Standard ECMA-119

5th Edition / December 2024

Volume and File Structure of CDROM for Information Interchange

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

Contents

Page

Introduction

In the middle 1980s, the compact disk, originally developed for recording music, began to also be used for recording data, as it could record large amounts of information in a reliable and economic manner. In addition, due to the feature of read-only medium, it seemed to be particularly suitable for use in applications such as auditing and legal documents. Therefore, there was an urgent need for stable standards for structures of the compact disk and of the files recorded thereon.

In October 1985 a number of industrial and software companies in the USA invited experts to participate in the development of a working paper for such a project. The result of this work, in which also expert members of Ecma TC15 as well as from Japan participated, was a report dated May 1986 and known as the “High Sierra Group” proposal.

This proposal was submitted in Europe to Ecma International for their consideration. Ecma TC15, in collaboration with experts from user organizations, invested a considerable amount of work into this proposal in order to clarify and complete its technical contents and to re-edit it in a form suitable for a Standard. Particular attention was given to conformance aspects by applying the same editing principles as for the other standards for labelling, such as ECMA-13 (ISO 1001) and ECMA-107 (ISO 9293). As a result, Standard ECMA-119 was issued in December 1986.

This Ecma Standard was submitted by Ecma International to ISO for processing as an International Standard under the ISO fast-track procedure, which resulted in International Standard ISO 9660:1988 in which a number of improvements and editorial amendments were introduced. The 2nd edition of ECMA-119 is technically identical with ISO 9660:1988.

Since the publication of the 2nd Edition of ECMA-119, Ecma TC31 and ISO/IEC JTC1/SC23, which is responsible for the international standardization of digitally recorded media for information interchange and storage, had collaboratively maintained the Standards in order to contain the "Joliet Specification" defined as an extension of ISO 9660:1988 and widely used on PC. As the results, ECMA-119 was revised as the 3rd edition in December 2017 and 4th edition in June 2019. And ISO 9660/Amd. 1:2013 and ISO 9660/Amd. 2:2020 were additionally published.

In 2023, SC23 developed ISO/IEC 9660:2023 in order to consolidate ISO 9660:1988 and the two amendments. In this development, a change of the normative reference specifying recording and addressing methods for the compact disk to ECMA-130 (ISO/IEC 10149), which specifies the characteristics of read-only compact disks for information interchange between information processing systems and for information storage and was first published in 1989, was also made.

This 5th edition of ECMA-119 has been developed for being technically identical with ISO/IEC 9660:2023.

This Ecma Standard was developed by Technical Committee 31 and was adopted by the General Assembly of December 2024.

Volume and File Structure of CDROM for Information Interchange

1 Scope

This Ecma Standard specifies the volume and file structure of compact disc read-only memory (CD-ROM) for the interchange of information between users of information processing systems.

This Ecma Standard specifies:

- the attributes of the volume and the descriptors recorded on it;
- the relationship among volumes of a volume set;
- the placement of the files;
- the attributes of the files;
- record structures intended for use in the input or output data streams of an application program when such data streams are required to be organized as sets of records;
- three nested levels of medium interchange;
- two nested levels of implementation;
- requirements for the processes which are provided within information processing systems, to enable information to be interchanged between different systems, utilizing recorded CD-ROM as the medium of interchange; for this purpose, this Ecma Standard specifies the functions to be provided within systems which are intended to originate or receive CD-ROM which conform to this Ecma Standard.

2 Conformance

2.1 Conformance of a CD-ROM

A CD-ROM is considered to conform to this Ecma Standard if all information recorded on it conforms to the requirements of Clauses 7 to 11. A statement of conformance shall identify the lowest level of interchange to which the contents of the CD-ROM conform.

A prerequisite to such conformance is conformance of the CD-ROM to ECMA-130.

2.2 Conformance of an information processing system

An information processing system is considered to conform to this Ecma Standard if it meets the requirements specified in Clauses 7 to 11 and 13 to 15, either for an originating system, or for a receiving system, or for both types of system. A statement of conformance shall identify which level of these requirements can be met by the system.

