if present, a leading tape mark. If a tape mark should precede the first data set and you do not specify LTM in the LABEL parameter field, you must add 1 to the data set sequence number.

If a multivolume data set from another system has a leading tape mark on one or more of the volumes, the operating system can process it as an unlabeled multivolume data set if the LABEL subparameter LTM is specified. Otherwise, the operating system cannot process it as an unlabeled multivolume data set.

The presence of a leading tape mark, that is, a tape mark that precedes the first data set on the tape, makes each data set the second in sequence on the tape. However, the operating system always assumes that continued data sets are first in sequence on the tape. By specifying LTM in the LABEL parameter field, the first data set on a tape can be accessed whether or not it is preceded by a leading tape mark.

The specification of LTM in the LABEL parameter field does not make allowances for any other excess tape marks. You must make those adjustments in the data set sequence number.

Tapes created on os/400

You can use the standard z/OS access methods to read tapes that were created on the IBM OS/400® operating system.

One incompatibility to consider is variable-length blocks that are shorter than 18 bytes. In the unlikely event that the maximum block size for the data set is less than 18 bytes, z/OS writes the block normally although these very short blocks are very inefficient. These very short blocks require a great deal of overhead processing for a small amount of data.

If the maximum block size for the data set is more than 18 bytes but a block is shorter than 18 bytes, then the access method pads those blocks to 18 bytes. This means the system adds bytes at the end of the block but does not increase the block length value in the block descriptor word (BDW). If the record format (RECFM) is V, then the access method adds bytes of X'00' to the end of the block. If the record format is D, then the access method adds bytes of X'5B' to the end of the block. X'5B' is called the ASCII circumflex.

On OS/400, the system pads a short format-V block with a byte of X'80' and then bytes of X'00'. The z/OS access method treats this as an invalid record and issues system ABEND 002.

When the z/OS access method reads variable spanned records, the access method ignores the rest of the block when it finds a segment that begins with X'80'. Therefore when you read non-spanned format-V records from OS/400, when the block might be shorter than 18 bytes, you might be able to take advantage of this. You can code RECFM=VS or RECFM=VBS on the DD statement even if the records are not spanned. This will cause the access method to ignore the rest of this block when a record begins with X'80'.

Appendix A. Using the IBM Tape Library

Mountable (or virtual) tape cartridges are supported in a tape library under System Managed Storage (SMS). DFSMS provides management facilities for the tape volumes and DFSMSrmm manages data sets on these volumes. For additional information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Tape library

A tape library is a named collection of tape volumes and tape drives. The mounting and demounting of tape cartridges in an automated tape library is performed by robotics in automatic mode and by operators in manual mode. The Tape Configuration Data Base (TCDB), an ICF catalog, contains an inventory of the tape libraries and volumes in the libraries.

Volume labels

All tape cartridges are labeled with either an IBM or ISO/ANSI standard volume label. The internal and external labels on a tape cartridge must be unique and equal to one another, and the external label must be machine readable by the vision system in the library. The internal labels on volumes in libraries are automatically labelled by the operating system during the first use. The only valid characters allowed in a volume label are uppercase A - Z and 0 - 9.

See “System-Managed mountable tape external labels” on page 19, “System-Managed mountable tape external labels” on page 59, and “Cartridge label” on page 117 for more information on external labels.

Tape processing

The system does not issue mount (IEC501A and IEC501E) and demount (IEC502E) messages to a system operator for drives in an automated tape library. These messages are recorded to the system log.

Tape label verification-automated tape library

Specific mount request

The external label is used to verify that the correct volume is mounted. The internal, external, and requested (JCL) volume labels must all match, otherwise the request fails. If there is a label mismatch while the automated tape library is in manual mode, then the volume is demounted and a mount request is reissued. All tape library error retry actions are repeated until the action is successful or a system retry default is exceeded.

Non-specific mount request

A volume's internal and external labels must be identical when the mount is for a non-specific volume. If the labels are not equal or unavailable (no label, uninitialized, etc.), then different recovery actions are taken depending on the mode of operation. If the tape library is in automatic mode, then the system will:

- Create an internal label if the tape is uninitialized or unlabeled.
- Reject the tape if the internal label is unreadable and RACF Tape Volume Protection is active.

