INTERNATIONAL STANDARD

ISO/IEC 13346-3

Second edition
1999-06-01

Information technology — Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange —

Part 3:
Volume structure

Technologies de l'information — Structure de volume et de fichier de moyens d'écriture unique et de réécriture utilisant un enregistrement non séquentiel pour l'échange d'information —

Partie 3: Structure de volume

Reference number
ISO/IEC 13346-3:1999(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialised system for worldwide standardisation. National Bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, governmental and non-governmental, in liaison with ISO and IEC, also take part in this work.

In the field of information technology, ISO and IEC have established a Joint Technical Committee, ISO/IEC JTC 1. Draft International Standards adopted by the Joint Technical Committee are circulated to National Bodies for voting. Publication of an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 13346-3 was prepared by ECMA, (as Standard ECMA-167) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by National Bodies of ISO and IEC.

This second edition cancels and replaces the first edition (ISO/IEC 13346-3:1995), which has been technically revised.

ISO/IEC 13346 consists of the following parts, under the general title *Information technology — Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange*:

— *Part 1: General*

— *Part 2: Volume and boot block recognition*

— *Part 3: Volume structure*

— *Part 4: File structure*

— *Part 5: Record structure*

Annex A of this part of ISO/IEC 13346 is for information only.

Introduction

ISO/IEC 13346 is a volume and file structure standard for interchanging files and as such, it is a peer to existing volume and file structure standards such as ISO 9293 and ISO 9660. It is rather different from those standards in at least two important ways. Firstly, it offers much more functionality, mainly because of user needs for increased character set support and for more powerful file system features. Secondly, it acknowledges the separate concerns of booting, volume structure and file system structure. Rather than bundling these different functions together, ISO/IEC 13346 carefully segregates these functions into separate parts and describes in detail how those parts fit together. It is expected that future volume and file structure standards will fit into this framework, rather than building other distinct and incompatible formats.

ISO/IEC 13346 is published in five Parts. Part 1 - general - specifies references, definitions, notations and basic structures used in the other four Parts. Part 2 - volume and boot block recognition - specifies formats and system requirements for recognising the volume structures on a medium and booting from a medium. Part 3 - volume structure - specifies how to record various volume-related entities such as volumes, volume sets and logical volumes. Part 4 - file structure - specifies how to record and interpret files, both file data and file attributes, and file hierarchies within logical volumes. Part 5 - record structure - specifies how to record and interpret file data encoded as records.

Information technology — Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange —

Part 3:
Volume structure

1 Scope

ISO/IEC 13346 specifies a format and associated system requirements for volume and boot block recognition, volume structure, file structure and record structure for the interchange of information on media between users of information processing systems.

The media shall be recorded as if the recording of sectors may be done in any order.

Note 1 - The medium is not restricted to being of only one type; the type of medium may be either write once, or read only, or rewritable, or a combination of these types.

ISO/IEC 13346 consists of the following five Parts:

Part 1: General
Part 2: Volume and Boot Block Recognition
Part 3: Volume Structure
Part 4: File Structure
Part 5: Record Structure
Annex A - ICB Strategies, is part of ISO/IEC 13346-4.

This part of ISO/IEC 13346 specifies a format and associated system requirements for volume structure by specifying:

— the attributes of a volume and the descriptors recorded on it;

— the relationship among volumes of a volume set;

— the attributes of a partition of a volume;

— the attributes of a logical volume and the descriptors recorded on it;

— levels of medium interchange;

— requirements for the processes which are provided within information processing systems, to enable information to be interchanged between different systems; for this purpose, it specifies the functions to be provided within systems which are intended to originate or receive media which conform to this part of ISO/IEC 13346.

2 Parts references

The first digit of a reference within ISO/IEC 13346 identifies the Part, e.g. 2/5 refers to clause 5 in ISO/IEC 13346-2, and figure 4/3 refers to figure 3 in ISO/IEC 13346-4.

3 Part interface

This clause specifies the interface of this part of ISO/IEC 13346 to other standards or Parts.

3.1 Input

This part of ISO/IEC 13346 requires the specification of the following by another standard or Part.

— A standard for recording (see 1/5.10).

— The size of a logical sector (see 3/8.1.2) of a volume.

— If the volume is recorded according to ISO/IEC 13346-2, a volume recognition sequence specified by ISO/IEC 13346-2 shall contain the descriptor described in 3/9.1 recorded at least once.

— If the volume is recorded according to ISO/IEC 13346-2, the volume recognition space (see 2/8.2) shall be the entire volume.

— If the volume is recorded according to ISO/IEC 13346-2, the initial sector in the volume (see2/3.1) shall be the first sector of the volume.

— Information to be recorded in the Partition Contents Use field of a Partition Descriptor (see 3/10.5.6).

— Information to be recorded in the Logical Volume Contents Use field of a Logical Volume Descriptor (see 3/10.6.7).

3.2 Output

This part of ISO/IEC 13346 specifies the following which may be used by other standards or Parts.

— Volume sets of one or more volumes (see 3/8.6).

— A volume space for a volume (see 3/8.2).

— Logical sectors of a fixed size for a volume (see 3/8.1.2).

— Partitions (see 3/8.7).

— Logical volumes composed of partitions (see 3/8.8).

— Numeric identification of the partitions within a logical volume (see 3/8.8).

— Logical blocks of a fixed size for a logical volume.

— The logical block size for a logical volume.

— Attributes of a volume.

— Attributes of a logical volume.

— Attributes of a partition.

— An indication that a volume may have been recorded to this part of ISO/IEC 13346 (see 3/9.1).

4 Conformance

4.1 Conformance of a medium

A medium shall be in conformance with ISO/IEC 13346 when it conforms to a standard for recording (see 1/5.10) and information recorded on sectors of the medium conform to the specifications of ISO/IEC 13346-1 and one or more of ISO/IEC 13346-2, -3, -4 and -5. A statement of conformance shall identify the sectors of the medium on which information is recorded according to the specifications of ISO/IEC 13346, and the Parts and the levels of medium interchange (see 1/10, 3/11, and 4/15) to which the contents of those sectors of the medium conform.

