ADDENDUM 1

VORTEX II Reference Manual

98A 9952 246

This addendum contains new sections 23 (Multitask Spooler) and 24 (Tape Labeling) to be added to the VORTEX II Operating System Reference Manual.

ISSUED: NOVEMBER 1977

TABLE OF CONTENTS

SECTION 23
MULTITASK SPOOLER

23.1 INTRODUCTION

The VORTEX II Multitask Spool system permits background and foreground tasks to produce printed output concurrently with the output directed to RMD files (spool files) rather than immediately transferred to a line printer.

A spool file, upon completion (closed), is available to be printed. Printing of spool files is controlled by the computer operator and is based on spool file status information maintained by the Multitask Spool system in a spool file directory.

The spool file directory is an RMD file named S00000 that resides on the RMD partition to which the non-reassignable logical unit number 107 is assigned.

Multitask Spool system facilities are available to obtain the status of spool files, create (open) and close spool files, and print and delete spool files. A description of these facilities, spool file allocation conventions, and VORTEX II system generation requirements for the Multitask Spool system are presented in this user guide.

23.2 SPOOL FILE·OPEN AND CLOSE CALLS

The subroutine calls SPOPN, spool file open, and SPCLS, spool file close, may be called from user tasks to control spool file print class association, space allocation, and to have open and direct output to multiple spool files concurrently from a single task.

Use of the subroutine calls SPOPN and SPCLS is not required to direct output to a spool file. The first record from a user task directed to a logical unit number assigned to the spool dummy driver, VZSDA, results in a spool file being created with the default values specified in the SPOPN call. Spool files that are not closed by use of the SPCLS call are closed when the task that directed output to the spool file exits from the VORTEX II system.

23.2.2 SPCLS (Spool File Close)

The format of the SPCLS call is:

```
          CALL SPCLS,pc,ln,rc
```

where

pc   is defined as in the SPOPN call.

ln   is defined as in the SPOPN call.

rc   return code, required, one word, integer with values returned as follows:

```
-1 = parameter not valid
 0 = successful close
+1 = close error
```

23.3 SCHEDULING A SPOOL PRINT TASK

Spool print tasks reside on the VORTEX II foreground library. From 1 to 14 spool print tasks may be active concurrently within a VORTEX II system. A spool print task is scheduled by use of the Operator Communication SCHED or TSCHED directive.

EXAMPLE: SCHEDULE A SPOOL PRINT TASK

Request: to schedule the spool print task S$SP01 at priority level 8 from the foreground library with protection code F

Console Sequence: ;SCHED,S$SP01,8,FL,F   ENTRY

Action: The spool print task S$SP01 is scheduled for execution at priority level 8 as a VORTEX II foreground task.

23.4 SPOOL COMMUNICATION

The spool communication task, S$SPC, must be scheduled to enter commands to S$SPC or a spool print task. S$SPC is scheduled by use of the Operator Communication SCHED or TSCHED directive.

S$SPC is used to obtain spool file status display on the operator console or to delete spool files or to initiate communication with a spool print task.

Upon completion of processing of a spool file status or spool file delete request, S$SPC writes a prompt message to the operator console and waits for the next operator request to be entered. After initiation of communication to a spool print task, S$SPC exits from the VORTEX II system.

23.4.1 EX (Spool Print Task Exit)

End of communication and exiting of either S$SPC or a spool print task may be requested by use of EX.

The format of the EX command is as follows:

```
          EX
```

23.4.2 SG (Spool Print Task Go)

End of communication with a spool print task may be requested by use of SG, the spool print task go command. A spool print task upon recognition of the SG command resumes processing at the point of interruption to communicate with the operator console.

The format of the SG command is as follows:

```
          SG
```

23.5 INITIATION OF COMMUNICATION WITH A SPOOL PRINT TASK

After scheduling S$SPC, SB, the spool print break command is entered from the operator console to initiate communication to a spool print task. If communication is initiated to the spool print task specified in the spool print break command, S$SPC outputs a communication initiated message on the operator console and exits from the VORTEX II system. If the spool break command is not valid or communication cannot be initiated to the spool print task specified in the break command, then S$SPC outputs an error message on the operator console. In this case, if no further attempt to communicate with a spool print task is desired EX, the spool communication exit command, can be entered from the operator console to request S$SPC to exit from the VORTEX II system.

The format of the SB command is as follows:

```
          SB,nn
```

where

nn — is a two-digit number in the range 01-14 which specifies the spool print task with which communication is desired.

EXAMPLE: INITIATE COMMUNICATION WITH A SPOOL PRINT TASK

Request: initiate communication with the spool print task S$SP01. S$SPC, the spool communication task, is to be scheduled at priority level 9 and resides on the foreground library whose protection code is F.

```
Console Sequence:   ;SCHED,S$SPC,9,FL,F      ENTRY
                    S$SPC,SP**               PROMPT
                    SB,01                    ENTRY
                    S$SPC,SP05,01            RESPONSE
```

Action: communication is initiated with the spool print task S$SP01, the communication initiated message (S$SPC, SP05,01) is written to the operator console and S$SPC exits from the VORTEX II system.

23.6 SPOOL FILE STATUS DISPLAY

The status of spool files is obtained by use of the DS, display status, command. The status of the requested spool files is displayed on the operator console in the order of the spool files in the spool file directory. The format of the DS command is:

```
         ( ALL                    )
    DS, <               jobname   >
         ( printclass ,           )
```

where

ALL — constant 'ALL' specifies a display of the status of all spool files in the spool file directory

printclass — one alphabetic character (A through Z)

jobname — one to eight-character VORTEX II jobname; applicable only to spool file output from VORTEX II background tasks.

A spool file may have one of four possible status settings:

Status	Indication Status Display
open for write	S$SPO
available for print	0
open for print	S$SPnn
printed and held	nnnnn

where

S$SPO is the name of the spool output task

0 indicates that the spool file is complete and has not been printed.

S$SPnn is the name of the spool print task and nn is a two digit number in the range 01-14

nnnnn is the print count of the held spool file and is a five-digit number in the range 1-32767

EXAMPLE: SPOOL FILE STATUS DISPLAY

Request: obtain the status of all spool files within print class A

```
Console Sequence:    ;SCHED,S$SPC,9,FL,F         ENTRY
                     S$SPC,SP**                  PROMPT
                     DS,A                        ENTRY
                                                 DISPLAY
                     S$SPC,SP05,04               RESPONSE
```

Action: operator console display of the three spool files associated to class A followed by the status display completion message (S$SPC,SP05,04)

```
JOB      TASK     SPOOL    PRINT                DST
NAME     NAME     NAME     CLASS     LUN        NO.  STATUS

JOBONE   TASKA    S00036   A         5          43   S$SP01
JOBONE   TASKB    S00037   A         5          43        0
JOBONE   TASKC    S00038   A         5          43   S$SPO
```

23.7 DD (SPOOL FILE DELETE)

A Multitask Spool system command is available to delete spool file(s). The spool file delete command, DD, removes the requested spool file(s) from the spool file directory and deletes the requested spool file(s) from the RMD partition directories on which the spool file(s) reside. To be declared, a spool file must have a status of either closed or held.

The format of the DD command is:

```
       (                           spool   )
   DD, { print  , jobname  ,    filename  }
       ( class                             )
```

where

print class   one alphabetic character (A through Z)

jobname   one to eight character VORTEX II jobname

spool filename a 6 character spool filename

EXAMPLE: SPOOL FILE DELETE

Request: delete the closed spool file S00038.

```
Console Sequence:    ;SCHED,S$SPC,9,FL,F       ENTRY
                     S$SPC,SP**                PROMPT
                     DD,,,S00038               ENTRY
                     S$SPC,SP05,05             RESPONSE
```

Action: the spool file S00038 is removed from the spool file directory, deleted from the RMD partition directory on which it resided and the delete complete message (S$SPC,SP05,05) is written to the operator console.