3 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this Ecma Standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ECMA-6, *7-Bit Coded Character Set*

ECMA-35, *Character code structure and extension techniques*

ECMA-130, *Data interchange on read-only 120 mm optical data disks (CD-ROM)*

ISO/IEC 1539-1, *Information technology — Programming languages - Fortran - Part 1: Base language*

ISO/IEC 2375, *Information technology — Procedure for registration of escape sequences and coded character sets*

4 Terms and definitions

For the purposes of this Ecma Standard, the following terms and definitions apply.

4.1
application program
program that processes the contents of a *file (4.7)*, and may also process selected attribute data relating to a file or to a *volume(s)* (4.16) on which a file is recorded

NOTE An application program is a specific class of user, as defined in this Ecma Standard.

4.2
byte
string of eight binary digits operated upon as a unit

4.3
data field
fixed-length field containing data of a *sector* (4.14)

4.4
data preparer
person or other entity which controls the preparation of data to be recorded on a *volume* (4.16) group

NOTE A data preparer is a specific class of *user* (4.15) as defined in this Ecma Standard.

4.5
descriptor
structure containing descriptive information about *a volume* (4.16) or *a file* (4.7)

4.6
extent
set of *logical blocks* (4.10), the logical block numbers of which form a continuous ascending sequence

4.7
file
named collection of information

4.8
file section
part of a *file* (4.7) that is recorded in any one *extent* (4.6)

4.9
implementation
set of processes which enable an information processing system to behave as an *originating system* (4.11), or as a *receiving system* (4.12), or as both types of system

4.10
logical block
group of 2^{n+9} *bytes* (4.2) treated as a logical unit, where *n* equals 0 or a positive integer

4.11
originating system
information processing system which can create a set of *files* (4.7) on a *volume set* (4.17) for a purpose of data interchange with another system

4.12
receiving system
information processing system which can read a set of *files* (4.7) from a *volume set* (4.17) created by another system for a purpose of data interchange

4.13
record
sequence of *bytes* (4.2) treated as a unit of information

4.14
sector
smallest addressable part of a recorded area on a CD-ROM that can be accessed independently of other addressable parts of the recorded area

NOTE In ECMA-167, a sector is defined as "the data field of the smallest addressable part of the medium that can be accessed independently of other addressable parts of the medium”.

4.15
user
person or other entity [for example, *application program* (4.1) that causes an invocation of services provided by *implementation* (4.9)

4.16
volume
dismountable CD-ROM

NOTE In ECMA-167, a volume is defined as “a sector address space as specified in a standard that specifies the recording method and the addressing method for the information recorded on a medium”.

4.17
volume set
collection of one or more *volumes* (4.16), on which a set of *files* (4.7) is recorded

5 Notation

5.1 Decimal and hexadecimal notations

Numbers in decimal notation are represented by decimal digits, namely 0 to 9.

Numbers in hexadecimal notation are represented by hexadecimal digits, namely 0 to 9 and A to F, shown in parentheses.

5.2 Other notations

BP	byte position within a descriptor, starting with 1
Digit(s)	any digit from ZERO to NINE
ONE	single bit with value 1
RBP	byte position within a descriptor field, starting with 1

ZERO	single bit with value 0

6 Requirements for a medium

The volume and file structure of CD-ROM are specified from the following two perspectives:

— requirements for the medium;

— requirements for systems.

The requirements for the medium are defined and specified in Clauses 7 to 11.

7 Volume structure

7.1 Arrangement of data on a CD-ROM

7.1.1 Physical addresses

Each sector shall be identified by a unique physical address in accordance with ECMA-130.

7.1.2 Logical sector

The sectors of a volume shall be organized into logical sectors. Each logical sector shall consist of a number of bytes equal to 2 048 or 2^n, whichever is larger, where *n* is the largest integer such that 2^n is less than, or equal to, the number of bytes in the data field of any sector recorded on the volume. The number of bytes in a logical sector shall be referred to as the logical sector size. Each logical sector shall begin in a different sector from any other logical sector and shall begin with the first byte of the data field of the sector in which it begins. If the number of bytes of the data field of each sector recorded on the volume is less than 2 048, a logical sector shall comprise more than one sector, and the set of the physical addresses of its constituent sectors shall form a consecutive ascending sequence. The data of a logical sector shall be recorded in the data fields of its constituent sectors.