4.2 Conformance of an information processing system

An information processing system shall be in conformance with ISO/IEC 13346 if it meets the requirements specified in ISO/IEC 13346-1 and one or more of ISO/IEC 13346-2, -3, -4 and -5 either for an originating system (see 2/13, 3/13, 4/17 and 5/11) or for a receiving system (see 2/13, 3/14, 4/18 and 5/12) or for both types of system. A statement of conformance shall identify the Parts, and the levels of the requirements for each of those Parts, which can be met by the system.

5 Definitions

In addition to the definitions of ISO/IEC 13346-1 (see 1/5), the following definitions apply for this part of ISO/IEC 13346.

5.1 Anchor point

One of a specified set of logical sector numbers at which descriptors, that identify an extent of a Volume Descriptor Sequence, may be recorded.

5.2 Cyclic Redundancy Check (CRC)

A method for computing a signature of a sequence of bytes.

5.3 Extent

A set of logical sectors whose logical sector numbers (see 3/8.1.2.1) form a continuous ascending sequence. The address, or location, of an extent is the first logical sector number in that sequence.

5.4 Logical block

The unit of allocation of a logical volume.

5.5 Logical sector

The unit of allocation of a volume.

5.6 Logical volume

A nonempty set of partitions.

5.7 Partition

An extent of logical sectors within a volume.

6 Notation

The notation of ISO/IEC 13346-1 (see 1/6) applies to this part of ISO/IEC 13346.

7 Basic types

In addition to the basic types of ISO/IEC 13346-1 (see 1/7), the following basic types apply for this part of ISO/IEC 13346.

7.1 Extent Descriptor

An Extent Descriptor, hereafter designated as `extent_ad`, shall be recorded in the format shown in figure 3/1.

RBP	Length	Name	Contents
0	4	Extent Length	`Uint32` (1/7.1.5)
4	4	Extent Location	`Uint32` (1/7.1.5)

Figure 1 — `extent_ad` format

7.1.1 Extent Length (RBP 0)

This field shall indicate the length of the extent, in bytes, identified by the Extent Location field. The length shall be less than 2^{30}. Unless otherwise specified, the length shall be an integral multiple of the logical sector size.

7.1.2 Extent Location (RBP 4)

This field shall specify the location of the extent, as a logical sector number. If the extent's length is 0, no extent is specified and this field shall contain 0.

7.2 Descriptor tag

Certain descriptors specified in this part of ISO/IEC 13346 have a 16 byte structure, or `tag`, recorded at the start of the descriptor. The `tag` shall be recorded with the format shown in figure 3/2.

Note 2 - There are two main motivations for using a generic tag structure. The first is that most descriptors need to handle common issues of CRCs and format versions. The second motivation is to support recovery after the medium has been damaged or corrupted in some (unspecified) way. With the tag described here, structures are self identifying and can be verified with very little context.

RBP	Length	Name	Contents
0	2	Tag Identifier	`Uint16` (1/7.1.3)
2	2	Descriptor Version	`Uint16` (1/7.1.3)
4	1	Tag Checksum	`Uint8` (1/7.1.1)
5	1	Reserved	#00 byte
6	2	Tag Serial Number	`Uint16` (1/7.1.3)
8	2	Descriptor CRC	`Uint16` (1/7.1.3)
10	2	Descriptor CRC Length	`Uint16` (1/7.1.3)
12	4	Tag Location	`Uint32` (1/7.1.5)

Figure 2 — `tag` format

7.2.1 Tag Identifier (RBP 0)

This field shall specify an identification of the descriptor type. Type 0 shall specify that the format of this descriptor is not specified by this part of ISO/IEC 13346. Types 1-7 and 9 are specified as shown in figure 3/3. Type 8 is specified identically in this part of ISO/IEC 13346 and ISO/IEC 13346-4. Types 256-265 are specified in ISO/IEC 13346-4. All other types are reserved for future standardisation. The descriptor types specified by this part of ISO/IEC 13346 are shown in figure 3/3.

Type	Interpretation
1	Primary Volume Descriptor (3/10.1)
2	Anchor Volume Descriptor Pointer (3/10.2)
3	Volume Descriptor Pointer (3/10.3)
4	Implementation Use Volume Descriptor (3/10.4)
5	Partition Descriptor (3/10.5)
6	Logical Volume Descriptor (3/10.6)
7	Unallocated Space Descriptor (3/10.8)
8	Terminating Descriptor (3/10.9 and 4/14.2)
9	Logical Volume Integrity Descriptor (3/10.10)

Figure 3 — Descriptor interpretation

7.2.2 Descriptor Version (RBP 2)

This field shall specify the version of this descriptor. This value shall be 2 or 3. The value 3 shall indicate the structure of this part of ISO/IEC 13346. The value 2 shall indicate the structure of this part of ISO/IEC 13346. See 3/13.1 and 3/14.1 for requirements.

Note 3 - Structures with version 2 descriptors may be on the medium due to changing the medium from NSR02 to NSR03 without rewriting all descriptors as version 3. Originating systems shall record a 3 in this field; receiving systems shall allow a 2 or 3 (see 3/13.1 and 3/14.1).

7.2.3 Tag Checksum (RBP 4)

This field shall specify the sum modulo 256 of bytes 0-3 and 5-15 of the `tag`.

7.2.4 Reserved (RBP 5)

This field shall be reserved for future standardisation and shall be set to 0.

7.2.5 Tag Serial Number (RBP 6)

This field shall specify an identification of a set of descriptors. If the field contains 0, then no such identification is specified.

Note 4 - This field can be used to distinguish between groups of descriptors. For example, when reusing rewritable media, an implementation might choose a different serial number from the previous use when initialising a volume. Thus, a disaster recovery mechanism can avoid recovering prior and unintended data. The only alternative to this scheme would be to force volume initialisation to clear the volume.

7.2.6 Descriptor CRC (RBP 8)

This field shall specify the CRC of the bytes of the descriptor starting at the first byte after the descriptor tag. The number of bytes shall be specified by the Descriptor CRC Length field. The CRC shall be 16 bits long and be generated by the CRC-ITU-T polynomial (see ITU-T V.41):

$$x^{16} + x^{12} + x^5 + 1$$

Note 5 - As an example, the CRC of the three bytes #70 #6A #77 is #3299. Implementations can avoid calculating the CRC by setting the Descriptor CRC Length to 0, as then the Descriptor CRC shall be 0.