23.8 SPOOL FILE PRINT

Three Multitask Spool system commands are available to print spool files. The three commands are:

Command Code — Description

PA — Spool File Print and Align; if the first record of the specified spool file is a forms control record, it is displayed on the operator console then the number of records specified in the PA command are written to the line printer specified in the PA command. Specifying the line printer lun as a dummy lun allows a forms control record to be displayed with no printing.

PH — Spool Print and Hold; executes the print request and does not delete spool file(s) after completion of print; if the first record of a spool file is a forms control record it is displayed on the operator console. Then the spool records are written to the line printer specified in the PH command.. If the print is of spool files in a print class and all spool files in the print class are printed then the print task waits for a time interval to lapse before searching the spool file directory for additional spool files to print.

PD — Spool Print and Delete; executes the print request and deletes each spool file after completion of print. As with PA and PH, a forms control record if present for a spool file is displayed on the operator console.

The format of the PA, PH, and PD commands is:

```
┌    ┐
│ PA │
│    │                        ┌                           ┐
│ PH │  , printer    print    │  spool           record   │
│    │       lun   , class    │, file name  ,  number     │
│    │                        └                           ┘
│ PD │
└    ┘
```

where

printer lun	a 1-3 digit number
print class	one alphabetic character (A through Z)
spool filename	a 6 character spool filename
record number	1-9 digit number; for PA specifies the number of records to be printed, for PH and PD specifies the record number within a spool file at which printing is to start.

EXAMPLE 1: SPOOL FILE PRINT AND ALIGN

Request: the spool print task S$SP01 is to print the first 10 records of the spool file S00037 from the print class A on the line printer to which the logical unit number 15 is assigned.

```
Console Sequence:    ;SCHED,S$SPC,9,FL,F        ENTRY
                     S$SPC,SP**                 PROMPT
                     SB,01                      ENTRY
                     S$SPC,SP05,01              RESPONSE
                     S$SP01,SP**                PROMPT
                     PA,15,A,S00037,11          ENTRY
                                                DISPLAY
                     S$SP01,SP05,06             RESPONSE
```

Action: The first record of the spool file S00037 if it is a forms control record is displayed on the operator console, the first 10 print records of the spool file are written to the line printer to which the lun 15 is assigned and the print and align complete message (S$SP01,SP05,06) is written to the operator console.

EXAMPLE 2: SPOOL PRINT AND HOLD

Request: the spool print task S$SP01 is to print all spool files generated for print class B on the line printer to which the logical unit number 15 is assigned.

```
Console Sequence:   ;SCHED,S$SPC,9,FL,F         ENTRY
                    S$SPC,SP**                  PROMPT
                    SB,01                       ENTRY
                    S$SPC,SP05,01               RESPONSE
                    S$SP01,SP**                 PROMPT
                    PH,15,B                     ENTRY
```

Action: each spool file in print class B that is available for printing (closed) is written to the line printer to which the lun 15 is assigned. When printing of all spool files in print class B is complete, the spool print task S$SP01 waits for a time interval to elapse and then searches the spool file directory for additional print class B spool files to print.

23.9 SC (SPOOL PRINT CANCELLATION)

Cancellation of the spool print currently in process by a spool print task is obtained by use of the SC command. If a spool file print is in progress by a print task to which this command is entered the spool file record # of the last printed is displayed on the operator console prior to output by the print task of a prompt message on the operator console. The format of the SC command is:

```
          SC
```

EXAMPLE: CANCEL A SPOOL PRINT IN PROGRESS

Request: to cancel the spool print of print class A by the spool print task S$SP01.

```
Console Sequence:   ;SCHED,S$SPC,9,FL,F              ENTRY
                    S$SPC,SP**                       PROMPT
                    SB,01                            ENTRY
                    S$SPC,SP05,01                    RESPONSE
                    S$SP01,SP**                      PROMPT
                    SC                               ENTRY
                    S00118,0000000047                RESPONSE
                    S$SP01,SP05,02                   RESPONSE
                    S$SP01,SP**                      PROMPT
```

Action: the spool print in progress by the spool print task S$SP01 of spool file S00118 within print class A is cancelled after the printing of record number 47. The spool print task S$SP01 is now waiting for a command to be entered by the console operator as indicated by the spool print prompt message (S$SP01,SP**)

23.10 ST (SPOOL PRINT TERMINATION)

Termination of the spool print currently progress is obtained by use of the ST command. If a spool file print is in progress by a print task to which this command is entered the spool file print continues to spool file end otherwise the spool print task outputs a prompt message to the operator console. The format of the ST command is:

```
ST
```

EXAMPLE: TERMINATE A SPOOL FILE PRINT

Request: to terminate the spool file print currently in progress by the print task S$SP01

Console Sequence:

```
;SCHED,S$SPC,9,FL,F          ENTRY
S$SPC,SP**                   PROMPT
SB,01                        ENTRY
S$SPC,SP05,01                RESPONSE
S$SP01,SP**                  PROMPT
ST                           ENTRY
```

Action: the spool print task S$SP01 completes the spool file print in progress, then waits for initiation of communication via S$SPC.

23.11 SPOOL FILE ALLOCATION

Spool file records are 80 words in length. Words 1-66 of a spool file record contain forms control or print data and words 67-80 contain spool file control information. Spool file records are written unblocked, one per RMD sector.

Spool files are created only on RMD partitions on the disc pack mounted on the drive to which the spool directory logical unit number, 107, is assigned.

Spool files are created with a primary space allocation as specified in the SPOPN subroutine call or with the default space allocation of 100 sectors. When the primary space allocation of a spool file is exhausted, the spool file is extended by the secondary space allocation specified in the SPOPN subroutine call or by the default space allocation of 100 sectors.

The search for space to allocate a spool file begins on the RMD partition on which the spool file directory resides. The search for space continues on each succeeding partition (until the end of the disc) until either the required amount of space is located or space allocation has been unsuccessfully attempted on each partition available. Note: Partitions with a protection key are used for allocation of spool files. If the space allocation search was not successful and it was for an amount of requested space greater than the default space allocation, then a search to allocate the default space amount is performed in the same manner as the previous search for space.

23.12 COMPONENT DESCRIPTION

The VORTEX II Multitask Spool system consists of six components:

1. VZSDA - spool dummy driver; VORTEX II nucleus, map 0, resident task

2. SPOPN - spool file open subroutine; resides in VORTEX II object module library

3. SPCLS - spool file close subroutine, resides in VORTEX II object module library

4. S$SPO - spool output task; VORTEX II foreground task with a resident TIDB

5. S$SPC - spool print communication task; VORTEX II foreground task with a non-resident TIDB

6. S$SP01 - spool print task; VORTEX II foreground task with a non-resident TIDB

VZSDA, the spool dummy driver, accepts spool file create (open), write, and close requests from either background or foreground tasks for logical units assigned to VZSDA. These requests are passed by VZSDA to S$SPO, the spool output task. VZSDA upon indication by S$SPO of request processing completion posts an I/O completion status to the initiating task.

A call to SPOPN, the spool file open subroutine, requests the creation of a spool file. The SPOPN call specifies for the spool file to be created the spool file RMD space allocation, the logical unit that will be used in the requesting task's WRITE statement to direct output to a spool file, and a print class to be associated with the spool file.

A print class is a single alphabetic character (A through Z) used to relate individual spool files. This relationship or grouping together of individual spool files may be for any desired reason but most often is based on the importance or similarity of information in the associated spool files. All spool files associated with a given print class may be printed by the computer operator by means of a print request that designates the print class to be printed and the line printer lun to which the print output is to be directed. This facility allows printing of spool files with no knowledge required of names of tasks that generated the information in the spool files or names of the spool files associated to a print class.