If the tape library is in automatic mode, but the external label is unreadable, then the system will:

- Create an internal label if the volume is uninitialized.
- Reject the volume if the internal and external labels are unequal, unlabeled, or unreadable, and RACF Tape Volume Protection is active.

If the tape library is in manual mode, then it will:

- Create an internal label if the volume is uninitialized or unlabeled.
- Issue WTOR IEC704A if the internal and external labels are unequal. The operator can supply the external label or reject the volume.
- Reject the volume if the internal label is unreadable and RACF Tape Volume Protection is active.

If the volume label is extracted from the VOLID MARK and an installation option allows its use in either automatic or manual mode, then the external label is used to verify the volume. The system uses the external label to verify the volume if all of the following are true:

- A higher formatted cartridge is mounted on a drive in response to an output request and that drive does not fully support that recording format.
- OPEN or EOV is able to read the VOLID mark.
- The OPEN or EOV volume label editor installation exit routine allows this relabelling.
- The user has RACF UPDATE authority to the volume.
- The DEVSUP member of SYS1.PARMLIB specifies VOLNSNS=YES.

The DEVSUP*xx* member is described in *z/OS MVS Initialization and Tuning Reference*.

Tape label creation

The issuing of WTOR IEC704A to permit operators to supply label information is the same as for drives outside a tape library. However, the external volume serial number is displayed in the WTOR. The operator can specify only the 'ownerinfo' and the ISO/ANSI access code, not the volume serial number. This is to keep the internal and external volume serial numbers equal.

See “System-Managed mountable tape external labels” on page 19, and “System-Managed mountable tape external labels” on page 59, for more information on external labels.

Tape library scratch tape pools

Scratch tapes in an automated tape library are kept in a common pool based on cartridge type. The system may select any volume from this group to satisfy a nonspecific tape mount. If the tape is rejected during volume verification, then the volume is placed in an error category. These volumes become ineligible for subsequent scratch mount requests until the error is corrected.

Appendix B. Component support of label processing

Job control statements make the label processing facilities of data management available to users of the operating system's assembler, linkage editor, Sort/Merge program product, utility programs, and high-level language program products. Table 6 on page 115 shows the component support for each type of label processing.

Table 6. Component support of label processing types

					COBOL			**IBM Compilers**
Type	**Assembler**	**Linkage Editor**	**DFSORT**	**Utilities**	**AD/CYCLE COBOL/370 COBOL for MVS™ and VM Enterprise COBOL for z/OS and OS/390®**	**VS COBOL II**	**OS/VS COBOL ANS COBOL**	
Uses Data Management Facilities for Label processing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard Labels (SL, AL)	Yes	Yes	Yes	SL-Yes AL-IEHINITT AL-IEBGENER	Yes	Yes	Yes	Yes
Supports Standard User Labels (SUL, AUL)	No	No	Yes	Yes	SUL-Yes AUL-No	SUL-Yes AUL-No	SUL-Yes AUL-Yes	No
Supports Nonstandard Labels (NSL) “1” on page 115	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Supports Unlabeled Tapes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Bypass Label Processing Option “2” on page 115	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Concatenated Data Sets with Unlike Attributes	No	Yes	Yes	No	No	No	No	No

Notes:

1. NSL can be specified only when installation—written routines that write and process the nonstandard labels have been incorporated into the operating system.
2. If the BLP option is not allowed by the system, its use defaults to NL.

Appendix C. External labels

External labels are affixed to the tape reels and cartridges to provide visual identification of the volume and its contents. Normal tape volume control may require several types of external tape labels. One is a *permanent label* that identifies the reel or cartridge; the other is a *temporary label* that identifies the contents. A third type of label, the *checkpoint security label*, may be applied to tape reels or cartridges that contain checkpoint data sets.

To write on external labels, you should use a pen or a felt-tip marker that 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 18 on page 117.