7.2.7 Descriptor CRC Length (RBP 10)

This field specifies how many bytes were used in calculating the Descriptor CRC.

7.2.8 Tag Location (RBP 12)

This field shall specify the number of the logical sector containing the first byte of the descriptor.

Note 6 - The location of the tag may appear to be redundant but its primary purpose is to make it extremely likely that if the first 16 bytes of a logical sector or logical block is a consistent descriptor tag, then it is a descriptor tag.

8 Volume structure

8.1 Arrangement of information on a volume

8.1.1 Sector numbers

Each sector of a volume shall be identified by a unique sector number. Sector numbers shall be consecutive integers assigned in an ascending sequence, in the order of ascending physical address of the volume as specified in the relevant standard for recording (see 1/5.10). Sector number 0 shall be assigned to the sector having the lowest physical address of the volume.

8.1.2 Logical sector

The sectors of a volume shall be organised into logical sectors of equal length. The length of a logical sector shall be referred to as the logical sector size and shall be an integral multiple of 512 bytes. The logical sector size shall be not less than the size of the smallest sector of the volume. Each logical sector shall begin in a different sector, starting with the sector having the next higher sector number than that of the last sector constituting the previous, if any, logical sector of the volume. The first byte of a logical sector shall be the first byte of the sector in which it begins, and if the size of this sector is smaller than the logical sector size, then the logical sector shall comprise a sequence of constituent sectors with consecutive ascending sector numbers.

8.1.2.1 Logical sector numbers

Each logical sector of a volume shall be identified by a unique logical sector number. Logical sector numbers shall be consecutive integers assigned in ascending sequence, in the order of ascending sector numbers of the volume. Logical sector number 0 shall be assigned to the logical sector beginning in sector number 0. The largest logical sector number of a volume shall be greater than 256.

8.1.2.2 Recording of logical sectors

Any unrecorded constituent sector of a logical sector shall be interpreted as containing all #00 bytes. Within the sector containing the last byte of a logical sector, the interpretation of any bytes after that last byte is not specified by this part of ISO/IEC 13346.

A logical sector is unrecorded if the standard for recording allows detection that a sector has been unrecorded and all of the logical sector's constituent sectors are unrecorded. A logical sector should either be completely recorded or unrecorded.

8.2 Volume space

The information on a volume shall be recorded in the set of all logical sectors in a volume. This set shall be referred to as the volume space of the volume. The bytes in the volume space shall be numbered with consecutive integers assigned in ascending sequence starting with 0. Let *s* be the number of bytes in a logical sector; then byte *b* of the volume space is byte *rem(b,s)* of logical sector *ip(b/s)*.

8.3 Volume descriptors

Characteristics of the volume shall be specified by volume descriptors recorded in Volume Descriptor Sequences as described in 3/8.4.2.

A volume descriptor shall be one of the following types:

— Primary Volume Descriptor (see 3/10.1)

— Implementation Use Volume Descriptor (see 3/10.4)

— Partition Descriptor (see 3/10.5)

— Logical Volume Descriptor (see 3/10.6)

— Unallocated Space Descriptor (see 3/10.8)

8.4 Volume Descriptor Sequence

8.4.1 Contents of a Volume Descriptor Sequence

A Volume Descriptor Sequence shall contain one or more Primary Volume Descriptors. A Primary Volume Descriptor shall identify the volume and the volume set to which it belongs, the sequence number of the volume within the volume set, attributes of the volume, and the character sets used in recording the contents of certain fields within the Primary Volume Descriptor. Each Primary Volume Descriptor shall have an assigned Primary Volume Descriptor Number. Only one prevailing Primary Volume Descriptor (see 3/8.4.3) of a Volume Descriptor Sequence shall have a Primary Volume Descriptor Number of 0.

A Volume Descriptor Sequence shall contain zero or more Implementation Use Volume Descriptors. An Implementation Use Volume Descriptor shall identify an implementation and contain information for that implementation's use.

A Volume Descriptor Sequence shall contain zero or more Partition Descriptors. A Partition Descriptor shall specify a partition, attributes of the partition and an identification of the partition, referred to as the partition number.

A Volume Descriptor Sequence shall contain zero or more Logical Volume Descriptors. A Logical Volume Descriptor shall specify an identification of the logical volume, the logical block size of the logical volume, identification of the partitions comprising the logical volume and attributes of the logical volume.

A Volume Descriptor Sequence shall contain zero or more Unallocated Space Descriptors. An Unallocated Space Descriptor shall identify volume space available for allocating to partitions or for recording the Volume Descriptor Sequences of the volume.

Each volume descriptor shall have an assigned Volume Descriptor Sequence Number. All volume descriptors with identical Volume Descriptor Sequence Numbers shall have identical contents.

Note 7 - Typically, an originating system will chose a new Volume Descriptor Sequence Number by adding 1 to the largest such number seen when scanning the Volume Descriptor Sequence.

8.4.2 Recording of the Volume Descriptor Sequence

A Volume Descriptor Sequence shall be recorded as a sequence of extents of logical sectors in the volume space. Any trailing sectors (see figure 3/4) shall be available for the recording of descriptors.

An extent of a Volume Descriptor Sequence shall be recorded according to the schema shown in figure 3/4.

```
[Volume Descriptor Sequence extent]{
          <volume descriptor>0+
     [Terminator]{
          <Volume Descriptor Pointer>
        | <Terminating Descriptor>
        | <unrecorded logical sector>
     } <trailing logical sector>0+
}
```

Figure 4 — Volume Descriptor Sequence schema

An extent of a Volume Descriptor Sequence shall be identified by an Anchor Volume Descriptor Pointer (see 3/10.2) recorded at two or more anchor points (see 3/8.4.2.1). Each, if any, subsequent extent in the Volume Descriptor Sequence shall be identified by a Volume Descriptor Pointer recorded in the previous extent of the sequence.

