Tape Formats [2]

The tape subsystem supports the IBM compatible tape format and ER90 (D2 cassettes) tape format. The ER90 format is not available on systems that run the UNICOS/mk operating system or that have GigaRing support. This chapter describes and illustrates both formats and label fields.

2.1 IBM compatible tape format

This section briefly describes and illustrates the IBM tape format. Tape format is determined by the presence or absence of labels and the number of files on a tape volume or number of volumes for a tape file.

System labels and tape marks are accessible to a user process without privileges only through the use of the `tpmnt`(1) command.

In the following figures, the character b represents the beginning of the tape and the character * represents a tape mark (HDR2, EOV2, and EOF2 labels are optional for input). The UNICOS and UNICOS/mk operating systems always creates these labels for labeled tapes; other systems may not.

2.1.1 Nonlabeled tapes

Nonlabeled tapes are of two formats, determined by the number of tape marks that indicate end-of-volume.

2.1.1.1 Two tape mark tapes

Nonlabeled tapes with two tape marks, implemented by the `-l nl` option of the `tpmnt`(1) command, may consist of a single-volume file; a multivolume file; or multifile, multivolume file formats. Figure 3 illustrates these formats. For tapes with multiple files, a single tape mark separates files on the same volume. End-of-volume is reached when two consecutive tape marks are encountered and there is another tape to read.

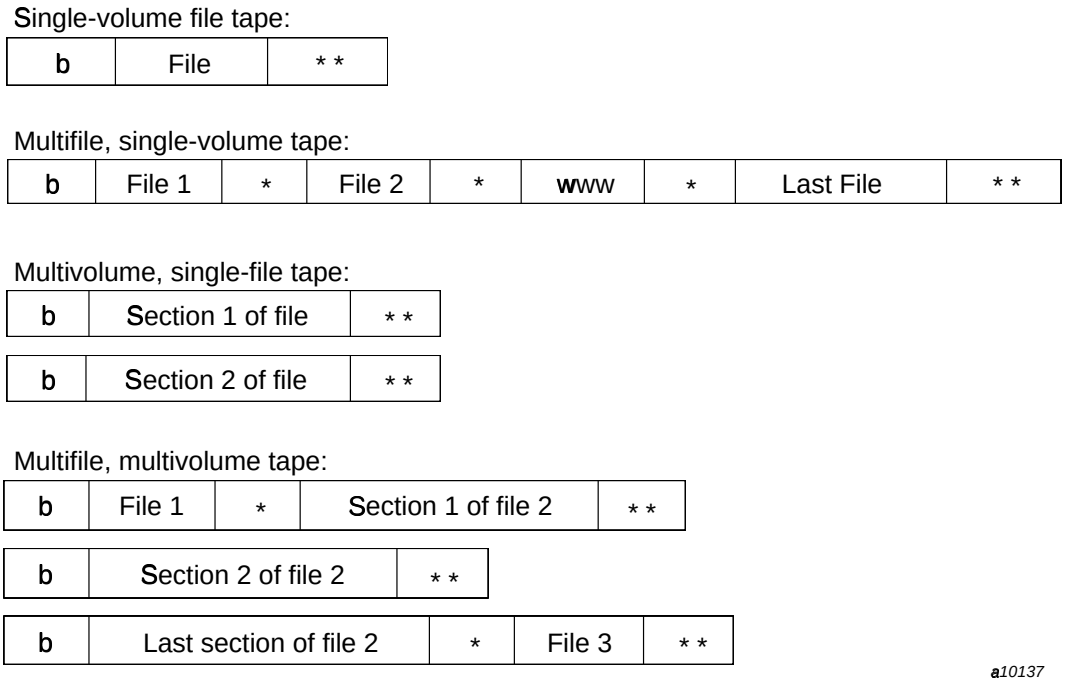

Figure 3. Nonlabeled, two tape mark formats

2.1.1.2 Single tape mark tapes

For nonlabeled, single tape mark format, implemented by the `-l st` option of the `tpmnt`(1)command, a single tape mark indicates end-of-volume. When using one tape mark tape as an input tape, the system reads only to the first tape mark encountered.