Each logical sector shall be identified by a unique logical sector number. Logical sector numbers shall be integers assigned in an ascending sequence, in order of ascending physical addresses of the constituent sectors, starting with 0 for the logical sector containing the sector having the lowest physical address which may contain recorded information. The numbering shall continue through successive logical sectors, each of which begins with the sector with the next higher physical address than that of the last sector constituting the previous logical sector.

7.1.3 Volume space

The information on a volume shall be recorded in the set of all logical sectors on the volume. This set shall be referred to as the volume space of the volume.

The bytes in the volume space shall be numbered consecutively. The numbering shall start with 1, which shall be assigned to the first byte of the first logical sector of the volume space. The numbering shall continue through successive bytes of the first logical sector, and then through successive bytes of each successive logical sector, of the volume space.

7.2 Arrangement of the volume space

7.2.1 System area and data area

The volume space shall be divided into a system area and a data area.

The system area shall occupy the logical sectors with logical sector numbers 0 to15. The system area shall be reserved for system use. Its content is not specified by this Ecma Standard.

The data area shall occupy the remaining logical sectors of the volume space.

7.2.2 Logical block

The volume space shall be organized into logical blocks. Each logical block shall consist of 2^{n+9} bytes, where *n* equals 0 or a positive integer. The number of bytes in a logical block shall be referred to as the logical block size which shall not be greater than the logical sector size.

Each logical block shall be identified by a unique logical block number. Logical block numbers shall be integers assigned in ascending order starting with 0. Logical block number 0 shall be assigned to the logical block which begins with the first byte of the volume space. Each successive logical block number shall be assigned to the logical block which begins with the byte in the volume space immediately following the last byte of the preceding logical block.

7.3 Arrangement of the data area

File sections shall be recorded in the data area.

The following types of descriptors shall be recorded in the data area to describe the use of the data area:

— volume descriptors;

— file descriptors;

— directory descriptor;

— path tables.

The volume descriptors shall be recorded in consecutively numbered logical sectors starting with the logical sector having logical sector number 16. The logical sectors in the data area shall be available for the assignment of volume partitions and the recording of file sections, file descriptors, directory descriptors and path tables.

Each file section shall be recorded in an extent, and shall be identified by a descriptor in a directory. An extended attribute record can be associated with the file section. If present, it shall be recorded in the same extent as the associated file section, and shall be identified by the descriptor that identifies the associated file section. Each directory shall be recorded as a file in a single extent, and shall be identified by a directory descriptor either in another directory or in a volume descriptor. Each directory shall also be identified by a record in a path table. Each path table shall be identified in a volume descriptor.

Space within the data area may be assigned to one or more volume partitions. Each volume partition shall be recorded in an extent and shall be identified by a volume descriptor.

7.4 Arrangement of an extent

7.4.1 Extent

An extent shall be a set of logical blocks, the logical block numbers of which form a continuous ascending sequence.

7.4.2 Mode of recording a file section

A file section, and its associated extended attribute record, if any, shall be recorded in an extent either in interleaved mode or in non-interleaved mode.

7.4.3 Interleaved mode

7.4.3.1 File unit

A file unit shall comprise a set of logical blocks that are within an extent and the logical block numbers of which form a continuous ascending sequence.

When a file section is recorded in interleaved mode, one or more file units, each consisting of the same number of logical blocks, shall be assigned to the file section within the same extent. The number of logical blocks in the file unit shall be the assigned file unit size for the file section.

The first logical block of each file unit shall have a logical block number which is the lowest logical block number in the logical sector that contains that logical block.

The sequence of the file units in an extent shall correspond to the sequence of the logical block numbers of the first logical block of each file unit.

The logical blocks comprising a file unit assigned to a file section may:

— also each be assigned to a different file section; and/or

— comprise part of one or more volume partitions.

7.4.3.2 Interleave gap

An interleave gap shall comprise the set of logical blocks that are within an extent and the logical block numbers of which lie between the last logical block number of a file unit and the first logical block number of the next file unit, if any, in the sequence. All interleave gaps between the file units assigned to a file section shall comprise the same number of logical blocks. This number shall be the assigned interleave gap size for the file section.