An Anchor Volume Descriptor Pointer shall identify the Main Volume Descriptor Sequence and may identify a Reserve Volume Descriptor Sequence (see 3/8.4.2.2). All Volume Descriptor Sequences specified by Anchor Volume Descriptor Pointers shall be equivalent (see 3/8.4.2.3).

8.4.2.1 Anchor points

Let k be $ip(n/59)$, where n is the largest logical sector number in the volume space. Anchor points shall be at two or more of the following logical sector numbers: 256, n–256, n and all the nonzero integral multiples of k not greater than n.

Note 8 - The value 59 was chosen as a number near 64 that was unlikely to be periodic with respect to the geometry of the underlying medium.

8.4.2.2 Reserve Volume Descriptor Sequence Set

A Reserve Volume Descriptor Sequence may be recorded on a volume. A Reserve Volume Descriptor Sequence, if any, shall be identified by an Anchor Volume Descriptor Pointer. If any Anchor Volume Descriptor Pointer of the volume identifies a Reserve Volume Descriptor Sequence, then all Anchor Volume Descriptor Pointers of the volume shall identify a Reserve Volume Descriptor Sequence. If the Reserve Volume Descriptor Sequence is identified, it shall specify a Volume Descriptor Sequence equivalent to the Main Volume Descriptor Sequence (see 3/8.4.2.3). There shall be no logical sector which belongs to both an extent of the Main Volume Descriptor Sequence and an extent of the Reserve Volume Descriptor Sequence.

8.4.2.3 Equivalent Volume Descriptor Sequences

The equivalence of two Volume Descriptor Sequences shall be determined by calculating a canonical form for each of the Volume Descriptor Sequences, and if both the canonical forms are identical, then the two Volume Descriptor Sequences specify equivalent sets of volume descriptors. The canonical form of a Volume Descriptor Sequence shall be constructed by performing the following steps in sequence:

— discard any Volume Descriptor Pointers

— discard all but one descriptor for each Volume Descriptor Sequence Number

— set the Tag Checksum, Descriptor CRC, Descriptor CRC Length, and Tag Location fields of the Descriptor Tag field in each descriptor to 0

— sort the remaining descriptors as byte sequences

— catenate the descriptors in sorted order

8.4.3 Prevailing descriptors

Within each of the following classes of descriptors:

— Primary Volume Descriptors, each of which has the same contents of the corresponding Volume Identifier, Volume Set Identifier and Descriptor Character Set fields

— Partition Descriptors with identical Partition Numbers

— Logical Volume Descriptors, each of which has the same contents of the corresponding Logical Volume Identifier and Descriptor Character Set fields

— Unallocated Space Descriptors

the one with the highest Volume Descriptor Sequence Number shall be used. This instance shall be referred to as the prevailing instance.

8.4.4 Recording of descriptors

All the descriptors in this part of ISO/IEC 13346 whose format is specified with Byte Positions (BP) shall be recorded so that the first byte of the descriptor coincides with the first byte of a logical sector.

The descriptors in this part of ISO/IEC 13346 whose format is specified with Relative Byte Positions (RBP) have no restrictions on where they may be recorded within a logical sector, except that their location within a descriptor shall be specified in the description of the applicable descriptor.

When the descriptors described in this part of ISO/IEC 13346 are recorded in a logical sector, all space, if any, after the end of the last descriptor up to the end of the logical sector is reserved for future standardisation and shall be recorded as all #00 bytes.

Note 9 - Most of the descriptors specified in this part of ISO/IEC 13346 have a length of 512 bytes.

8.5 Allocation of the volume space

The logical sector is the unit of allocation for the volume space. Volume space may be allocated for the recording of Volume Descriptor Sequences, or Anchor Volume Descriptor Pointers, or may be allocated to partitions. This allocation shall be done from the unallocated volume space which shall be specified as extents of logical sectors by the prevailing instance of the Unallocated Space Descriptor (see 3/10.8).

Volume Descriptor Sequences and Anchor Volume Descriptor Pointers shall not be recorded in volume space that has been allocated to a rewritable or overwritable partition.

Note 10 - Implementations should not assume that the sum of the allocated logical sectors and the unallocated logical sectors in a volume equals the size of the volume space. Parts of the volume space might be unallocated for several reasons, including media defects or for use by processes external to this part of ISO/IEC 13346.

8.6 Volume set

A volume set shall consist of one or more volumes having a volume set identification common to all volumes in the volume set. The volumes in a volume set shall be numbered with consecutive integers assigned in an ascending sequence starting from 1. This number shall be the assigned volume sequence number of the volume.

Each prevailing Primary Volume Descriptor recorded on a volume contains a volume set identification consisting of the contents of the Volume Set Identifier and Descriptor Character Set fields, a volume identification consisting of the contents of the Volume Identifier and Descriptor Character Set fields, and specifies whether that volume set identification is common to each volume of the volume set. Exactly one of the volume set identifications specified on a volume shall be marked as being common to each volume of the volume set (see 3/10.1.21). The same volume identification shall not be specified by more than one volume of a volume set.

8.7 Partition

A partition is an extent of a volume and shall be identified by a Partition Number in the range 0 to 65 535 inclusive. The information about a partition shall be recorded in a Partition Descriptor. The prevailing instance of the Partition Descriptor with a specific Partition Number shall specify whether volume space has been allocated to the partition and may specify an identification of the partition's contents.

Note 11 - Partitions may overlap. This allows media to be initialised with several predefined partition definitions of varying sizes and locations. A user can then simply select a set of nonoverlapping partitions to use. In general, it is inadvisable to use file systems on overlapping partitions.

8.8 Logical volume

A Logical Volume Descriptor specifies a logical volume identification, the logical block size of the logical volume, and an ordered list of partitions comprising the logical volume. The partitions of a logical volume may be on different volumes of a volume set.

The partitions in a logical volume shall be numbered with consecutive integers assigned in an ascending sequence starting from 0. This number shall be the assigned partition reference number and shall be the numeric identification referred to in clause 3/3.2.

A logical volume shall be described by the prevailing instance of a Logical Volume Descriptor recorded on the volume with the highest volume sequence number in a volume set. Each Logical Volume Descriptor recorded on a volume set contains a logical volume identification consisting of the contents of the Logical Volume Identifier and Descriptor Character Set fields. More than one logical volume may be recorded on a volume set. Logical Volume Descriptors for all the logical volumes in a volume set shall be recorded in the volume with the highest volume sequence number.