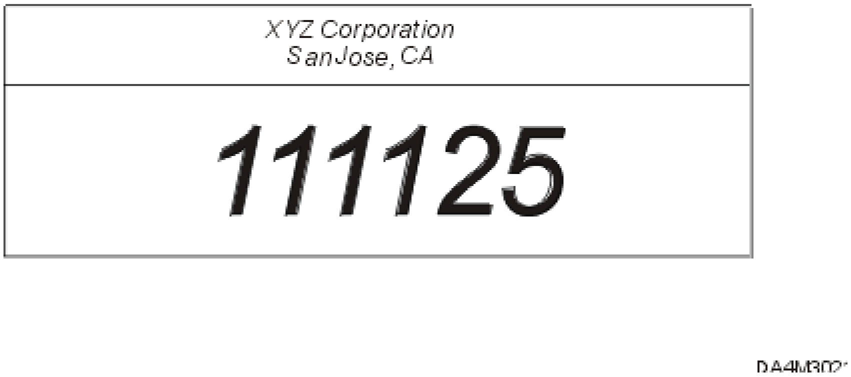


Figure 18. External label for reel identification

Cartridge label

The cartridge 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 cartridge by a unique number and to file the tapes in the tape rack. An example of a cartridge label is shown in Figure 19 on page 118.

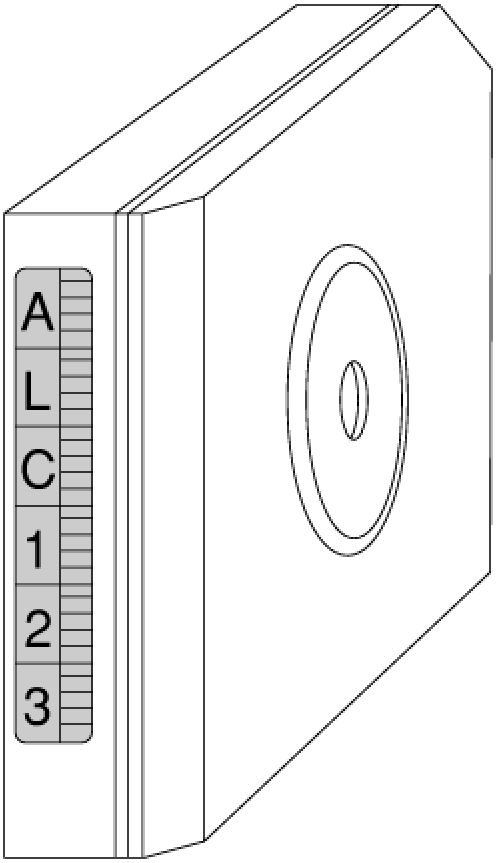


Figure 19. External label for cartridge identification

The external label must be applied within the indentation on the side of the cartridge; however, a larger information label, supplied by the customer, can be used on the top of the cartridge. An example of a cartridge label is shown in Figure 19 on page 118. See "System-Managed mountable tape external labels" on page 19, "System-Managed mountable tape external labels" on page 59, "Volume labels" on page 113, and "Tape label verification-automated tape library" on page 113 for more information on the cartridge external label.

Contents label

The contents label is used to identify the current contents of a particular volume. Because 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.

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 from others. 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 20 on page 119.

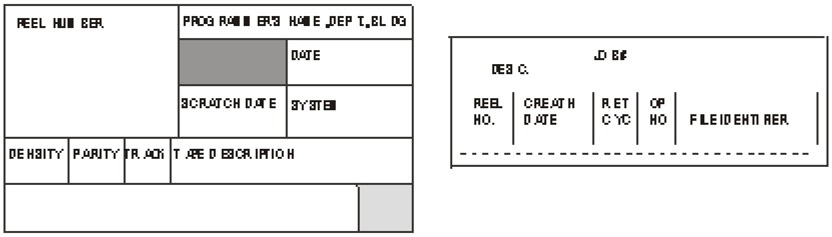


Figure 20. External labels for contents identification

Checkpoint security label

A checkpoint security label may be put on tape volumes containing checkpoint data sets. This label, applied by the operator, helps ensure security of the checkpoint data set.

The checkpoint security label is a temporary label; it should be applied with adhesive strong enough to hold the label securely and yet allow easy removal of the label later when the checkpoint data set is deleted. The size and placement of the label should not interfere with the handling of the tape.