When using a single tape mark tape as an output tape, the system terminates the tape with three tape marks, allowing it to be read as a nonlabeled tape later on. Note that because the system processes only the data blocks and the first tape mark, you cannot have multifiles on a single tape mark tape. That is, you cannot use the `-l st` option with the `-q` option of the `tpmnt`(1) command.

Figure 4 illustrates nonlabeled, single tape mark formats.

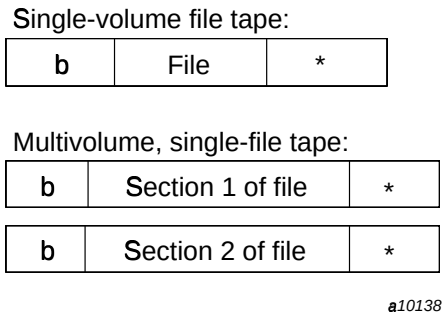

Figure 4. Nonlabeled, single tape mark formats

2.1.2 Labeled tapes

Labeled tapes are implemented by the `-l al` (ANSI standard labels) and the `-l sl` (IBM standard labels) options of the `tpmnt`(1) command. ANSI standard labels and IBM standard labels are similar, with the exception that in IBM standard labels the character fields are represented by EBCDIC characters while in ANSI standard labels the character fields use ASCII characters.

Labeled tapes have the following labels for the tape subsystem (see Section 2.2, page 16, for a description of these labels):

- Volume header label (VOL1)
- First file header (HDR1)
- First end-of-volume (EOV1)
- First end-of-file (EOF1)
- Second file header (HDR2)
- Second end-of-volume (EOV2)
- Second end-of-file (EOF2)

Figure 5 illustrates labeled tape formats.