A spool file may also be created without the use of the SPOPN call. Upon each WRITE from a background or foreground task to a logical unit assigned to VZSDA, the spool output task, S$SPO, checks to see if a spool file has been created (opened) for the task issuing the WRITE and the logical unit number to which the WRITE is directed. If a spool file is not found to be open, then the spool output task creates a spool file with a default RMD space allocation and a default print class.

A call to SPCLS, the spool file close subroutine, is a request to end a spool file for the task issuing the SPCLS call and for the logical unit number specified in the call. A spool file may also be closed without use of the SPCLS call. At least once every 500 milliseconds S$SPO, the spool output task, scans a core table containing an entry for each open spool file to determine if the task that requested opening of the spool file remains active within the VORTEX II system. Open spool files for which the associated task is not active are closed and removed from the core table of open spool files.

S$SPO, the spool output task, processes spool file created (open), write, and close requests from VZSDA, the spool dummy driver. S$SPO also performs spool file close processing based on a core table of open spool files as previously described in the discussion of the SPCLS subroutine. After processing an open, write or close request, S$SPO informs VZSDA of the result of request processing and waits for either another request to process from VZSDA or a time interval to lapse before resuming execution.

S$SPC, the spool print communication task, is scheduled by the computer operator for spool file status display or spool file delete or to initiate communication with a spool print task.

The Multitask Spool system allows from 1 to 14 spool print tasks to be concurrently active within a VORTEX II system. Spool print tasks are named S$SP01 through S$SP14. One spool print task, S$SP01, is standard at VORTEX II system generation time. The System Generation Procedures in the users guide contain instructions for use of the File Maintenance ENTER directive to obtain multiple spool print tasks.

A spool print task is scheduled by entry of the SCHED directive from the operator console. A spool print task once it is scheduled remains active until terminated by entry of the EX, spool print task exit command from the operator console. Use of the ABORT directive to terminate a spool print task may result in unpredictable Multitask Spool system operation.

The following Multitask Spool system actions are controlled by a spool print task and are available from each scheduled spool print task.

- o Spool File Print
- o Spool File Print Cancellation
- o Spool File Print Termination
- o Spool Print Task Exit

When a spool print task is scheduled or when a scheduled spool print task recognizes a communication request from S$SPC, the spool print communication task, it writes a prompt message to the operator console and waits for an operator entry. If the operator entry is a valid command for a spool print task then the action directed by the entered command is taken otherwise an error message and then the prompt message are written to the operator console and the spool print task again waits for an operator entry.

23.13 SYSTEM GENERATION

To incorporate the Multitask Spool sub-system into a VORTEX II system, include the following EQP directive at SYSGEN time:

```
EQP,SD0A,0,1,0,0
```

This will incorporate the dummy driver front-end to spool. Include from one (1) to fifteen (15) ASN directives for each spool logical unit desired. For example, if it is desired to incorporate three spool channels using logical unit numbers six (6), forty (40) and forty-one (41), include:

```
          ASN,6=SD00
          ASN,40=SD00
          ASN,41=SD00
```

Include an ASN directive for logical unit number 107; it must be assigned to an unprotected RMD partition. For example,

```
          ASN,107=D00J
```

Include an ASN directive for logical unit numbers 108 and 109; they must be assigned to a dumy device as follows:

```
          ASN,108=DUM
          ASN,109=DUM
```

Include an ASN directive for logical unit numbers 110 through 123 as needed. 110 is used by print task number one (S$SP01), 111 is used by print task number two (S$SP02), etc. They must be assigned to a dummy device. For example, for a system using spool print tasks S$SP01,S$SP02 and S$SP04, include:

```
          ASN,110=DUM
          ASN,111=DUM
          ASN,113=DUM
```

Spool print tasks in addition to S$SP01 are obtained by use of the File Maintenance ENTER directive. For example, to reference also the spool print task S$SP01 as S$SP02 and S$SP04 from the foreground library with logical unit number 106 and protection code F, use the File Maintenance ENTER directive as follows:

```
          ENTER,106,F,S$SP01,S$SP02
          ENTER,106,F,S$SP01,S$SP04
```

23.14 COMMAND SUMMARY

The general form of a Multitask Spool command is

```
          cc,p(1),p(2),...p(n)
```

where

cc is a two-character alphabetic command code

each p(n) is a command parameter

Multitask Spool commands may begin at any console position and comprise sequences of character strings and delimiters with embedded blanks allowed but ignored in the interpretation of the command. The characters strings, parameters, are separated by delimiters commas (,) or equal signs (=). Although not required, a period (.) is a command terminator. Comments may be inserted after a period. Optional periods and the optional replacement of commas by equal signs are omitted from the command summary.

Command Code	Command	Valid Forms
SB	Print Task Break	SB,nn
EX	Communication or Print Task Exit	EX
DS	Spool File Status Display	DS,ALL DS,print class DS,print class, jobname DS,,jobname
PA	Print and Align	PA, printer lun,print class, spool filename, #record print PA,printer lun, print class,, #records print
PH	Print and Hold	PH,printer lun,print class PH,printer lun,print class, spool filename PH,printer lun,print class, spool filename,starting record#
PD	Print and Delete	PD,printer lun,print class PD,printer lun,print class, spool filename PD,printer lun,print class, spool filename,starting record#

Command Code	Command	Valid Forms
DD	Spool File Delete	DD,print class
		DD,print class,jobname
		DD,,jobname
		DD,print class,,spool filename
SC	Print Cancel	SC
ST	Print Terminate	ST
SG	Print Task Go	SG

23.15 CONSOLE MESSAGES

The general form of a Multitask Spool console message is:

```
st,ms,mm
```

where

st	is a 5 or 6 character Multitask Spool taskname
ms	is a 4 character message
mm	is a 2 digit message modifier

The Multitask Spool tasknames that appear in console messages are:

S$SPC	Spool Print Communication Task
S$SPnn	Spool Print Task

where

nn	is a two-digit number that identifies a particular print task
S$SPO	Spool Output Task

CONSOLE MESSAGES AND MODIFIERS

SP** prompt to console

SP01 command error

SP02 requested resource in use

- 01 communication with print task cannot be initiated: communication request to another print task is outstanding
- 02 print task cannot be activated: print task identical name is already active
- 03 print request cannot be honored: print is in progress by print task
- 04 exit request cannot be honored: print is in progress by print task

SP03 requested resource not available

- 01 communication to print task cannot be initiated: specified print task is not active
- 02 spool input task bootstrap initialization not successful: no lun available for spool file directory and/or at least one spool print task
- 03 spool file directory full; directory pack initiated
- 04 spool file directory full; directory pack did not free directory entries
- 05 spool file name not found

SP04 input/output (I/O) error

- 01 spool file create error
- 02 spool file directory open error
- 03 spool file directory read/write error
- 04 spool file open error
- 05 spool file read/write error
- 06 printer write error

SP05	requested service complete	
	01	print task communication initiation
	02	print cancel
	03	print task terminate
	04	spool file status display
	05	spool file delete
	06	spool file print align
	07	spool file print
	08	spool file directory pack

SECTION 24
TAPE LABELING

24.1 INTRODUCTION TO TAPE PROCESSING

Labels are used to identify magnetic tape reels and the files they contain. With VORTEX II, tapes can be processed with or without labels. However, the use of labels is recommended as a basis for efficient control of tape libraries.