Appendix D. Equivalent ASCII and EBCDIC codes

ASCII

The routine included in the system converts ASCII 7-bit code to and from EBCDIC.

All EBCDIC codes that convert to an ASCII code that cannot be contained in seven bits are represented by the substitute character X'1A'.

The following tables show the relationship of EBCDIC code to ASCII 8-bit code.

Note: This conversion is not necessarily reversible. The ASCII characters that cannot be converted are changed to EBCDIC substitute characters (X'3F'). The EBCDIC characters that cannot be converted are changed to ASCII substitute characters (X'1A').

EBCDIC to ASCII

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
00	00	NUL
01	01	SOH
02	02	STX
03	03	ETX
04	1A	SEL
05	09	HT
06	1A	RNL
07	7F	DEL
08	1A	GE
09	1A	SPS
0A	1A	RPT
0B	0B	VT
0C	0C	FF
0D	0D	CR
0E	0E	SO
0F	0F	SI
10	10	DLE
11	11	DC1
12	12	DC2
13	13	DC3
14	1A	RES/ENP
15	1A	NL
16	08	BS
17	1A	POC
18	18	CAN

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
19	19	EM
1A	1A	UBS
1B	1A	CU1
1C	1C	IFS
1D	1D	IGS
1E	1E	IRS
1F	1F	ITB/IUS
20	1A	DS
21	1A	SOS
22	1A	FS
23	1A	WUS
24	1A	BYP/INP
25	0A	LF
26	17	ETB
27	1B	ESC
28	1A	SA
29	1A	SFE
2A	1A	SM/SW
2B	1A	CSP
2C	1A	MFA
2D	05	ENQ
2E	06	ACK
2F	07	BEL
30	1A	
31	1A	
32	16	SYN
33	1A	IR
34	1A	PP
35	1A	TRN
36	1A	NBS
37	04	EOT
38	1A	SBS
39	1A	IT
3A	1A	RFF
3B	1A	CU3
3C	14	DC4
3D	15	NAK

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
3E	1A	
3F	1A	SUB
40	20	(space)
41	1A	RSP
42	1A	
43	1A	
44	1A	
45	1A	
46	1A	
47	1A	
48	1A	
49	1A	
4A	5B	¢
4B	2E	.
4C	3C	<
4D	28	(
4E	2B	+
4F	21	|
50	26	&
51	1A	
52	1A	
53	1A	
54	1A	
55	1A	
56	1A	
57	1A	
58	1A	
59	1A	
5A	5D	!
5B	24	$
5C	2A	*
5D	29	)
5E	3B	;
5F	5E	¬
60	2D	–
61	2F	/
62	1A	

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
63	1A	
64	1A	
65	1A	
66	1A	
67	1A	
68	1A	
69	1A	
6A	7C	¦
6B	2C	,
6C	25	%
6D	5F	–
6E	3E	>
6F	3F	?
70	1A	
71	1A	
72	1A	
73	1A	
74	1A	
75	1A	
76	1A	
77	1A	
78	1A	
79	60	
7A	3A	:
7B	23	#
7C	40	@
7D	27	‘
7E	3D	=
7F	22	“
80	1A	
81	61	a
82	62	b
83	63	c
84	64	d
85	65	e
86	66	f
87	67	g

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
88	68	h
89	69	i
8A	1A	
8B	1A	
8C	1A	
8D	1A	
8E	1A	
8F	1A	
90	1A	
91	6A	j
92	6B	k
93	6C	l
94	6D	m
95	6E	n
96	6F	o
97	70	p
98	71	q
99	72	r
9A	1A	
9B	1A	
9C	1A	
9D	1A	
9E	1A	
9F	1A	
A0	1A	
A1	7E	°
A2	73	s
A3	74	t
A4	75	u
A5	76	v
A6	77	w
A7	78	x
A8	79	y
A9	7A	z
AA	1A	
AB	1A	
AC	1A	