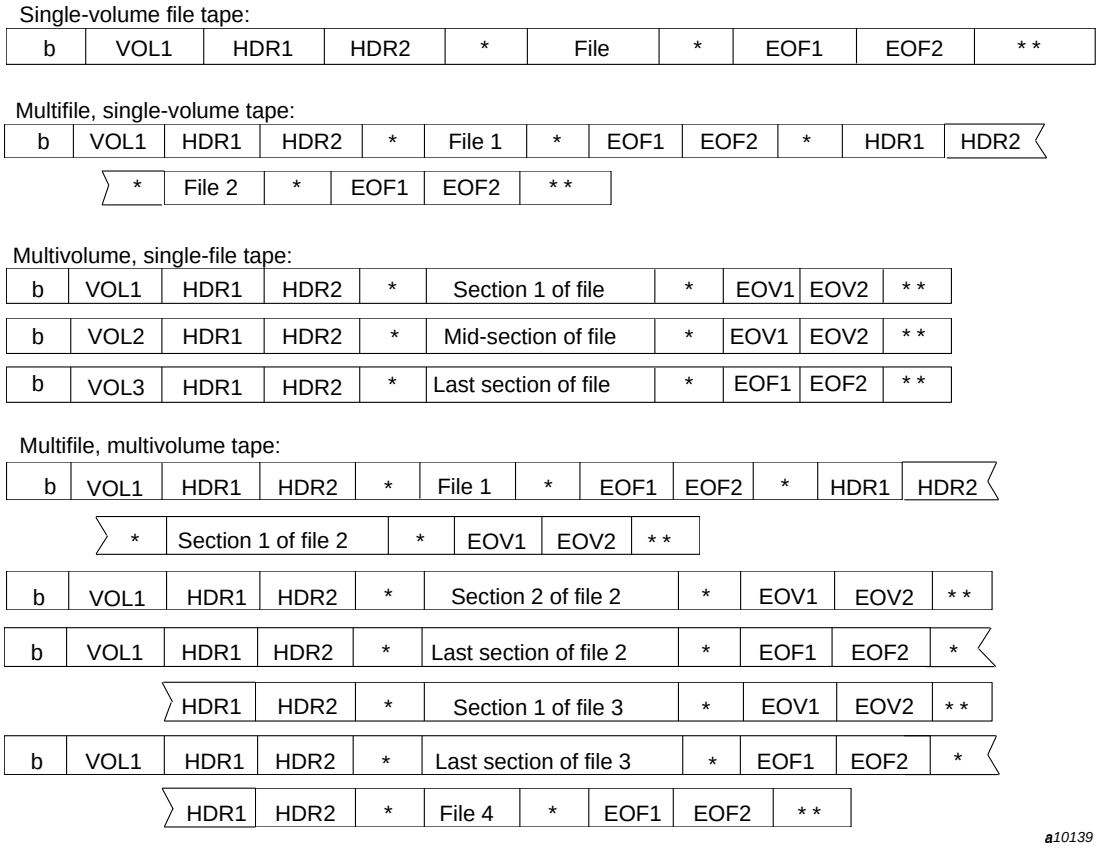

Figure 5. Labeled tape formats

2.1.3 IBM compatible tape format summary

The formats described previously are illustrated in Figure 6 through Figure 9 grouped by number of files and number of volumes. Figure 6 shows a single-volume file; Figure 7 shows a multifile, single-volume tape; Figure 8 shows a multivolume, single-file tape; and Figure 9 shows a multifile, multivolume tape. For each format type, the figures show both labeled (ANSI or IBM) and nonlabeled tapes.