The logical blocks comprising an interleave gap between the file units assigned to a file section may:

— also each be assigned to a different file section; and/or

— comprise part of one or more volume partitions.

7.4.3.3 Relation of file section to file unit

When a file section is recorded in interleaved mode, the file section, and its associated extended attribute record, if any, shall be recorded over the sequence of file units assigned to the file section.

7.4.3.4 Recording of an extended attribute record

If an extended attribute record is recorded, it shall be recorded in the first file unit of the sequence. The recording shall begin at the first byte of the first logical block of the file unit. It shall continue through successive bytes of that logical block, and then through successive bytes of successive logical blocks, if any, of the file unit, until all of the extended attribute record is recorded.

The assigned extended attribute record length shall be equal to the assigned file unit size.

7.4.3.5 Recording of a file section

The successive parts, if any, of the file section shall be recorded in successive file units, starting from the second file unit in the sequence if an extended attribute record is recorded, and starting from the first file unit in the sequence if no extended attribute record is recorded.

7.4.3.6 Data space

The set of file units in which the successive parts of the file section are recorded shall be the data space of the file section.

The bytes in the data space shall be numbered consecutively. The numbering shall start from 1 which shall be assigned to the first byte of the first logical block of the first file unit, if any, of the data space. The numbering shall continue through successive bytes of that logical block, then through successive bytes of each successive logical block, if any, of the first file unit, and then through successive bytes of the logical block(s) of each successive file unit, if any, assigned to the file section.

The numbering shall end with a number equal to the number of bytes in which the file section is recorded in the data space, or this number shall equal zero if there are no bytes of the file recorded in the data space.

7.4.4 Non-interleaved mode

7.4.4.1 General

When a file section is recorded in non-interleaved mode, the file section and its associated extended attribute record, if any, shall be recorded over the sequence of logical blocks in an extent.

7.4.4.2 Recording of an extended attribute record

If an extended attribute record is recorded, it shall be recorded over one or more logical blocks, the logical block numbers of which form a continuous ascending sequence. The recording shall begin at the first byte of the first logical block of the extent. It shall continue through successive bytes of that logical block, and then through successive bytes of successive logical blocks, if any, of the extent, until all of the extended attribute record is recorded.

The number of logical blocks over which the extended attribute record is recorded shall be the assigned extended attribute record length for the file section.

The logical blocks comprising an extended attribute record assigned to a file section may:

— also each be assigned to a different file section; and/or

— comprise part of one or more volume partitions.

7.4.4.3 Recording of a file section

The file section shall be recorded over zero or more logical blocks, the logical block numbers of which form a continuous ascending sequence. If no extended attribute record is recorded, the sequence shall start with the first logical block of the extent. If an extended attribute record is recorded, the sequence shall start with the first logical block, if any, immediately following the last logical block over which the extended attribute record is recorded.

7.4.4.4 Data space

The set of logical blocks over which the file section is recorded shall be the data space of the file section.

The bytes in the data space shall be numbered consecutively. The numbering shall start from 1 which shall be assigned to the first byte of the first logical block, if any, of the data space. The numbering shall continue through successive bytes of that logical block, and then through successive bytes of each successive logical block, if any, of the data space.

The numbering shall end with a number equal to the number of bytes in which the file section is recorded in the data space; or this number shall equal zero if there are no bytes of the file recorded in the data space.

7.4.5 Data length of a file section

The data length of a file section shall be the number of bytes in which the file section is recorded in the data space. If this number is less than the number of bytes in the data space, then any remaining bytes in the data space shall be ignored in interchange.

7.4.6 Relation of extended attribute record to file section

An extended attribute record may be associated with a file section. If present, the extended attribute record shall identify certain attributes of the file of which the file section forms a part.

A subset of those attributes shall apply to all file sections of a file that contains records according to 7.10. If any of those attributes are assigned to the file, an extended attribute record shall be recorded in association with each of the file sections of the file.

The other attributes identified in an extended attribute record shall apply to that file section and all preceding file sections of the file (see 7.5.1). If no extended attribute record is recorded in association with the last file section of a file, then these attributes are not specified for the file.