Note 12 - A logical volume provides a segmented address space that can span multiple partitions and volumes of a volume set. As a consequence of this, a logical volume may only belong to one volume set.

8.8.1 Logical blocks

The logical sectors of a partition in a logical volume shall be organised into logical blocks of equal length. The length of a logical block shall be referred to as the logical block size and shall be an integral multiple of 512 bytes not less than the size of the logical sector of the volume. The logical block size of each partition of a logical volume shall be the same.

Each logical block shall begin in a different logical sector, and shall start with the logical sector having the next higher logical sector number than that of the last logical sector constituting the previous, if any, logical block of the partition. The first byte of a logical block shall be the first byte of the logical sector in which it begins, and if the logical sector size is smaller than the logical block size, then the logical block shall comprise a sequence of constituent logical sectors with consecutive ascending logical sector numbers. Within the logical sector containing the last byte of a logical block, the interpretation of any bytes after that last byte is not specified by this part of ISO/IEC 13346.

Each logical block of a partition shall be identified by a unique logical block number. Logical block numbers shall be consecutive integers assigned in ascending sequence. The logical block containing logical sector 0 shall have a logical block number of 0.

8.8.2 Logical volume integrity

Logical volume integrity describes the status of the information recorded on a logical volume. This status shall be specified by a Logical Volume Integrity Descriptor (see 3/10.10).

The Logical Volume Integrity Descriptors for a logical volume shall be recorded in a Logical Volume Integrity Sequence which shall be recorded as a sequence of extents. The first extent shall be specified by the prevailing Logical Volume Descriptor for the logical volume. Succeeding extents, if any, shall be specified by a Logical Volume Integrity Descriptor. Processing of an extent of Logical Volume Integrity Descriptors shall be as if the descriptors

were processed in order of ascending order of their addresses and processing was terminated by an unrecorded logical sector, or a Terminating Descriptor (see 3/10.9) or after a descriptor specifying a subsequent extent. After processing all such extents, the last Logical Volume Integrity Descriptor processed shall be used and shall be referred to as the prevailing Logical Volume Integrity Descriptor.

The status of a logical volume shall be specified by the prevailing Logical Volume Integrity Descriptor as follows:

— An Open Integrity Descriptor shall be recorded before any data is recorded in the logical volume since the last Close Integrity Descriptor, if any, was recorded.

— A Close Integrity Descriptor may be recorded only after the data recorded on the logical volume is in some consistent form not specified by this part of ISO/IEC 13346.

9 Volume recognition structures

9.1 NSR Descriptor

The NSR Descriptor shall be recorded in the format shown in figure 3/5.

Note 13 - This descriptor only indicates that a volume may have been recorded to this part of ISO/IEC 13346; in particular, see 3/3.1 and 3/3.2.

BP	Length	Name	Contents
0	1	Structure Type	`Uint8` (1/7.1.1) = 0
1	5	Standard Identifier	bytes = “NSR03”
6	1	Structure Version	`Uint8` (1/7.1.1) = 1
7	1	Reserved	#00 byte
8	2 040	Structure Data	#00 bytes

Figure 5 — NSR Descriptor format

9.1.1 Structure Type (BP 0)

This field shall specify 0.

9.1.2 Standard Identifier (BP 1)

This field shall specify “NSR03”.

9.1.3 Structure Version (BP 6)

This field shall specify the version of this descriptor. The value 1 shall indicate the structure of this part of ISO/IEC 13346.

9.1.4 Reserved (BP 7)

This field shall be reserved for future standardisation and shall be set to 0.

9.1.5 Structure Data (BP 8)

This field shall be reserved for future standardisation and all bytes shall be set to #00.

10 Volume data structures

10.1 Primary Volume Descriptor

The Primary Volume Descriptor shall identify a volume and certain attributes of that volume. It shall be recorded in the format shown in figure 3/6.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=1)
16	4	Volume Descriptor Sequence Number	`Uint32` (1/7.1.5)
20	4	Primary Volume Descriptor Number	`Uint32` (1/7.1.5)
24	32	Volume Identifier	`dstring` 1/7.2.12)
56	2	Volume Sequence Number	`Uint16` (1/7.1.3)
58	2	Maximum Volume Sequence Number	`Uint16` (1/7.1.3)
60	2	Interchange Level	`Uint16` (1/7.1.3)
62	2	Maximum Interchange Level	`Uint16` (1/7.1.3)
64	4	Character Set List	`Uint32` (1/7.1.5)
68	4	Maximum Character Set List	`Uint32` (1/7.1.5)
72	128	Volume Set Identifier	`dstring` (1/7.2.12)
200	64	Descriptor Character Set	`charspec` (1/7.2.1)
264	64	Explanatory Character Set	`charspec` (1/7.2.1)
328	8	Volume Abstract	`extent_ad` (3/7.1)
336	8	Volume Copyright Notice	`extent_ad` (3/7.1)
344	32	Application Identifier	`regid` (1/7.4)
376	12	Recording Date and Time	`timestamp` (1/7.3)
388	32	Implementation Identifier	`regid` (1/7.4)
420	64	Implementation Use	bytes
484	4	Predecessor Volume Descriptor Sequence Location	`Uint32` (1/7.1.5)
488	2	Flags	`Uint16` (1/7.1.3)
490	22	Reserved	#00 bytes

Figure 6 — Primary Volume Descriptor format

10.1.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 1.

10.1.2 Volume Descriptor Sequence Number (BP 16)

This field shall specify the Volume Descriptor Sequence Number for this descriptor.

10.1.3 Primary Volume Descriptor Number (BP 20)

This field shall specify the assigned Primary Volume Descriptor Number for this Primary Volume Descriptor.

10.1.4 Volume Identifier (BP 24)

This field shall specify an identification of the volume.

10.1.5 Volume Sequence Number (BP 56)

This field shall specify the ordinal number of the volume in the volume set of which the volume is a member.