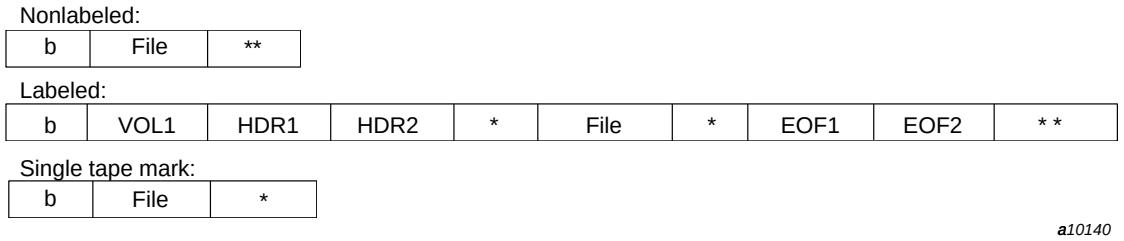

Figure 6. Single-volume file

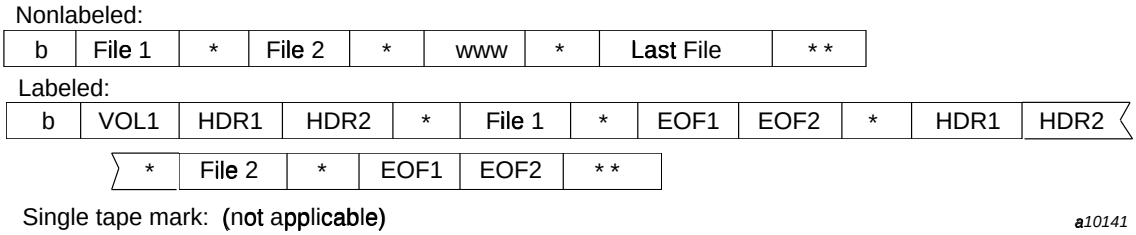

Figure 7. Multifile, single-volume tape

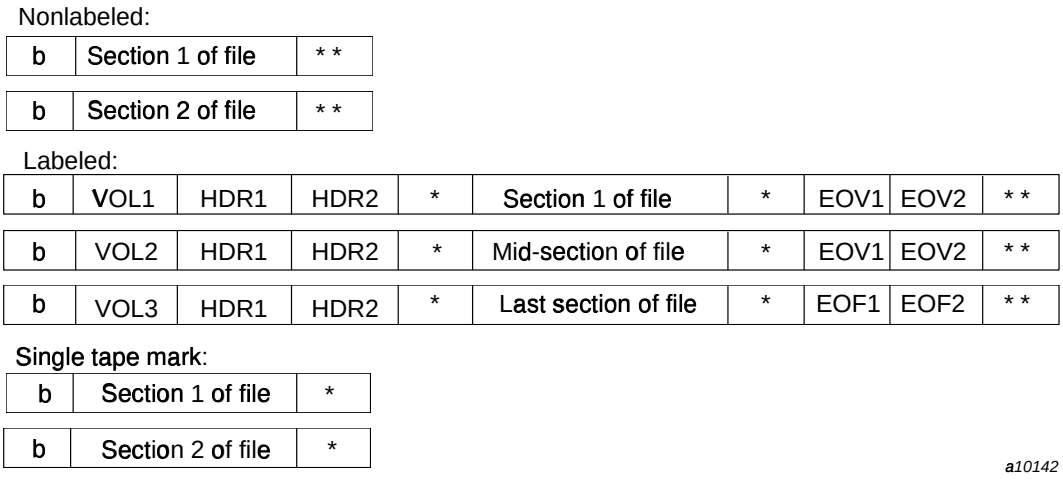

Figure 8. Multivolume, single-file tape

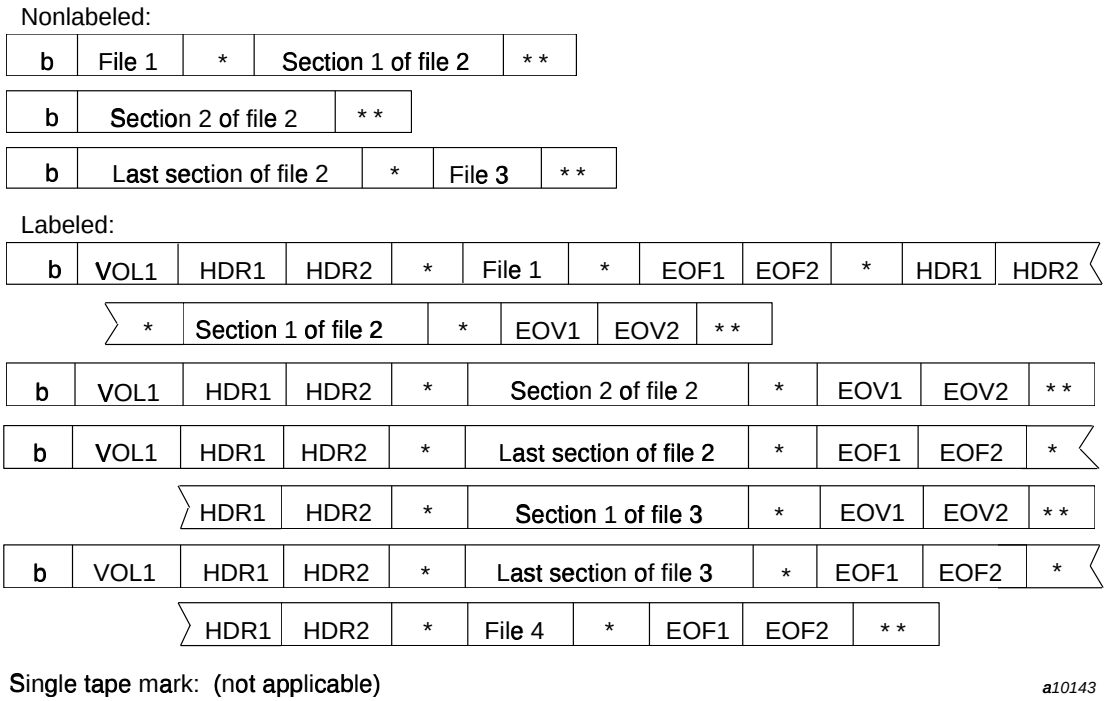

Figure 9. Multifile, multivolume tape

2.2 Tape label fields

This section describes the various tape label fields for ANSI standard and IBM standard labels. Specifically, it describes the fields in which label types are supported. These are checked by the system when reading or writing a tape and those that are filled in with parameter values when you use the `tpmnt`(1) command to create a labeled tape. The following tape labels are described for the tape subsystem:

- Volume header label (VOL1)
- First file header (HDR1)
- First end-of-volume (EOV1)
- First end-of-file (EOF1)
- Second file header (HDR2)

- Second end-of-volume (EOV2)
- Second end-of-file (EOF2)

In IBM standard-label character fields, the characters are represented by EBCDIC characters. ANSI standard labels use ASCII characters.