VORTEX II tape labels consist of reel labels and file labels, herein referred to as volume labels and data set labels respectively. The volume label is the first record on the tape; it identifies the volume (reel) and its owner. The labels preceding the data set (file) are called header labels, while the labels following the data set are called trailer labels; these labels identify and describe the data set. VORTEX II labeled tapes have the following basic layout:

VOLUME LABEL	DATA SET HEADER LABELS	TM	DATA SET	TM	DATA SET TRAILER LABELS	TM	TM

Where TM identifies a tapemark (filemark)

VORTEX II tapes without labels contain only data sets and tapemarks, as shown in the following layout:

DATA SET	TM	TM

24.2 LABEL DEFINITIONS

VORTEX II supports label processing for 9-track tapes at densities of 800 bpi and 1600 bpi. Labels are 40 word (80 byte) records encoded in EBCDIC and odd parity. The first two words (four bytes) identify the labels as follows:

VOL1	Volume label
HDR1 and/or HDR2	Header labels
EOV1 or EOV2	Trailer labels (end of volume)
EOF1 or EOF2	Trailer labels (end of data set)

The HDR1/EOV1/EOF1 labels use identical formats; the HDR2/EOV2/EOF2 labels use identical formats.

Format of Volume Label:

Bytes	Contents	Description
1-4	VOL1	Label Identifier
5-10	numeric	Volume Serial Number - a unique identification code assigned to the volume when it enters the system. This code may be from one to six digits but, if fewer than six digits, the code must be left justified with trailing blanks.
11	0	Reserved
12-41	b̸	Reserved
42-51	alphanumeric	Owner Code - a one to ten character code identifying the owner of the volume. If the code is less than ten characters long, it is left-justified with trailing blanks.
52-80	b̸	Reserved

Format of Standard Data Set Label 1:

Bytes	Contents	Description
1-4	HDR1	Header Label (at the beginning of a data set)
	EOV1	Trailer Label (at the end of a tape volume)
	EOF1	Trailer Label (at the end of a data set)
5-21	alphanumeric	Data Set Name - a one to seventeen character name for the data set. If the name is less than seventeen characters long, it is left justified with trailing blanks. The name may contain embedded blanks.
22-27	numeric	Data Set Serial Number - see volume serial number. Contains the volume serial number of the first or only tape volume.

Bytes	Contents	Description
28-31	numeric	Volume Sequence Number - a number (0001 to 9999) that indicates the order of the volume within a multi-volume group.
32-35	numeric	Data Set Sequence Number - a number 0001 to 9999) that indicates the order of the data set within a multi-data-set group.
36-41	numeric	Reserved.
42-47	ƀyyddd	Creation Date - yy is the creation year (00 to 99) and ddd is the creation day (001 to 366)
48-53	ƀyyddd	Expiration Date - yy is the expiration year (00 to 99) and dd is the expiration day (001 to 366).
54	0	Reserved.
55-60	numeric	Block Count - this field is always zero (000000) in the header label; this field contains the number of physical records in the data set on the current volume (000000 to 999999) in the trailer labels.
61-80	ƀ	Reserved.

Format of Standard Data Set Label 2:

Bytes	Contents	Description
1-4	HDR2	Header Label (at the beginning of a data set)
	EOV2	Trailer Label (at the end of a tape volume)
	EOF2	Trailer Label (at the end of a data set)

5	alpha	Record Format - a single letter identifying the format of the records of the data set:
		F = Fixed length
		V = Variable length
		U = Undefined length

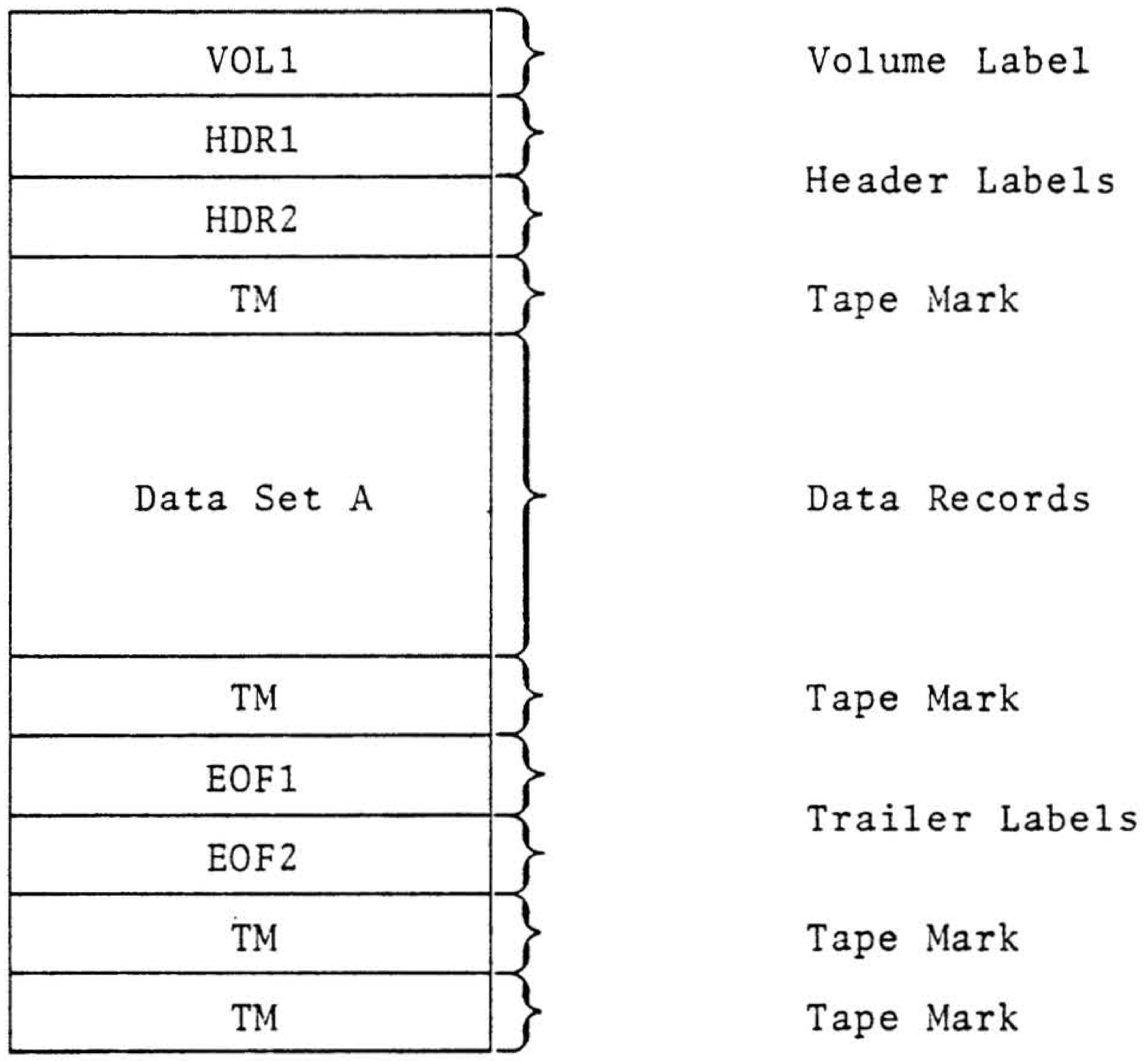
Bytes	Contents	Description
6-10	numeric	Block Length - A number (00000 to 32760) specifying the physical record length in bytes.
11-15	numeric	Record Length - A number (00000 to 32760) specifying the logical record length in bytes.
16	numeric	Tape Density - A single digit identifying the recording density of the tape: 2 = 800BPI (NRZI) 3 = 1600BPI (PE)
17	numeric	Data Set Position - A single digit indicating volume switch: 0 = No volume switch has occurred 1 = Volume switch has occurred
18-36	b̸	Reserved

37	alpha	Control Characters - A one letter code indicating whether a control character set was used to create the data set:
		A = ASCII control characters
		M = Machine control characters
		b̸ = No control characters

38	b̸	Reserved
39	alpha	Block Attribute - A one letter code indicating the block attribute: B = Blocked records S = Spanned records R = Blocked and spanned records b̸ = No blocked and no spanned records