10.1.6 Maximum Volume Sequence Number (BP 58)

This field shall specify the ordinal number of the volume in the volume set with the largest assigned volume sequence number at the time this descriptor was recorded. If this field contains 0, there is no such identification.

10.1.7 Interchange Level (BP 60)

This field shall specify the current level of medium interchange (3/11) of the volume described by this descriptor.

10.1.8 Maximum Interchange Level (BP 62)

This field shall specify the maximum value that may be specified for the Interchange Level field of this descriptor.

10.1.9 Character Set List (BP 64)

This field shall identify the character sets specified by any field, whose contents are specified to be a `charspec` (1/7.2.1), of any descriptor specified in this part of ISO/IEC 13346 and recorded on the volume described by this descriptor.

10.1.10 Maximum Character Set List (BP 68)

The Character Set List field in this descriptor shall not specify a character set (see 1/7.2.11) not specified by the Maximum Character Set List field.

Note 14 - The Interchange Level, Maximum Interchange Level, Character Set List and Maximum Character Set List fields permit an implementation to:

— determine whether it can process all of the information on the volume.

— restrict the recording of information on the volume so that the volume does not exceed the level given in the Maximum Interchange Level field.

— restrict the recording of information on the volume so that all character sets recorded belong to the Maximum Character Set List field.

This allows a user to create a volume that can be processed when it is returned to the user.

10.1.11 Volume Set Identifier (BP 72)

This field shall specify an identification of the volume set of which the volume is a member.

10.1.12 Descriptor Character Set (BP 200)

This field shall specify the d-characters (1/7.2) allowed in the Volume Identifier and Volume Set Identifier fields.

10.1.13 Explanatory Character Set (BP 264)

This field shall specify how to interpret the contents of the Volume Abstract and Volume Copyright Notice extents.

10.1.14 Volume Abstract (BP 328)

This field shall specify an extent of logical sectors containing an abstract for this volume. If the extent's length is 0, no abstract is specified.

10.1.15 Volume Copyright Notice (BP 336)

This field shall specify an extent of logical sectors containing a copyright notice for this volume. If the extent's length is 0, no copyright notice is specified.

10.1.16 Application Identifier (BP 344)

This field shall specify an identification of an application. If this field contains all #00 bytes, then no such application is identified.

10.1.17 Recording Date and Time (BP 376)

This field shall indicate the date and time of the day at which this descriptor was recorded.

10.1.18 Implementation Identifier (BP 388)

This field shall specify an identification of an implementation which can recognise and act upon the contents of the Implementation Use field. If this field contains all #00 bytes, then no such implementation is identified. The scope of this `regid` includes the contents of all descriptors, other than Implementation Use Volume Descriptors, in the Volume Descriptor Sequence in which the Primary Volume Descriptor is recorded.

10.1.19 Implementation Use (BP 420)

This field shall be reserved for implementation use. Its content is not specified by ISO/IEC 13346.

10.1.20 Predecessor Volume Descriptor Sequence Location (BP 484)

This field shall specify the address of the extent of logical sectors in which the immediately preceding extent of the Volume Descriptor Sequence of the volume is recorded. If this field contains 0, it shall mean that no such extent is identified.

Note 15 - This is intended for use in disaster recovery.

10.1.21 Flags (BP 488)

This field shall specify certain characteristics of this Primary Volume Descriptor as shown in figure 3/7.

Bit	Interpretation
0	Volume Set Identification: If set to ZERO, shall mean that the volume set identification in this descriptor need not be common among all volumes in the volume set that this volume belongs to; If set to ONE, shall mean that the volume set identification in this descriptor is common among all volumes in the volume set that this volume belongs to.
1-15	Shall be reserved for future standardisation and all bits shall be set to ZERO.

Figure 7 — Primary Volume Descriptor characteristics

10.1.22 Reserved (BP 490)

This field shall be reserved for future standardisation and all bytes shall be set to #00.

10.2 Anchor Volume Descriptor Pointer

The Anchor Volume Descriptor Pointer shall specify an extent of the Main and Reserve Volume Descriptor Sequences recorded on the volume. It shall be recorded in the format shown in figure 3/8.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=2)
16	8	Main Volume Descriptor Sequence Extent	`extent_ad` (3/7.1)
24	8	Reserve Volume Descriptor Sequence Extent	`extent_ad` (3/7.1)
32	480	Reserved	#00 bytes

Figure 8 — Anchor Volume Descriptor Pointer format

10.2.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 2.

10.2.2 Main Volume Descriptor Sequence Extent (BP 16)

This field shall specify an extent of the Main Volume Descriptor Sequence.

Note 16 - The extent specifies allocation rather than recording; that is, space that may be available for recording rather than what is actually recorded. The extent need not be completely recorded.

10.2.3 Reserve Volume Descriptor Sequence Extent (BP 24)

This field shall specify an extent of the Reserve Volume Descriptor Sequence. If the extent's length is 0, no such extent is specified.

Note 17 - The extent specifies allocation rather than recording; that is, space that may be available for recording rather than what is actually recorded. The extent need not be completely recorded.

10.2.4 Reserved (BP 32)

This field shall be reserved for future standardisation and all bytes shall be set to #00.

10.3 Volume Descriptor Pointer

The Volume Descriptor Pointer shall specify an extent of a Volume Descriptor Sequence recorded on the volume. It shall be recorded in the format shown in figure 3/9.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=3)
16	4	Volume Descriptor Sequence Number	`Uint32` (1/7.1.5)
20	8	Next Volume Descriptor Sequence Extent	`extent_ad` (3/7.1)
28	484	Reserved	#00 bytes

Figure 9 — Volume Descriptor Pointer format

10.3.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 3.

10.3.2 Volume Descriptor Sequence Number (BP 16)

This field shall specify the Volume Descriptor Sequence Number for this descriptor.

10.3.3 Next Volume Descriptor Sequence Extent (BP 20)

This field shall specify the next extent in the Volume Descriptor Sequence. If the extent's length is 0, no such extent is specified.

Note 18 - The extent specifies allocation rather than recording; that is, space that may be available for recording rather than what is actually recorded. The extent need not be completely recorded.

10.3.4 Reserved (BP 28)

This field shall be reserved for future standardisation and all bytes shall be set to #00.