2.2.1 VOL1 label

The VOL1 label is the first block on a labeled tape. Table 1 describes the fields for an ANSI standard label. Figure 10 shows the format of the VOL1 label.

Table 1. VOL1 label values

Field	Starting byte	Length in bytes	Contents	Description
`label id`	1	4	VOL1	VOL1 label; required system-supplied character string.
`volume id`	5	6	*vi*	Volume identifier of the tape; it is specified with the `-v` option or contained in the file specified with the `-V` option of the `tpmnt`(1) command. It is checked on all labeled tapes and contains up to 6 alphanumeric characters.
`owner id`	38	14	*owner_id*	User ID of the tape owner.
`standard level`	80	1	*label standard version*	ANSI standard version number for label and data formats. For Cray Research systems, the version number is 4.

The fields of the ANSI standard VOL1 label are the same as the IBM standard VOL1 label, with the following exceptions:

- The `owner id` field of the IBM standard VOL1 label starts at byte 42 and has a length of 10 bytes.
- The standard level field is not used in the IBM standard VOL1 label.

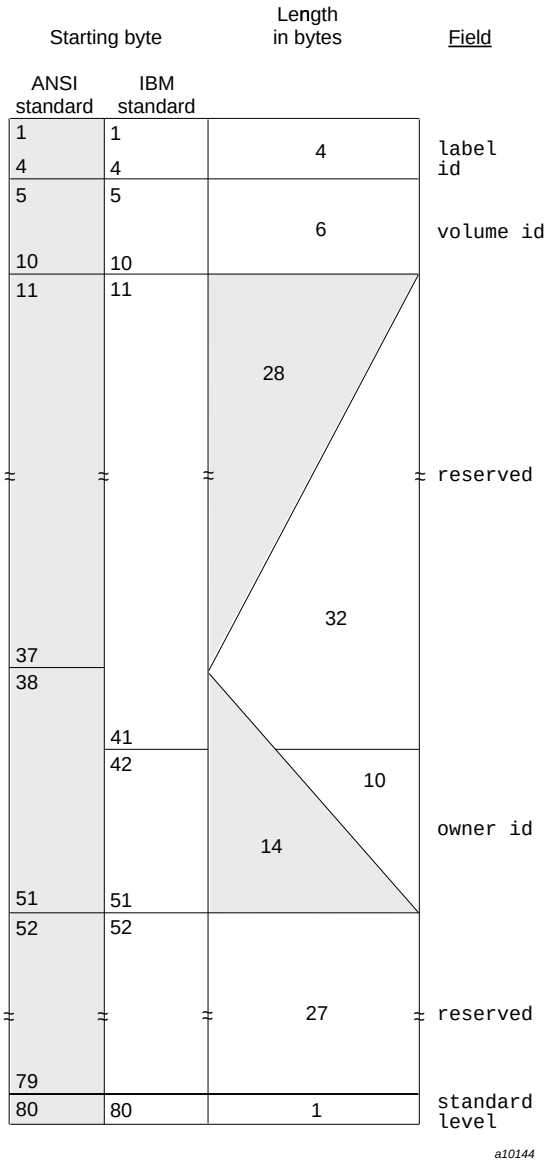

Figure 10. VOL1 label

2.2.2 HDR1, EOV1, and EOF1 labels

The HDR1 label is located before each file or section of a file on a tape volume. If a file is not completed on a tape volume and extends to the following tape volume, the data in the file is followed by an EOV1 label. If a file or file section is completed on a tape volume, the data in the file is followed by an EOF1 label.