24.3 LABEL ORGANIZATION

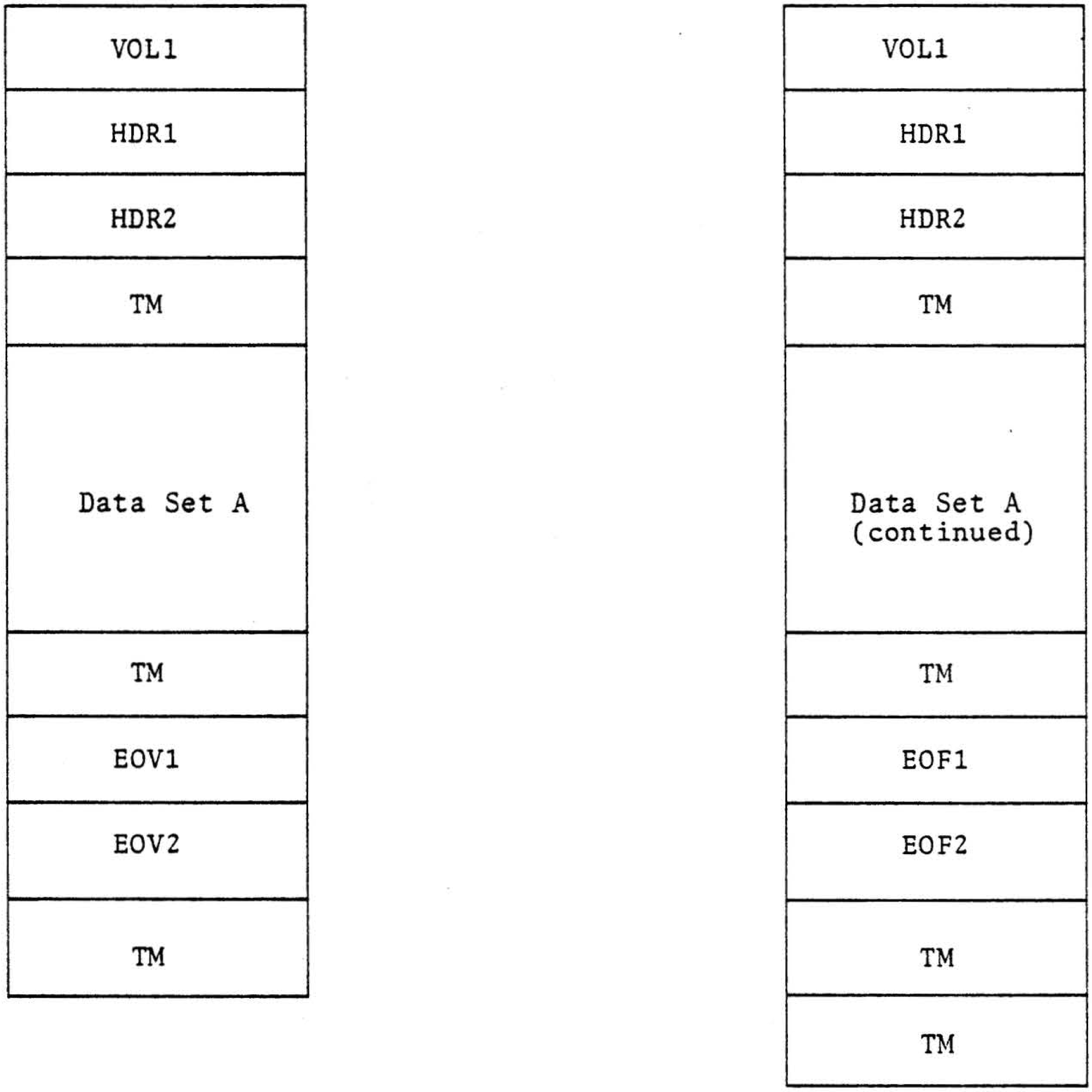
VORTEX II labeled tapes may contain one or more data sets, and data sets may reside on one or more volumes. The following layouts illustrate four typical organizations for standard label tapes.