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
AD	1A	
AE	1A	
AF	1A	
B0	1A	
B1	1A	
B2	1A	
B3	1A	
B4	1A	
B5	1A	
B6	1A	
B7	1A	
B8	1A	
B9	1A	
BA	1A	
BB	1A	
BC	1A	
BD	1A	
BE	1A	
BF	1A	
C0	7B	
C1	41	A
C2	42	B
C3	43	C
C4	44	D
C5	45	E
C6	46	F
C7	47	G
C8	48	H
C9	49	I
CA	1A	
CB	1A	
CC	1A	
CD	1A	
CE	1A	
CF	1A	
D0	7D	
D1	4A	J

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
D2	4B	K
D3	4C	L
D4	4D	M
D5	4E	N
D6	4F	O
D7	50	P
D8	51	Q
D9	52	R
DA	1A	
DB	1A	
DC	1A	
DD	1A	
DE	1A	
DF	1A	
E0	5C	\
E1	1A	
E2	53	S
E3	54	T
E4	55	U
E5	56	V
E6	57	W
E7	58	X
E8	59	Y
E9	5A	Z
EA	1A	
EB	1A	
EC	1A	
ED	1A	
EE	1A	
EF	1A	
F0	30	0
F1	31	1
F2	32	2
F3	33	3
F4	34	4
F5	35	5
F6	36	6

EBCDIC (hex)	ASCII (hex)	EBCDIC Meaning
F7	37	7
F8	38	8
F9	39	9
FA	1A	
FB	1A	
FC	1A	
FD	1A	
FE	1A	
FF	1A	

ASCII to EBCDIC

ASCII (hex)	EBCDIC (hex)	ASCII Meaning
00	00	NUL
01	01	SOH
02	02	STX
03	03	ETX
04	1A	SEL
05	09	HT
06	1A	RNL
07	7F	DEL
08	1A	GE
09	1A	SPS
0A	1A	RPT
0B	0B	VT
0C	0C	FF
0D	0D	CR
0E	0E	SO
0F	0F	SI
10	10	DLE
11	11	DC1
12	12	DC2
13	13	DC3
14	3C	DC4
15	3D	NAK
16	32	SYN
17	26	ETB
18	18	CAN

ASCII (hex)	EBCDIC (hex)	ASCII Meaning
19	19	EM
1A	3F	SUB
1B	27	ESC
1C	1C	FS
1D	1D	GS
1E	1E	RS
1F	1F	US
20	40	(space)
21	4F	!
22	7F	“
23	7B	#
24	5B	$
25	6C	%
26	50	&
27	7D	‘
28	4D	(
29	5D	)
2A	5C	*
2B	4E	+
2C	6B	,
2D	60	–
2E	4B	.
2F	61	/
30	F0	0
31	F1	1
32	F2	2
33	F3	3
34	F4	4
35	F5	5
36	F6	6
37	F7	7
38	F8	8
39	F9	9
3A	7A	:
3B	5E	;
3C	4C	<
3D	7E	=

ASCII (hex)	EBCDIC (hex)	ASCII Meaning
3E	6E	>
3F	6F	?
40	7C	@
41	C1	A
42	C2	B
43	C3	C
44	C4	D
45	C5	E
46	C6	F
47	C7	G
48	C8	H
49	C9	I
4A	D1	J
4B	D2	K
4C	D3	L
4D	D4	M
4E	D5	N
4F	D6	O
50	D7	P
51	D8	Q
52	D9	R
53	E2	S
54	E3	T
55	E4	U
56	E5	V
57	E6	W
58	E7	X
59	E8	Y
5A	E9	Z
5B	4A	[
5C	E0	\
5D	5A	]
5E	5F	^
5F	6D	_
60	79	′
61	81	a
62	82	b

ASCII (hex)	EBCDIC (hex)	ASCII Meaning
63	83	c
64	84	d
65	85	e
66	86	f
67	87	g
68	88	h
69	89	i
6A	91	j
6B	92	k
6C	93	l
6D	94	m
6E	95	n
6F	96	o
70	97	p
71	98	q
72	99	r
73	A2	s
74	A3	t
75	A4	u
76	A5	v
77	A6	w
78	A7	x
79	A8	y
7A	A9	z
7B	C0	{
7C	6A	¦
7D	D0	}
7E	A1	~
7F	07	DEL
80-FF	3F	