The fields of the HDR1, EOV1, and EOF1 labels are the same in both the ANSI standard label format and the IBM standard label format. Table 2 describes the specified fields. Figure 11, page 21, shows the format of the HDR1/EOV1/EOF1 labels.

Table 2. HDR1/EOV1/EOF1 labels

Field	Starting byte	Length in bytes	Contents	Description
`label id`	1	4	HDR1 EOV1 EOF1	Label type; required system-supplied character string.
`file id`	5	17	*file_id*	File identifier; 1 through 17 alphanumeric character field specified by the `-f` option of the `tpmnt`(1) command. If the `-f` option is not specified, the file identifier is taken from the path name of the `-p` or `-P` option of `tpmnt`. The level of checking on `file id` is installation specified.
`sequence`	28	4	*number*	Order of this volume in a multivolume set; it is specified by a decimal number (1 through 9999) on the `-O` option of `tpmnt`.
`file sequence`	32	4	*number*	File order within a multifile tape; it is specified by a decimal number (1 through 9999) on the `-q` option of `tpmnt`. The system uses the specified value to position the tape volume to the proper file.
`creation date`	42	6	*cyyddd*	Creation date (pseudo-Julian format) of a new tape; *c* = century (blank = 19, 0 = 20, 1 = 21...), *yy* = year (00-99), and *ddd* = day (001-366).

Field	Starting byte	Length in bytes	Contents	Description
`expiration date`	48	6	*cyyddd*	Expiration date (pseudo-Julian format) at which time a tape may be scratched or overwritten. Normally, it is specified in the *cyyddd* format by using the `-x` option of `tpmnt`. Otherwise, you can specify the number of days on the `-t` option by using `tpmnt`(1). The specified number is added to the `creation date`, thus creating the `expiration date`.
`block count`	55	6	*number*	Number of data blocks in the preceding file section or file on the current tape volume for EOV1 and EOF1 labels. The block count in the HDR1 label contains a value of 000000. In EOV and EOF labels for standard labels (`sl`), if the block count is greater than 999,999, the `block count` field will represent the block count as mod 1,000,000. The overflow (`block count` / 1000000) will be stored in bytes 76 through 80. This is the `extended block count` field. For ANSI standard labels (`al`), if the block count is greater than 999,999, the `block count` field will represent the block count as mod 1,000,000.
`extended block count`	76	5	*number*	For standard labels (`sl`), if the block count is greater than 999,999, the `block count` field will represent the block count as mod 1,000,000. The `extended block count` field will contain the overflow (`block count` / 1000000).