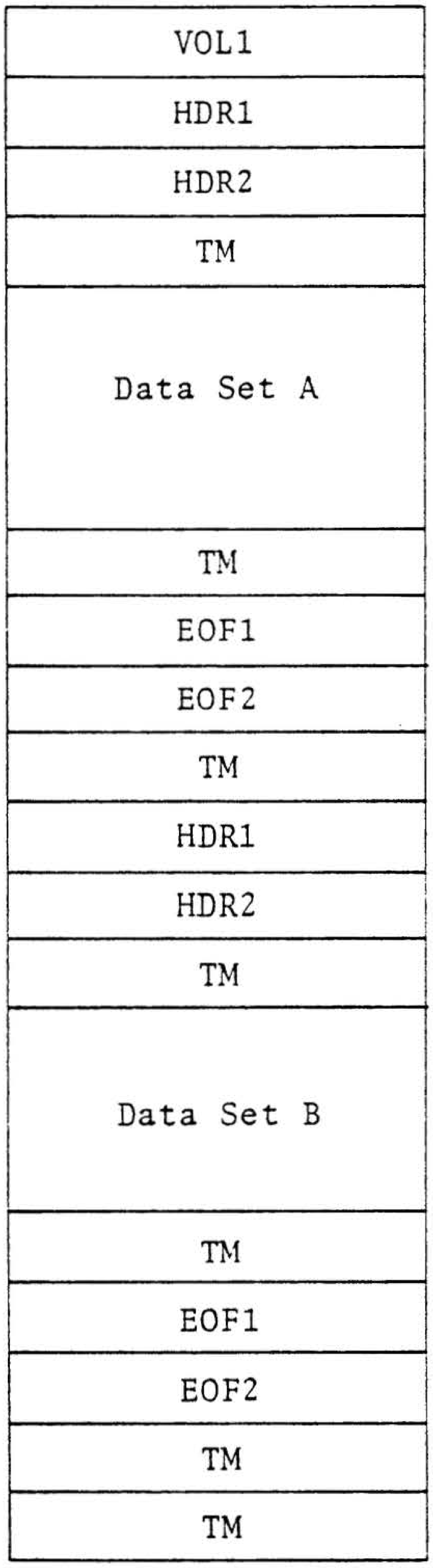
Single Data Set - Single Volume

Vol 1 of 2

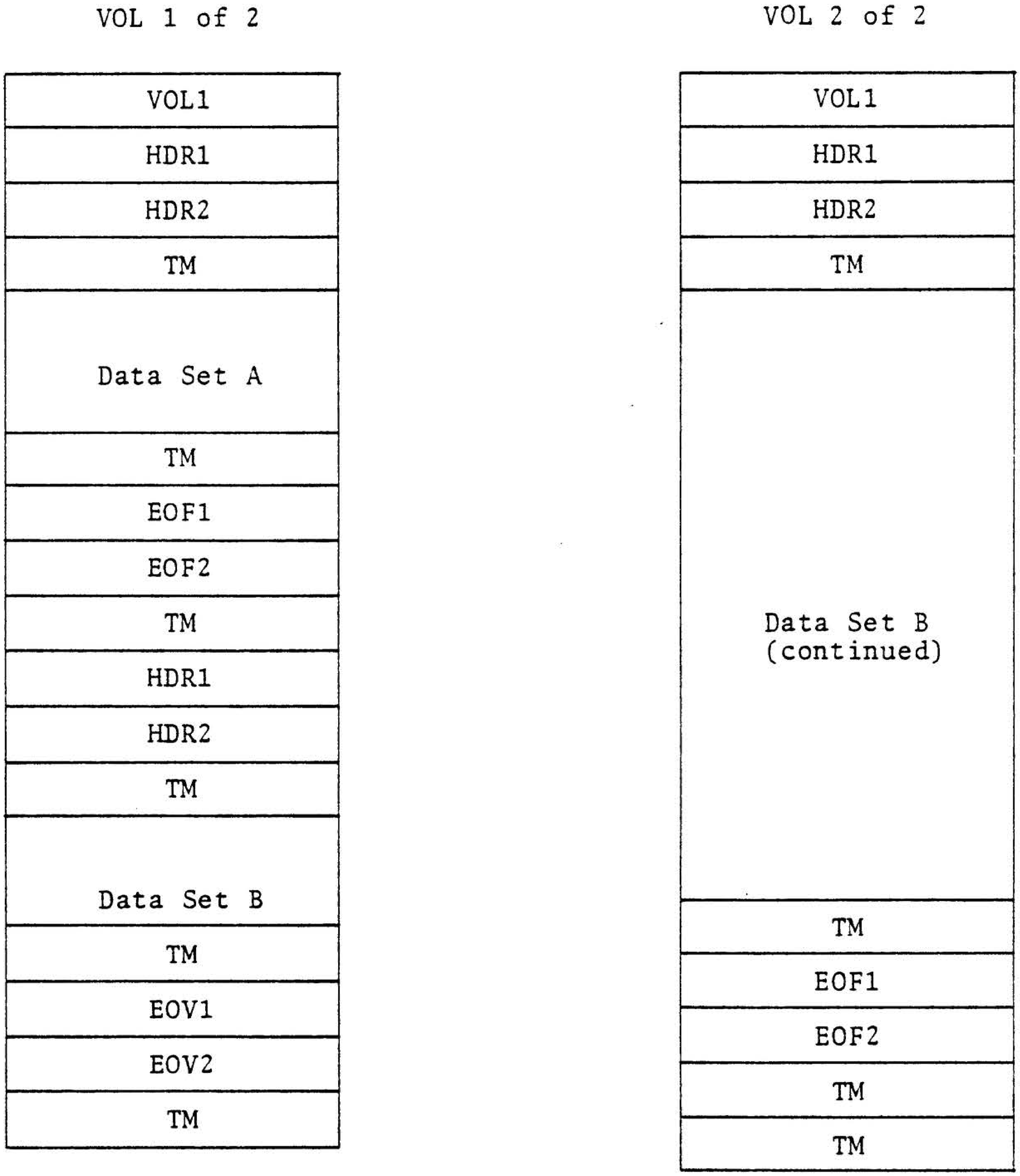
Vol 2 of 2



Single Data Set - Multiple Volumes



Multiple Data Sets - Single Volume



Multiple Data Sets - Multiple Volumes

24.4 DATA SET DEFINITION

Data sets are defined via TFILE commands to the Job Controller Processor (JCP) for background tasks, or to the Operator Communication Task (OPCOM) for foreground tasks. The format of this command is:

```
/TFILE,LUN,IO {,M,ID,L,T,SN,DSN,DN {,RF,BL,RL,TD,CC,BA,EXP} }

     or

;TFILE,LUN,IO {,M,ID,L,T,SN,DSN,DN {,RF,BL,RL,TD,CC,BA,EXP} }
```

where

Name	Description	Default
LUN	Logical Unit name or Number - must be assigned to magnetic tape.	-

IO	I/O Option:
	I = Input data set
	O = Output data set
		-

M	Mount Option:
	I = Immediate mount
	D = Deferred mount
		D

ID	Identification - a one to six character alphanumeric identifier which serves as a reference to the data set definition.	JCP or OPCOM

L	Label Option:
	B = Bypass labels
	S = Standard labels
	U = No labels
		U

T	Translate Option:
	N = No translation
	Y = Translate from EBCDIC to ASCII if IO = I
	Y = Translate from ASCII to EBCDIC if IO = O
		N

SN	Serial Number - a one to six character alphanumeric number specifying the first or only volume serial number. This field is required if IO = I and L = S or L = B.	b̸

Name	Description	Default
DSN	Data Set Name - a one to seventeen character name corresponding to the desired data set name - must be enclosed in apostrophes if the name contains embedded blanks or other special characters	b̸
DN	Data Set Number - a one to four digit number corresponding to the ordinal data set number	1

RF	Record Format:
	F = Fixed length records
	V = Variable length records
	U = Undefined length records
		U

BL	Block Length - a one to five digit number specifying the block length in bytes	32760
RL	Record Length - a one to five digit number specifying the record length in bytes	0
TD	Tape Density: 2 = 800 bpi 3 = 1600 bpi	3

CC	Control Characters:
	A = ASCII control characters
	M = Machine control characters
		b̸

BA	Block Attribute:
	B = Blocked records
	S = Spanned records
	R = Blocked and spanned records
		b̸

EXP	Expiration - a one to four digit number specifying the number of days until the data set expires	30

Example: to define an unlabeled output data set which is to be written in ASCII, the following command suffices:

```
/TFILE,22,O
```

Example: to define a labeled input data set residing on file two of tape reel 123456, in EBCDIC, and named NEWMASTER, the following command is required:

```
/TFILE,22,I,,,S,Y,123456,NEWMASTER,2
```

Example: to access the same input data set, given that the data set name is not known, the following command is required:

```
/TFILE,22,I,,,B,Y,123456,,2
```

A special facility is provided for tape volumes which do not contain labels. The term data set is defined as the set of all data sets; that is to say, all data sets, even though separated by tapemarks, are considered to be a single data set. This facility is provided primarily to enable VORTEX II users who have implemented the tape labeling system to access tapes in a fashion similar to VORTEX II users who have not. Only a single TFILE command is required to access an unlabeled tape.

Data set definitions (TFILE commands) are queued to a particular physical magnetic tape drive at the time they enter the system; the definitions are activated when they are at the head of the queue to a particular tape drive and an open request is made to that tape drive; and the definitions are de-queued when they are active and an error or close request occurs on the tape drive. The OPCOM command

```
;TLIST
```

will result in the listing upon OC of all active and queued data set definitions in sequence by controller-subchannel number. The general format of this listing is:

```
MTmn IO ID LT SN DSN STATUS
```

where

m   is the controller number

n   is the subchannel number

IO   is RD for input data sets, or
      WT for output data sets

```
ID        is the ID parameter from the TFILE command

LT        is NL for no labels, or
             BL for bypass labels, or
             SL for standard labels

SN        is the SN parameter from the TFILE command

DSN       is the DSN parameter from the TFILE command

STATUS    is QUEUED if data set not yet accessed, or
             ACTIVE if data set being accessed.
```

For example, given that the three TFILE command examples were entered and that logical unit number 22 was assigned to MT00, a TLIST command would give:

```
MT00  WT   JCP   NL                               QUEUED

MT00  RD   JCP   SL    123456   NEWMASTER         QUEUED

MT00  RD   JCP   BL    123456                     QUEUED
```

If the first data set was then opened, the TLIST command would give:

```
MT00  WT   JCP   NL                               ACTIVE

MT00  RD   JCP   SL    123456   NEWMASTER         QUEUED

MT00  RD   JCP   BL    123456                     QUEUED
```

If the first data-set was then closed, the TLIST command would give:

```
MT00  RD   JCP   SL    123456   NEWMASTER         QUEUED

MT00  RD   JCP   BL    123456                     QUEUED
```

An additional OPCOM command

```
        ;TPURGE,ID
```

where

```
     ID   is the ID parameter from the TFILE command
```

will result in the deletion from the system of all non-active data set definitions with a matching ID. For example, assume a TLIST command results in the following print-out:

```
MT00   RD   JCP       SL    107530   INVENTORY       QUEUED

MT00   RD   UPDATE    SL    106044   EXTRACT         QUEUED

MT10   RD   JCP       SL    108665   OLDMASTER       ACTIVE

MT10   WT   UPDATE    SL    106047   TESTTAPE        QUEUED
```

the command

```
;TPURGE,JCP
```

would result in the data set definition for INVENTORY being deleted. EXTRACT and TESTTAPE would not be deleted since the IDs differ; OLDMASTER would not be deleted since it is already being accessed (i.e., it is active).