Conversion irregularities

The following irregularities exist in the graphic character set between EBCDIC and ASCII 7-bit and 8-bit codes:

EBCDIC Code		8-Bit Code		7-Bit Code	
Graphic	Hex	Graphic	Hex	Graphic	Hex
¢	X'4A'	[	X'5B'	[	X'5B'

EBCDIC Code		8-Bit Code		7-Bit Code	
Graphic	Hex	Graphic	Hex	Graphic	Hex
!	X'5A'	]	X'5D'	]	X'5D'
[	X'AD'	(n/a)	X'D5'	Sub	X'1A'
]	X'BD'	(n/a)	X'E5'	Sub	X'1A'
|	X'4F'	!	X'21'	!	X'21'
¦	X'6A'	¦	X'7C'	¦	X'7C'

For example, an exclamation mark is coded in EBCDIC as X'5A', but in ASCII 7-bit and 8-bit codes as X'21'.

Appendix E. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Documentation with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines `3.1 USERID` and `3.1 SYSTEMID`, your syntax can include either `USERID` or `SYSTEMID`, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The `*` symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element `*FILE` with dotted decimal number 3 is given the format `3 \* FILE`. Format `3* FILE` indicates that syntax element `FILE` repeats. Format `3* \* FILE` indicates that syntax element `* FILE` repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines `5.1*, 5.1 LASTRUN, and 5.1 DELETE` mean that if you use more than one of the `LASTRUN` and `DELETE` syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the `%` symbol, it indicates a reference that is defined elsewhere. The string that follows the `%` symbol is the name of a syntax fragment rather than a literal. For example, the line `2.1 %OP1` means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element

The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the `?` symbol is displayed on the same line as the syntax element, (for example `5? NOTIFY`). If there is more than one syntax element with a dotted decimal number, the `?` symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines `5 ?, 5 NOTIFY`, and `5 UPDATE`, you know that the syntax elements `NOTIFY` and `UPDATE` are optional. That is, you can choose one or none of them. The `?` symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element

The exclamation mark (`!`) symbol indicates a default syntax element. A dotted decimal number followed by the `!` symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines `2? FILE, 2.1! (KEEP), and 2.1 (DELETE)`, you know that `(KEEP)` is the default option for the `FILE` keyword. In the example, if you include the `FILE` keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the `FILE` keyword is omitted, the default `FILE(KEEP)` is used. However, if you hear the lines `2? FILE, 2.1, 2.1.1! (KEEP)`, and `2.1.1 (DELETE)`, the default option KEEP applies only to the next higher dotted decimal number, `2.1` (which does not have an associated keyword), and does not apply to `2? FILE`. Nothing is used if the keyword `FILE` is omitted.

*** indicates an optional syntax element that is repeatable**

The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line `5.1* data area`, you know that you can include one data area, more than one data area, or no data area. If you hear the lines `3* , 3 HOST, 3 STATE`, you know that you can include `HOST, STATE`, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write `HOST STATE`, but you cannot write `HOST HOST`.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line `6.1+ data area`, you must include at least one data area. If you hear the lines `2+, 2 HOST, and 2 STATE`, you know that you must include `HOST, STATE,` or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user’s name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at ibm.com®/privacy and IBM’s Online Privacy Statement at ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/OS DSMS.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks might also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual.

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

See:

This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

ACF
Automated Cartridge Facility

aggregate
A collection of tape volumes that contains multiple data sets

AL
American National Standard Labels

allocation
Generically, the entire process of obtaining a volume and unit of external storage, and setting aside space on that storage for a data set.

ASCII
American National Standard Code for Information Interchange

ATLDS
Automated Tape Library Dataserver

AUL
ANSI and user header or trailer labels

automatic class selection (ACS)
A mechanism for assigning SMS classes and storage groups to data sets.

automatic volume recognition (AVR)
A mechanism for assigning SMS classes and storage groups to data sets.

backup data set
A copy that can be used to replace or reconstruct a damaged data set.