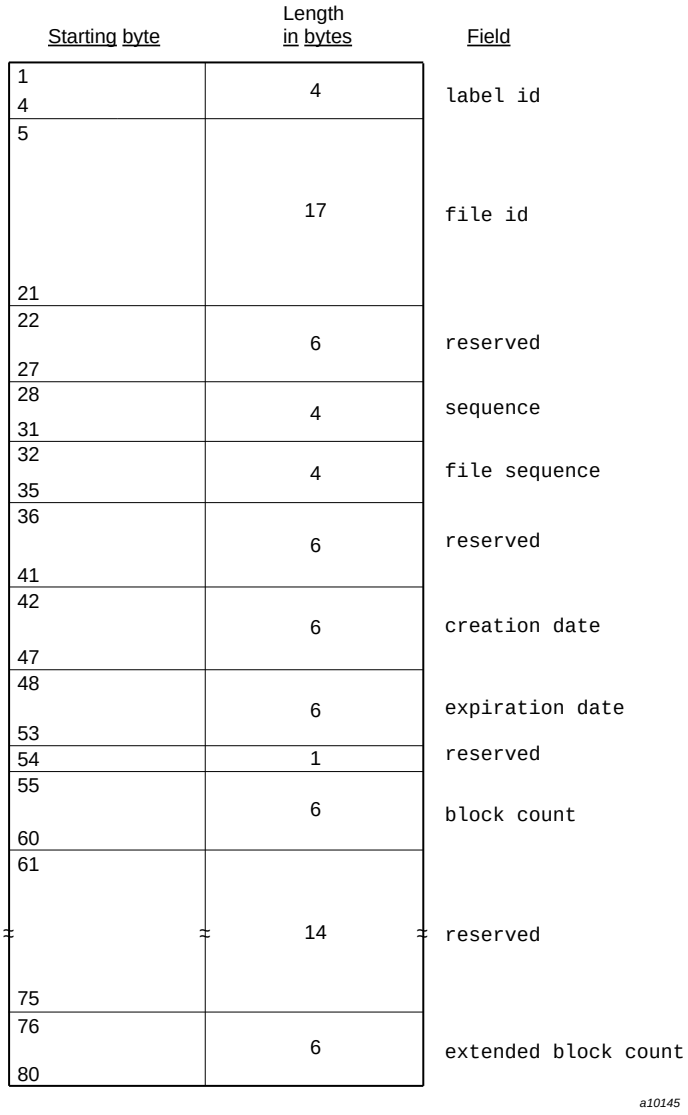

Figure 11. HDR1/EOV1/EOF1 labels

2.2.3 HDR2, EOV2, and EOF2 labels

An HDR2 label immediately follows an HDR1 label, and it is followed by a tape mark. An EOV2 label immediately follows an EOV1 label, and it is

followed by two tape marks. An EOF2 label immediately follows an EOF1 label, and if more files follow this file, it is followed by one tape mark. If the EOF2 label is the last file on the tape volume, it is followed by two tape marks.

ANSI standard does not specify a format for these labels, except for the first 4 bytes. IBM standard labels use the HDR2, EOV2, and EOF2 labels to store additional information concerning the file they bracket. The operating system automatically writes these labels when you use the `-l sl` or `-l al` options of `tpmnt`(1). These labels follow the IBM standard format. Table 3 describes the specified fields. Figure 12 shows the format of the HDR2/EOV2/EOF2 labels.

Table 3. HDR2/EOV2/EOF2 labels

Field	Starting byte	Length in bytes	Contents	Description
`label id`	1	4	HDR1 EOV1 EOF1	Label type; required system-supplied character string.
`record format`	5	1	*format*	Record format; 1-character filed specified by the `-F` option of `tpmnt`(1).
`block length`	6	5	*number*	Maximum block length, in bytes, for the associated file; specified by a decimal number (1 through 99999) on the `--b` option of `tpmnt`. If the block length is greater than 100000, the block length field will represent the block length as mod 100000.
`record length`	11	5	*number*	Record length, in bytes; specified by the `--L` option of `tpmnt`.
`density`	16	1	*number*	Tape density; specified by the `-d` option of `tpmnt`. The operating system supports 1600 bpi (this field contains the value 3) and 6250 bpi (this field contains the value 4).
`security level`	55	3	*number*	Security level.
`compartments`	59	17	*numbers*	Security compartments.