24.5 TAPE MOUNTING

When the operating system receives a request to open a data set (or when volume switching occurs), a mount message is issued to the operator (via OC). The general format of this message is

```
I037, MTmnxx
```

where

m   is the controller number

n   is the subchannel number

xx   is MU for Mount Unlabeled tape, or
     ML for Mount Labeled tape, or
     SU for mount Scratch Unlabeled tape, or
     SL for mount Scratch Labeled tape, or
     NU for mount Next volume of Unlabeled tape, or
     NL for mount Next volume of Labeled tape.

The operator is then responsible for mounting the proper tape on the selected tape drive and keying:

```
;RESUME,MTmnxx
```

If the operating system is able to detect that the mounted tape does not meet the requirements of the data set definitions, the tape is rewound and the message repeated. The operator should mount the proper tape and restart the tape drive. If the operator does not know which specific tape is being requested, he should key in

```
;TLIST
```

to obtain a listing of the current data set definitions. The entry marked ACTIVE for the tape drive in question corresponds to the data set being requested by the operating system. If the operator is unable to mount the proper tape (i.e., if the data set definition is in error), he should key in

```
;DEVDN,MTmn
;RESUME,MTmnxx
```

This will clear the mount request from the system and cause the open to complete with an error. The operator should then key in

```
;DEVUP,MTmn
```

in order to clear the status of the affected tape drive.

24.6 DATA SET ACCESS

User tasks accessing labeled tapes are responsible for performing physical I/O in such a fashion as to maintain consistency between the definitions of the data set labels and the actual contents of the data set; in other words, the operating system is responsible for processing the labels and the data set definitions (TFILE commands), while the user is responsible for processing the data set and creating the data set definitions. In all other respects, however, access to magnetic tape is identical in operation to access to any other VORTEX II I/O device: the I/O macros are identical in format; the DCB macros are identical in usage; error status and data transfer length are returned in the I/O macro; etc.

24.7 INPUT DATA SETS

24.7.1 Opening an Input Data Set

An input data set can be opened by (1) specifying immediate mount in the TFILE request, (2) performing an OPEN from a user program, or (3) performing the first READ from a user program.

If standard label processing is specified, the first record on the tape is inspected to ensure that it is a VOL1 label: the serial number of the VOL1 label must match the serial number in the data set definition. The next record on the tape is inspected to ensure that is is a HDR1 label: the volume sequence number and the file sequence number of the HDR1 label must both be one (i.e., multi-volume groups must be accessed starting with the first volume). If necessary, successive data sets are skipped until positioning to the data set number specified in the TFILE request is achieved. The HDR1 label of the data set is inspected;

the data set name of the HDR1 label must match the data set name of the TFILE request. The tape is positioned ready to input the first data record. If the data set is being opened via READ, the first data record is input to the user program.

If bypass label processing is specified, a similar procedure is performed except that the data set name is not validated.

If no label processing is specified, the first record on the tape is inspected to ensure that it is not a VOL1 label. The tape is rewound. If the data set is being opened via READ, the first data record is input to the user program.

24.7.2 Accessing an Input Data Set

An input data set can be accessed only if properly opened. A READ request retrieves the next tape record and may perform volume switching. An SREC forward request spaces over the next tape record and may perform volume switching. An SREC backward request backspaces over the previous tape record unless the tape is positioned to the first record of the data set on a given volume; in which case, no tape movement occurs and end-of-file status is returned. WRITE and WEOF requests are invalid. FUNC requests perform no operation.

24.7.3 Closing an Input Data Set

If standard or bypass label processing is specified, an input data set can be closed in three ways. If a READ or an SREC forward request results in an end-of-file condition (as opposed to an end-of-volume condition), the data set is closed automatically and end-of-file status is returned. A CLOSE request skips successive records until an end-of-file condition occurs and closes the data set. An REW request closes the data set and rewinds the current volume.

If no label processing is specified, an input data set can be closed only by a REWIND request. Such request closes the data set and rewinds the current volume. A READ or SREC forward request resulting in an end-of-file condition returns end-of-file status but does not close the data set.

24.8 OUTPUT DATA SETS

24.8.1 Opening an Output Data Set

An output data set can be opened by (1) specifying immediate mount in the TFILE request, (2) performing an OPEN from a user program, or (3) performing the first WRITE from a user program.

If standard label processing is specified, the first record on the tape is inspected to ensure that it is a VOL1 label: the serial number of the VOL1 label must match the serial number in the data set definition. If the data set number in the TFILE request is not one (i.e., if a data set is being appended to an existing data set), the next record on the tape is inspected to ensure that it is a HDR1 label: the volume sequence number and the file sequence number of the HDR1 label must both be one (i.e., multi-volume groups must be accessed starting with the first volume). If necessary, successive data sets are skipped until positioning to the data set number specified in the TFILE request is achieved. The next tape record is accessed to determine whether or not an attempt to over-write an existing data set is being made. If a HDR1 label is encountered, the data set must be expired as determined by the expiration date of the HDR1 label. The header labels are written to the tape; the tape is positioned ready to output the first data record. If the data set is being opened via WRITE, the first data record is output from the user program.

If bypass label processing is specified, the tape is considered to be positioned ready to output the first data record. No protection is afforded the user. This technique should only be used in order to initialize new tapes. If the data set is being opened via WRITE, the first data record is output from the user program.

If no label processing is specified, the first record on the tape is inspected to ensure that it is not a VOL1 label. The tape is rewound. If the data set is being opened via WRITE, the first data record is output from the user program.

24.8.2 Accessing an Output Data Set

An output data set can be accessed only if properly opened. A WRITE request outputs the next tape record and may perform volume switching (including writing end-of-volume labels and header labels). READ and SREC requests are invalid. A FUNC 037 (31) forces volume switching; other FUNC requests perform no operation.

24.8.3 Closing an Output Data Set

If standard label processing is specified, an output data set can be closed in three ways. A WEOF request writes the end-of-file labels and closes the data set. A CLOSE request writes the end-of-file labels and closes the data set. An REW request writes the end-of-file labels, closes the data set, and rewinds the current volume.

If bypass or no label processing is specified, an output data set can be closed only by an REW request. Such a request writes a tape mark, closes the file, and rewinds the current volume. A CLOSE or WEOF request writes a tape mark but does not close the data set.

24.9 TAPE INITIALIZATION

All magnetic tapes must be initialized prior to first usage. The tape labeling system accesses the first record of a tape volume during open processing; if the tape has not been initialized (i.e., nothing has ever been written on the tape), this access will result in an I/O error and the tape will spin off the end of the reel. A VOL1 label must be written onto standard label tapes (the serial number assigned to the tape must of course match the serial number of the VOL1 label); while anything but a VOL1 label (usually a tapemark) must be written onto unlabeled tapes. The following two jobstreams will initialize either a standard label tape or an unlabeled tape respectively.

```
/JOB,INITIALIZE SL TAPE
/TFILE,xx,O,I,INITAP,B,Y
/IOUTIL
COPYR,1,SI,1,40,7,1,40
VOL1ssssss0ƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀƀnnnnnnnnnn
/WEOF,xx
/REW,xx
/FINI
```

where

xx — is the logical name or number assigned to the mag tape driver

ssssss — is the serial number (left justified)

nnnnnnnnnn — is the owner name and address code

```
/JOB,INITIALIZE NL TAPE
/TFILE,xx,O,I,INITAP,B,Y
/WEOF,xx
/REW,xx
/FINI
```

Note that tapes must be initialized at the tape density at which they will be used.

24.10 SYSTEM GENERATION

To incorporate label processing into a VORTEX II system, simply include an EQP directive at SYSGEN time specifying model code MTnB instead of MTnA for each magnetic tape controller. The magnetic tape driver (VZMTB) responsible for performing the standard label processing performs recovery from parity errors detected during WRITE operations by (1) backspacing over the record in error, (2) writing a tape mark, (3) backspacing over the tape mark, and (4) attempting the WRITE again. This forces a longer gap between data records; thus, successive attempts to write the record are performed farther and farther along the tape and cause longer and longer gaps. Thus, the bad section of the tape is hopefully bypassed and left unused. For this reason, the retry count of the EQP directive should be set reasonably high. A retry count of ten, for example, allows for about five inches of bad tape, twenty allows for ten inches, etc.