7.4.7 Recording of a volume partition

If a volume partition is recorded, it shall be recorded over one or more logical blocks, the logical block numbers of which form a continuous ascending sequence. The recording shall begin at the first byte of the first logical block of the extent. It shall continue through successive bytes of that logical block, and then through successive bytes of successive logical blocks, if any, of the extent, until all of the volume partition is recorded. The first logical block of each volume partition shall have a logical block number which is the lowest logical block number in the logical sector that contains that logical block.

The number of logical blocks over which the volume partition is recorded shall be the assigned volume partition size for the volume partition.

7.5 File structure

7.5.1 Relation to file sections

Each file shall consist of one or more file sections. Each file section of a file shall be identified by a record in the same directory. The sequence of the file sections of a file shall be identified by the order of the corresponding records in the directory.

A file section may be part of more than one file and may occur more than once in the same file. A file section may be identified by more than one record in the same or a different directory.

Each file section of a file may be recorded on a different volume. More than one file section of a file may be recorded on the same volume.

7.5.2 Numbering of bytes in a file

The bytes comprising a file shall be numbered consecutively. The numbering shall start from 1 which shall be assigned to the first byte of the first file section, if any. The numbering shall continue through successive bytes of that file section, and then through successive bytes of each successive file section, if any, of the file.

The numbering shall end with a number equal to the sum of the number of bytes in all file sections of the file.

7.5.3 Contents of a file

The information in a file shall be interpreted according to the relevant standards for the coded representation of information.

NOTE The identification of these standards is the subject of an agreement between the originator and the recipient of the file.

7.5.4 Associated file

An associated file has a relationship not specified by this Ecma Standard to another file that has been assigned the same file identifier (see 8.5) as that of the associated file in the same directory.

7.6 Volume set

A volume set shall be the set of volumes on which a set of files is recorded.

A volume set shall consist of one or more volumes having common volume set identification and identifying the same coded graphic character sets for use within selected descriptor fields (see 8.4). All volumes in a volume set shall be numbered consecutively starting from 1.

A volume group within a volume set shall consist of one or more consecutively numbered volumes, the contents of which are established at the same time. The sequence number of the volume that has the highest sequence number within the volume group shall be the assigned volume set size.

Each volume of a volume set shall contain a description of all the directories and files that are recorded on those volumes the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume.

Such description recorded on a volume shall supersede the description recorded on any volume of the volume set having a lower assigned volume set size.

The logical block size shall be the same for all volumes of a volume set.

7.7 Volume descriptors

7.7.1 General

A volume descriptor shall be one of the following types:

— primary volume descriptor;

— supplementary volume descriptor;

— enhanced volume descriptor;

— volume partition descriptor;

— boot record;

— volume descriptor set terminator.

7.7.2 Volume descriptor set

A volume descriptor set shall be a sequence of volume descriptors recorded in consecutively numbered logical sectors starting with the logical sector number 16. Each successive volume descriptor shall be recorded in the logical sector with the next higher logical sector number than that of the logical sector in which the previous volume descriptor is recorded. The sequence shall consist of two or more volume descriptors consecutively recorded as follows.

a) The sequence shall contain one primary volume descriptor (see 9.4) recorded at least once.

- The primary volume descriptor shall describe the volume space, and identify the attributes of the volume, the locations of a root directory and of a group of path tables, and the number of volumes in the volume set.

b) The sequence may contain zero or more supplementary volume descriptors (see 9.5) each recorded at least once.

- A supplementary volume descriptor shall describe the volume space, and identify the attributes of the volume, the locations of a root directory and of a group of path tables, and the number of volumes in the volume set. It shall also identify the coded graphic character sets used within selected fields of this descriptor, and of the fields in associated file descriptors, directory descriptors and path tables.

c) The sequence may contain zero or more enhanced volume descriptors (see 9.5) each recorded at least once.

- An enhanced volume descriptor shall describe the volume space, and identify the attributes of the volume, the locations of a root directory and of group path tables, and the number of volumes in the volume set. It shall also identify the coded graphic character sets used within selected fields of this descriptor, and of the fields in associated file descriptors, directory descriptors and path tables.

d) The sequence may contain zero or more volume partition descriptors (see 9.6).