10.4 Implementation Use Volume Descriptor

The Implementation Use Volume Descriptor shall identify an implementation which can recognise and act upon the contents of this descriptor's Implementation Use field. It shall be recorded in the format shown in figure 3/10.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=4)
16	4	Volume Descriptor Sequence Number	`Uint32` (1/7.1.5)
20	32	Implementation Identifier	`regid` (1/7.4)
52	460	Implementation Use	bytes

Figure 10 — Implementation Use Volume Descriptor format

10.4.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 4.

10.4.2 Volume Descriptor Sequence Number (BP 16)

This field shall specify the Volume Descriptor Sequence Number for this descriptor.

10.4.3 Implementation Identifier (BP 20)

This field shall specify an identification of an implementation which can recognise and act upon the contents of the Implementation Use field. If this field contains all #00 bytes, then no such implementation is identified.

10.4.4 Implementation Use (BP 52)

This field shall be reserved for implementation use. Its content is not specified by ISO/IEC 13346.

10.5 Partition Descriptor

The Partition Descriptor shall specify the size and location of a partition and shall be recorded in the format shown in figure 3/11.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=5)
16	4	Volume Descriptor Sequence Number	`Uint32` (1/7.1.5)
20	2	Partition Flags	`Uint16` (1/7.1.3)
22	2	Partition Number	`Uint16` (1/7.1.3)
24	32	Partition Contents	`regid` (1/7.4)
56	128	Partition Contents Use	bytes
184	4	Access Type	`Uint32` (1/7.1.5)
188	4	Partition Starting Location	`Uint32` (1/7.1.5)
192	4	Partition Length	`Uint32` (1/7.1.5)
196	32	Implementation Identifier	`regid` (1/7.4)
228	128	Implementation Use	bytes
356	156	Reserved	#00 bytes

Figure 11 — Partition Descriptor format

10.5.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 5.

10.5.2 Volume Descriptor Sequence Number (BP 16)

This field shall specify the Volume Descriptor Sequence Number for this descriptor.

10.5.3 Partition Flags (BP 20)

This field shall specify certain characteristics of the partition as shown in figure 3/12.

Bit	Interpretation
0	Allocation: If set to ZERO, shall mean that volume space has not been allocated for this partition; If set to ONE, shall mean that volume space has been allocated for this partition.
1-15	Shall be reserved for future standardisation and all bits shall be set to ZERO.

Figure 12 — Partition characteristics

10.5.4 Partition Number (BP 22)

This field shall specify the numeric identifier for the partition.

Note 19 - The Partition Number may be 0.

10.5.5 Partition Contents (BP 24)

This field shall specify an identification of how to interpret the contents of the partition. The identifications specified by this part of ISO/IEC 13346 are given in figure 3/13. Other identifications shall be specified according to 1/7.4.

Contents	Interpretation
“+FDC01”	As if it were a volume recorded according to ISO 9293-1987.
“+CD001”	As if it were a volume recorded according to ISO 9660.
“+CDW02”	As if it were a volume recorded according to ISO/IEC 13490.
“+NSR03”	According to ISO/IEC 13346-4.

Figure 13 — Partition content interpretation

10.5.6 Partition Contents Use (BP 56)

This field shall specify information required for the interpretation of the information recorded on the partition identified by this Partition Descriptor. The contents of this field shall be specified by the relevant standard for the interpretation of the information recorded on the partition.

10.5.7 Access Type (BP 184)

This field shall specify the access methods which are permitted on the logical sectors of the partition described by this Partition Descriptor. The access types are given in figure 3/14.

Type	Interpretation
0	The type of access is not specified by this field.
1	Read only: there shall be no restriction on reading logical sectors; logical sectors shall not be recorded.
2	Write once: there shall be no restriction on reading logical sectors; logical sectors shall only be recorded once.
3	Rewritable: there shall be no restriction on reading logical sectors; logical sectors may require preprocessing before recording.
4	Overwritable: there shall be no restriction on reading or recording logical sectors.
5 and above	Reserved for future standardisation.

Figure 14 — Access interpretation

10.5.8 Partition Starting Location (BP 188)

This field shall specify the logical sector number at which the partition begins.

10.5.9 Partition Length (BP 192)

This field shall specify the number of logical sectors which comprise the partition.

10.5.10 Implementation Identifier (BP 196)

This field shall specify an identification of an implementation which can recognise and act upon the contents of the Implementation Use field. If this field contains all #00 bytes, then no such implementation is identified.

Note 20 - The scope of this `regid` does not include the contents of the partition.

10.5.11 Implementation Use (BP 228)

This field shall be reserved for implementation use and its contents are not specified by ISO/IEC 13346.

10.5.12 Reserved (BP 356)

This field shall be reserved for future standardisation and all bytes shall be set to #00.

10.6 Logical Volume Descriptor

The Logical Volume Descriptor shall be recorded in the format shown in figure 3/15.

BP	Length	Name	Contents
0	16	Descriptor Tag	`tag` (3/7.2) (Tag=6)
16	4	Volume Descriptor Sequence Number	`Uint32` (1/7.1.5)
20	64	Descriptor Character Set	`charspec` (1/7.2.1)
84	128	Logical Volume Identifier	`dstring` (1/7.2.12)
212	4	Logical Block Size	`Uint32` (1/7.1.5)
216	32	Domain Identifier	`regid` (1/7.4)
248	16	Logical Volume Contents Use	bytes
264	4	Map Table Length (=MT_L)	`Uint32` (1/7.1.5)
268	4	Number of Partition Maps (=N_PM)	`Uint32` (1/7.1.5)
272	32	Implementation Identifier	`regid` (1/7.4)
304	128	Implementation Use	bytes
432	8	Integrity Sequence Extent	`extent_ad` (3/7.1)
440	MT_L	Partition Maps	bytes

Figure 15 — Logical Volume Descriptor format

10.6.1 Descriptor Tag (BP 0)

The Tag Identifier field of the `tag` (3/7.2) for this descriptor shall contain 6.

10.6.2 Volume Descriptor Sequence Number (BP 16)

This field shall specify the Volume Descriptor Sequence Number for this descriptor.