24.11 ERROR MESSAGES

The following errors are recognized by the tape labeling system:

Error	Description
IO00	Unit not ready
IO07	Access error. READ attempted on output file, WRITE attempted on input file, etc.
IO10	Data set not found or has expired.
IO12	Data Set definition missing.
IO14	Timeout error
IO20	BIC error
IO37	Volume mount request
IO46	Unassigned memory error

All I/O errors cause the offending tape reel to be rewound; if a data set is open to the drive, it is closed.

24.12 EXAMPLES OF USAGE

The examples below assume the following configuration:

One 800 bpi tape drive (MT00)

One 1600 bpi tape drive (MT10)

and also assume that logical unit number 22=MT00 and logical unit number 26=MT10.

Example 1

Copy an unlabeled 800 bpi tape containing 3 files onto an unlabeled 1600 bpi tape.

```
/JOB,COPY
/TFILE,22,I,D,INPUT
/TFILE,26,O,D,OUTPUT
/IOUTIL
COPYF,3,22,0,0,26,0,0
/REW,22
/REW,26
/FINI
```

COMMENTS

This causes at execution time the message

```
I037,MT00MU
```

to be output to OC. The operator should then mount the tape to be copied onto the 800 bpi drive (if he has not already done so) and key in

```
;RESUME,MT00MU
```

Almost immediately the message

```
I037,MT10SU
```

will be output to OC. The operator should then mount a 1600 bpi unlabeled scratch tape onto the 1600 bpi drive (if he has not already done so) and key in

```
;RESUME,MT10SU
```

The three files will then be copied and both drives rewound, at which point the tapes may be dismounted.

Example 2

Copy the input tape from Example 1 onto a 1600 bpi tape, but make each file a separate data set (i.e., the output tape will be labeled). Assume that the first file contains only 40 word records (in SEDIT format), the second file contains 60 word records (in object module format), and the third file contains 40 word records blocked six to one (i.e, 240 word blocks). Further, data set 1 is to be called SOURCE, data set 2 is to be called RELOCATABLE, and data set 3 is to be called TEST-DATA. The files are to be placed onto reel 123456.

```
/JOB,SPLIT
/TFILE,22,I,D,INPUT
/TFILE,26,O,D,FONE,S,,123456,SOURCE,1,F,80,80,3,,B,100
/TFILE,26,O,D,FTWO,S,,123456,RELOCATABLE,2,F,120,120,3,,B,100
/TFILE,26,O,D,FTRE,S,,123456,TEST-DATA,3,F,480,80,3,,B,100
/IOUTIL
COPYF,1,22,0,40,26,0,40
COPYF,1,22,0,60,26,0,60
COPYF,1,22,0,240,26,0,240
/REW,22
/REW,26
/FINI
```

COMMENTS

At the console the same procedure is followed for mounting the input tape. However, the mount request for the output tape will appear as

```
IO37,MT10ML
```

The operator should then mount the 1600 bpi standard label tape 123456 on the 1600 bpi drive and key in

```
;RESUME,MT10ML
```

Since the magnetic tape driver retains information regarding the current tape volume mounted after a data set is closed, no mount requests are made for RELOCATABLE and TEST-DATA.

Example 3

Perform the same procedure as in Example 2, but copy the data sets onto a labeled scratch tape.

The only change required in the jobstream is the replacement of the TFILE commands for SOURCE, RELOCATABLE, and TEST-DATA. They should be

```
/TFILE,26,O,D,FONE,S,,,SOURCE,1,F,80,80,3,,B,100
/TFILE,26,O,D,FTWO,S,,,RELOCATABLE,2,F,120,120,3,,B,100
/TFILE,26,O,D,FTRE,S,,,TEST-DATA,3,F,480,80,3,,B,100
```

COMMENTS

The request for the 1600 bpi drive will appear as

```
IO37,MT10SL
```

The operator should tten mount a 1600 bpi standard label scratch tape and key in

```
;RESUME,MT10SL
```

Example 4

Assuming that Example 3 resulted in the three files being written onto reel 777777, add the object module RELOCATABLE from this tape to the object module library.

```
/JOB,ADD
/TFILE,26,I,,,S,,777777,RELOCATABLE,2
/FMAIN
INPUT,26
ADD,OM,D
/REW,26
/FINI
```

COMMENTS

At the console, the mount request will appear as

```
IO37,MT10ML
```

The operator should mount reel 777777 on MT10 and key in

```
;RESUME,MT10ML
```

Example 5

List the first file of the Example 3 tape, assuming the name of the data set is not known.

```
     /JOB,LIST
     /TFILE,26,I,,,B,,777777,,1
     /SEDIT
     AS,IN,26
     AS,OU,DUM
     LI
     FC
     /REW,26
     /FINI
```

COMMENTS

Mounting procedure is the same as in Example 4.

Example 6

Backup the exceedingly large disc file LARGE on lun 8 (SS) onto magnetic tape. Assume that the file contains 120-word records which are not blocked, that the file will not fit on one reel of tape, and that a tape without labels is to be used.

```
     /JOB,TWO TAPES
     /TFILE,22,O
     /IOUTIL
     PFILE,8,,120,LARGE
     COPYF,1,8,0,120,22,0,120
     /REW,22
     /FINI
```

COMMENTS

At execute time the mount request

```
     IO37,MT00SU
```

will appear. The operator should mount an 800 bpi unlabeled scratch tape on MT00 and key in

```
     ;RESUME,MT00SU
```

When the first reel is filled, the drive will rewind and the message

```
     IO37,MT00SU
```

will appear. The operator should dismount the first reel and label it reel 1 of 2, mount another 800 bpi unlabeled scratch tape, and key in

```
;RESUME,MT00SU
```

Upon job completion, the operator should dismount the second reel and label it 2 of 2.

Example 7

Place the same disc file onto a labeled tape using data set name LARGE.

The only change required in the jobstream of Example 6 is the replacement of the TFILE command. It should be

```
/TFILE,22,O,,,S,,,LARGE,1,F,240,240,2,,B
```

COMMENTS

The mount requests will appear as

```
IO37,MT00SL
```

The operator should of course mount an 800 bpi labeled scratch tape and key in

```
;RESUME,MT00SL
```

in response to each request.

Example 8

Assume that Example 7 resulted in reels 001276 and 001233 respectively being used. Copy the data set back to disc.

```
/JOB,RESTORE
/TFILE,22,I,,,S,,001276,LARGE
/IOUTIL
PFILE,8,,120,LARGE
COPYF,1,22,0,120,8,0,120
CFILE,8,,LARGE,1
/REW,22
/FINI
```

COMMENTS

The mount requests will appear as

```
     IO37,MT00ML
```

and

```
     IO37,MT00NL
```

respectively. The operator should mount 001276 and 001233 respectively and key in

```
     ;RESUME,MT00ML
```

and

```
     ;RESUME,MT00NL
```

respectively.

Example 9

Append the disc file SMALL on lun 8 (SS) to the output tape from Example 2. The file contains 40 word records blocked three to one.

```
     /JOB,APPEND
     /TFILE,26,0,,F4,S,,123456,SMALL,4,F,240,80,3,,B
     /IOUTIL
     PFILE,8,,120,SMALL
     COPYF,1,SMALL,0,120,26,0,120
     /REW,22
     /FINI
```

COMMENTS

At the console

```
     IO37,MT10ML
```

appears. The operator should mount reel 123456 and key in

```
     ;RESUME,MT10ML
```

The tape will be spaced past data set 3; the file will then be written to the tape; the tape will be rewound.