- A volume partition descriptor shall identify a volume partition within the volume space, its position and size, and its attributes.

e) The sequence may contain zero or more boot records (see 9.2).

- A boot record shall contain information which may be used to achieve a specific state in a receiving system or an application program.

f) The sequence shall be terminated by the recording of one or more volume descriptor set terminators (see 9.3).

7.8 Directory structure

7.8.1 Directory

7.8.1.1 General

A directory shall be recorded as a file containing a set of records each of which identifies a file section or another directory. A directory shall not be recorded as an associated file, shall not be recorded in interleaved mode and shall consist of only one file section.

The identification of a file shall be different from the identification of any other file (unless the file is an associated file; see 7.5.3) or of any directory identified in the same directory. The identification of a directory shall be different from the identification of any file or of any other directory identified in the same directory.

The first logical block of the extent in which a directory is recorded shall have a logical block number which is the lowest logical block number in the logical sector that contains that logical block.

7.8.1.2 Directory record

A directory record shall contain:

- information to locate a file section;

- information to locate any extended attribute record associated with the file section;

— the identification of the file;

— certain attributes of the file;

— certain attributes of the file section.

The first or only directory record recorded in a logical sector shall begin at the first byte of the first data field of that logical sector. Each subsequent directory record recorded in that logical sector shall begin at the byte immediately following the last byte of the preceding directory record in that logical sector. Each directory record shall end in the logical sector in which it begins. Unused byte positions after the last directory record in a logical sector shall be set to (00).

7.8.1.3 Order of directory records

The records in a directory shall be ordered according to 10.3.

7.8.1.4 Directory length

The length of a directory shall be the sum of:

— the lengths of all directory records in the directory;

— the number of unused byte positions after the last directory record in all logical sectors in which the directory is recorded.

7.8.2 Directory hierarchy

7.8.2.1 General

A directory hierarchy shall be a set of directories related to each other as follows.

— The root of the hierarchy, called the root directory, shall be a directory identified in a primary volume descriptor, supplementary volume descriptor or enhanced volume descriptor.

— Each directory, other than the root directory, shall be identified by a record in another directory.

— A directory identifying another directory shall be called the parent directory of the identified directory. Each directory shall contain a record which identifies its parent directory. Different directories may have the same parent directory.

— A hierarchical relationship shall exist between the root directory and all other directories as shown in Figure 1.

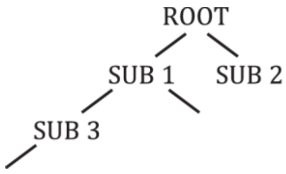

Figure 1 — Example of directory hierarchy

— The hierarchy shall consist of a number of levels (i.e. for *n* levels: level 1, level 2, …, level *n*). The root directory shall be the only directory at level 1 of the hierarchy.

— If a directory is at level *m* of the hierarchy, its parent directory shall be at level (*m* - 1). The parent directory of the root directory shall be the root directory.

7.8.2.2 Depth of directory hierarchy

For a directory hierarchy identified in a primary volume descriptor or in a supplementary volume descriptor, the number of levels in the hierarchy shall not exceed eight. For a directory hierarchy identified in an enhanced volume descriptor, the number of levels in the hierarchy may exceed eight. In addition, for each file recorded, the sum of the following shall not exceed 255:

— the length of the file identifier (see 8.5.2);

— the length of the directory identifiers (see 8.6) of all relevant directories;

— the number of relevant directories.

7.8.2.3 Identification of directories

For a root directory:

— the first directory record of the root directory shall describe the root directory and shall have a directory identifier consisting of a single (00) byte;

— the second directory record of the root directory shall describe the root directory and shall have a directory identifier consisting of a single (01) byte;

— a directory record describing the root directory shall be contained in the root directory field of the volume descriptor that identifies the directory hierarchy.

For each directory other than the root directory:

— the first directory record of the directory shall describe that directory and shall have a directory identifier consisting of a single (00) byte;

— the second directory record of the directory shall describe the parent directory for that directory and shall have a directory identifier consisting of a single (01) byte;

— a directory record in its parent directory shall describe the directory.