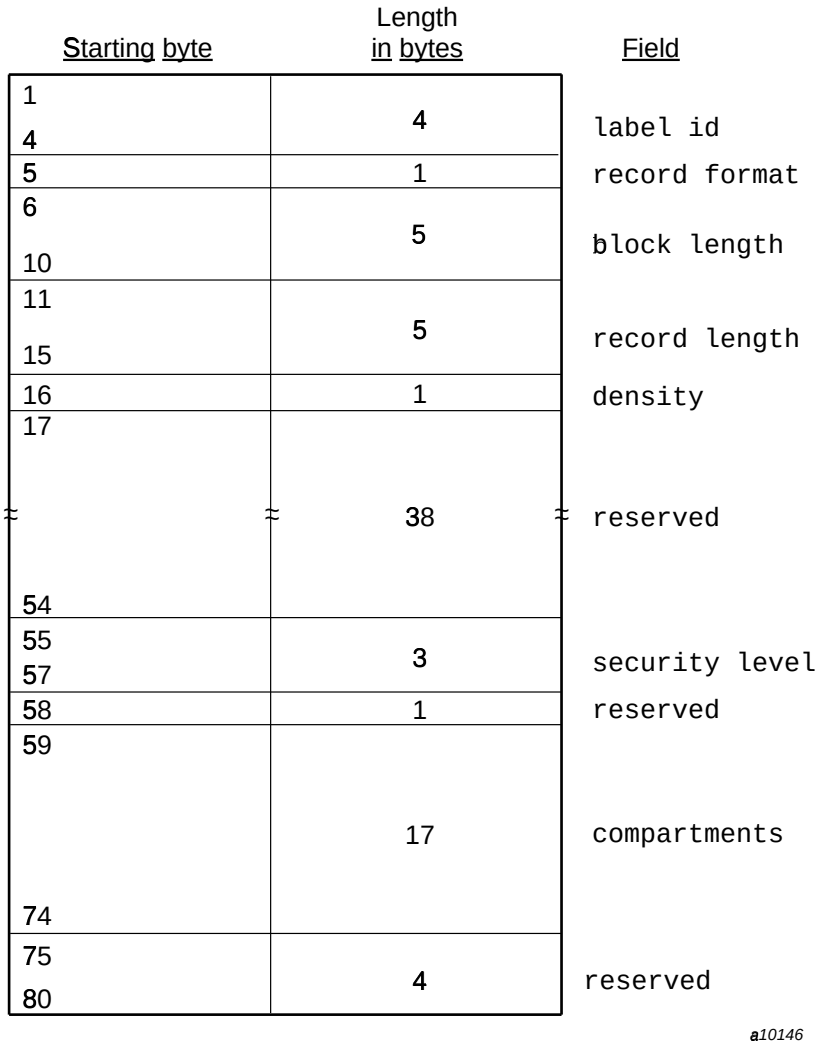

Figure 12. HDR2/EOV2/EOF2 labels

2.3 ER90 volumes

The ER90 supports D2 cassettes with 19-mm tapes in three cassette sizes: 25 Gbyte, 75 Gbyte, and 165 Gbyte.

Note: The ER90 format is not available on systems that run the UNICOS/mk operating system or that have GigaRing support.

You must format a volume before it can be used. To format a volume, you must create single or multiple partitions on the cassette, record a volume identifier, and, if requested, create system zones. A volume can be preformatted, or it can be formatted during write operations.

Partitions are logical volumes within a physical volume. A partition can span the length of the tape, or multiple partitions can be created. The tape subsystem treats partitions as individual volumes; they are accessed individually, and tape operations to one partition do not affect other partitions on the volume. Multiple partitions cannot be created during write operations.

The ER90 device records a format identifier (ID) as part of a volume format operation. The format identifier is an alphanumeric string consisting of up to 6 ASCII characters that uniquely identifies the cassette after tapes are mounted. The format identifier is recorded throughout the volume after each system zone and at the beginning-of-tape (BOT) and end-of-tape (EOT) markers.

You can format a volume with or without system zones. *System zones* are data-free areas on the tape that can be used to load and unload the cassette. With system zones, a cassette does not have to be positioned at the BOT or the EOT to be unloaded.

If a volume is formatted to have the default number of system zones, a tape unload takes approximately 16 seconds for small cassettes, 21 seconds for medium cassettes, and 24 seconds for large cassettes. Volumes that are formatted without system zones can take up to 185 seconds to be unloaded. The disadvantage of formatting with system zones is that if a tape is created during write operations, the ER90 device must suspend I/O operations to create the system zone. It takes approximately 18 seconds to create a system zone for small cassettes, 31.3 seconds for medium cassettes, and 55.8 seconds for large cassettes. It takes 3.2 seconds to skip over a system zone and continue writing for small cassettes, 3.6 seconds for medium cassettes, and 5.3 seconds for large cassettes.

By default, the tape daemon formats a blank tape as a single partition volume with system zones. The format ID specified on the tape mount is recorded on the volume, which is formatted during write operations.

The tape administrator can use the `tpformat`(8) command to preformat ER90 volumes. This command reserves an ER90 device, mounts the volume, issues the format request to the ER90 device, and then, after the format is completed, releases the reserved resource.