BCDIC
Binary coded decimal interchange code

BDW
Block descriptor word

BLP
Bypass label processing

BPI
Bytes per inch

control volume (CVOL)
A type of catalog commonly used on very old levels of MVS.

CVT
Communication vector table

data class
A list of the data set allocation parameters and their values, used when allocating a new SMS-managed data set in an MVS operating environment.

data set
In DFSMS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In z/OS UNIX environments, the terms *data set* and *file* have quite distinct meanings.

Data Facility Sort (DFSORT)
An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

DCB
Data control block

DCBE
Data control block extension

DD
Data definition

DDR
Dynamic device reconfiguration

DECB
Data event control block

DSCB
Data set control block

DFSORT
See *Data Facility Sort*

EBCDIC
Extended binary coded decimal interchange code

entry
A collection of information about a cataloged object in a master or user catalog. Each entry resides in one or more 512-byte records.

entry name
A unique name for each component or object as it is identified in a catalog. The entry name is the same as the dsname in a DD statement that describes the object.

EOF
End-of-file

EOF1
Standard data set label 1 trailer label

EOF2
Standard data set label 2 trailer label

EOV
End-of-volume

EOV1
Standard data set label 1 trailer label

EOV2
Standard data set label 2 trailer label

FEOV
Force end of volume

FIPS
Federal Information Processing Standard

file
A collection of information treated as a unit. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

GCR
Group coded recording

generation data group (GDG)
A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is called a generation data set

GDG
See *generation data group*

GDS
Generation data set

HDR1
Standard data set label 1 header label

HDR2
Standard data set label 2 header label

header label
Data set labels that precede a data set on tape.

ICF
Integrated Catalog Facility

IDRC
Improved Data Recording Capability

IOB
Input/output block

IRG
Interrecord gap

JFCB
Job file control block

leading tape mark bypass (LTM)
Code signaling that if a leading tape mark is encountered on an unlabeled tape, it is bypassed.

LTM
Leading tape mark

management class
A list of the migration, backup, and retention parameters and their values, for an SMS-managed data set.

MPF
Message processing facility

NL
No label

NSL
Nonstandard label

O/C/EOV
Open/close/end-of-volume

RDBACK
Read backward

reflective strip
A physical signal that marks the logical end of 7-track and 9-track reels of tape. Newer IBM magnetic tape subsystems, like the IBM 3480, use tape cartridges instead of tape reels. These subsystems have internal mechanisms that sense the beginning and end of tape, and do not use reflective strips.

simple name
The rightmost component of a qualified name. For example, APPLE is the simple name in TREE.FRUIT.APPLE. The simple name corresponds to the lowest index level in the catalog for the data set name.

SKP
Skip sequential processing

SL
IBM standard label

SML
Storage Management Library

SMS
See *Storage Management Subsystem*. Also used as an abbreviation for system-managed storage.

SMS class
A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or backup and retention (management class) needs.

SMS-managed data set
A data set that has been assigned a storage class.

SSL
Storage Subsystem Library

storage administrator
A person in the data processing installation who is responsible for defining, implementing, and maintaining storage management policies.

storage class
A list of DASD storage performance, security, and availability service level requirements for an SMS-managed data set.

Storage Management Subsystem (SMS)
An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

SUL
IBM standard and user header or trailer label

Tape mark (TM)
A mark-logical block that can be recorded on the medium under program direction. The program uses tape marks to delineate collections of data-logical blocks on a given volume.

task control block (TCB)
Holds control information related to a task.

TCDB
Tape configuration data base

TIOT
Task I/O table

TM
See *tape mark*

translator
A tape device feature that takes 8-bit EBCDIC characters from the buffer and writes them as 6-bit BCD characters. It translates the other way for read operations. For 7-track tapes.

TTR
Track record address

UHL
User header label

unit control block (UCB)
A control block in storage that describes the characteristics of a particular I/O device on the operating system.

UPD
Update mode

user labels
Data set label groups that can include standard user labels on tape.

UTL
User trailer label

UVL
User volume label

volume
The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, and *tape volume*.

VOL1
Standard volume label

VOLSER
Volume serial number

VTOC
Volume table of contents

Index

Product Number: 5650-ZOS

SC23-6858-50