10.6.3 Descriptor Character Set (BP 20)

This field shall specify the d-characters (1/7.2) allowed in the Logical Volume Identifier field.

10.6.4 Logical Volume Identifier (BP 84)

This field shall specify an identification of the logical volume.

10.6.5 Logical Block Size (BP 212)

This field shall specify the size of a logical block in bytes.

10.6.6 Domain Identifier (BP 216)

This field shall specify an identification of a domain which shall specify rules on the use of, and restrictions on, certain fields in descriptors subject to agreement between the originator and recipient of the medium. If this field contains all #00 bytes, then no such domain is identified. The scope of this `regid` (1/7.4) shall include all information recorded in the logical volume described by this descriptor, and shall include the scope of the Implementation Identifier field.

10.6.7 Logical Volume Contents Use (BP 248)

This field shall specify information required for the interpretation of the information recorded on the logical volume identified by this Logical Volume Descriptor. The contents of this field shall be specified by the relevant standard for the interpretation of the information recorded on the logical volume.

10.6.8 Map Table Length (=MT_L) (BP 264)

This field shall specify the length of the Partition Maps field in bytes.

10.6.9 Number of Partition Maps (=N_PM) (BP 268)

This field shall specify the number of Partition Maps recorded in the Partition Maps field.

10.6.10 Implementation Identifier (BP 272)

This field shall specify an identification of an implementation which can recognise and act upon the contents of the Implementation Use field. If this field contains all #00 bytes, then no such implementation is identified.

The scope of this `regid` includes the contents of any Partition Descriptors identified by Type 1 Partition Maps recorded in the Partition Maps field.

10.6.11 Implementation Use (BP 304)

This field shall be reserved for implementation use and its contents are not specified by ISO/IEC 13346.

10.6.12 Integrity Sequence Extent (BP 432)

This field shall specify the first extent of the Logical Volume Integrity Sequence. The extent shall be within the volume in which this descriptor is recorded. If N_PM is 0, then the extent's length may be 0. If the extent's length is 0, then no such extent is specified.

10.6.13 Partition Maps (BP 440)

This field shall contain N_PM Partition Maps recorded contiguously starting at the first byte of the field. The Partition Maps may be of different types. The length of the Partition Maps shall not exceed MT_L bytes and any unused bytes shall be set to #00.

As specified by 3/8.4.4, the remainder of the last logical sector comprising the Logical Volume Descriptor shall be recorded with #00 bytes.

10.7 Partition maps

10.7.1 Generic partition map

A partition map shall be recorded in the format shown in figure 3/16.

RBP	Length	Name	Contents
0	1	Partition Map Type	Uint8 (1/7.1.1)
1	1	Partition Map Length (= PM_L)	Uint8 (1/7.1.1)
2	PM_L-2	Partition Mapping	bytes

Figure 16 — Generic partition map format

10.7.1.1 Partition Map Type (RBP 0)

The number in this field shall specify the type of the partition map. The types are given in figure 3/17.

Type	Interpretation
0	Shall mean that the type of the partition map is not specified by this field.
1	Shall mean that the partition map is a Type 1 Partition Map (see 3/10.7.2).
2	Shall mean that the partition map is a Type 2 Partition Map (see 3/10.7.3).
3-255	Reserved for future standardisation.

Figure 17 — Partition maps

10.7.1.2 Partition Map Length (= PM_L) (RBP 1)

This field shall specify the length, in bytes, of this partition map, including the Partition Map Type and Partition Map Length fields.

10.7.1.3 Partition Mapping (RBP 2)

The interpretation of this field shall be specified by the standard or clause identified in the Partition Map Type field, or shall be subject to agreement between the originator and recipient of the medium if the number in the Partition Map Type field is 0.

10.7.2 Type 1 Partition Map

This map type identifies a partition on a volume in the volume set on which the logical volume is recorded. It shall be recorded in the format shown in figure 3/18.

RBP	Length	Name	Contents
0	1	Partition Map Type	Uint8 (1/7.1.1) = 1
1	1	Partition Map Length	Uint8 (1/7.1.1) = 6
2	2	Volume Sequence Number	Uint16 (1/7.1.3)
4	2	Partition Number	Uint16 (1/7.1.3)

Figure 18 — Type 1 Partition Map format

10.7.2.1 Partition Map Type (RBP 0)

This field shall specify 1.

10.7.2.2 Partition Map Length (RBP 1)

This field shall specify 6.

10.7.2.3 Volume Sequence Number (RBP 2)

This field specifies the volume, in the volume set on which this logical volume is recorded, whose volume sequence number is identical to the contents of this field.

10.7.2.4 Partition Number (RBP 4)

This field specifies the partition of the volume specified by the Volume Sequence Number field, identified by the Partition Descriptor whose Partition Number field is identical to the contents of the Partition Number field.

10.7.3 Type 2 Partition Map

This map type identifies a partition in a manner subject to agreement between the originator and recipient of the medium.

It shall be recorded in the format shown in figure 3/19.

Note 21 - Type 2 maps allow partitions to be identified in an implementation specific way that is outside the scope of this part of ISO/IEC 13346. This allows partitions recorded in a manner not specified by this part of ISO/IEC 13346, such as a local disk partition which might locally be referred to as "`/dev/dsk/ipi0d2p4`" or "`a:`" or "`NODE::DEVICE:`", to be part of a logical volume. Type 2 maps may present problems when interchanging media.

RBP	Length	Name	Contents
0	1	Partition Map Type	`Uint8` (1/7.1.1) = 2
1	1	Partition Map Length	`Uint8` (1/7.1.1) = 64
2	62	Partition Identifier	bytes

Figure 19 — Type 2 Partition Map format

10.7.3.1 Partition Map Type (RBP 0)

This field shall specify 2.

10.7.3.2 Partition Map Length (RBP 1)

This field shall specify 64.

10.7.3.3 Partition Identifier (RBP 2)

This field shall specify an identification of the partition in a manner subject to agreement between the originator and recipient of the medium.

10.8 Unallocated Space Descriptor

The Unallocated Space Descriptor shall specify extents that are unallocated. It shall be recorded in the format shown in figure 3/20.