7.8.3 Relation of directory hierarchies

One or more directory hierarchies shall be recorded on a volume.

A directory hierarchy shall be identified in the primary volume descriptor.

Each additional directory hierarchy shall be identified in a supplementary volume descriptor or an enhanced volume descriptor.

The directories within each hierarchy shall identify zero or more of the files that are recorded in those volumes, the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume.

A directory shall not be a part of more than one directory hierarchy.

7.9 Path table

7.9.1 General

A path table recorded on a volume of a volume set shall contain a set of records describing a directory hierarchy for those volumes of the volume set for which the sequence numbers are less than, or equal to, the assigned volume set size of the volume.

For each directory in the directory hierarchy other than the root directory, the path table shall contain a record which identifies the directory, its parent directory and its location. The records in a path table shall be numbered starting from 1. The first record in the path table shall identify the root directory and its location.

The directory number of a directory shall be the ordinal number of the path table record that identifies the directory.

7.9.2 Order of path table records

The records in a path table shall be ordered by the following criteria in descending order of significance:

— in ascending order according to the level in the directory hierarchy;

— in ascending order according to the directory number of the parent directory of the directory identified by the record;

— in ascending order according to the relative value of the directory identifier field in the record, where the directory identifiers shall be valued as follows.

- If two directory identifiers do not contain the same number of byte positions, the shorter directory identifier shall be treated as if it were padded on the right with all padding bytes set to filler (see 8.4.3.2), and as if both directory identifiers contained the identical number of byte positions;

- After any padding necessary to treat the directory identifiers as if they were of equal length, the characters in the corresponding byte positions (starting with the first position) of the directory identifiers are compared until a byte position is found that does not contain the same character in both directory identifiers. The greater directory identifier is the one that contains the character with the higher code position value in the coded graphic character sets used to interpret the directory identifier of the path table record.

7.9.3 Path table group

A path table shall be either a type L path table or a type M path table.

In a type L path table, a numerical value shall be recorded according to 8.2.2 if represented as a 16-bit number and according to 8.3.2 if represented as a 32-bit number.

In a type M path table, a numerical value shall be recorded according to 8.2.3 if represented as a 16-bit number and according to 8.3.3 if represented as a 32-bit number.

A path table group shall comprise one or two identical type L path tables and one or two identical type M path tables.

7.9.4 Recorded occurrences of the path table

One or more path table groups shall be recorded on a volume. The primary volume descriptor shall identify the size and locations of the constituent path tables of a path table group. These path tables shall identify the directories in the directory hierarchy which is identified by the primary volume descriptor.

For each additional directory hierarchy recorded on a volume, an additional path table group shall be recorded on the volume. For each such path table group, the corresponding supplementary volume descriptor shall identify the size and locations of its constituent path tables. These path tables shall identify the directories in the corresponding directory hierarchy.

7.9.5 Consistency of path tables between volumes of a volume group

7.9.5.1 Contents of path tables identified in a primary volume descriptor

The contents of a type L path table identified in a primary volume descriptor shall be identical to the contents of any other type L path table identified in a primary volume descriptor on a volume of the same volume group.

The contents of a type M path table identified in a primary volume descriptor shall be identical to the contents of any other type M path table identified in a primary volume descriptor on a volume of the same volume group.

7.9.5.2 Contents of path tables identified in a supplementary volume descriptor

The contents of a type L path table identified in a supplementary volume descriptor shall be identical to the contents of any other type L path table identified in a supplementary volume descriptor, having the same volume set identification and identifying the same coded graphic character sets for use within selected descriptor fields (see 8.4), on a volume of the same volume group.

The contents of a type M path table identified in a supplementary volume descriptor shall be identical to the contents of any other type M path table identified in a supplementary volume descriptor, having the same volume set identification and identifying the same coded graphic character sets for use within selected descriptor fields (see 8.4), on a volume of the same volume group.

7.10 Record structure

7.10.1 General

The information in a file may be organized as a set of records according to 7.10.

7.10.2 Characteristics

A record shall be a sequence of bytes treated as a unit of information.

The length of a record shall be the number of bytes in the record.

A record shall be either a fixed-length record or a variable-length record.

All records in a file shall be either fixed-length records or variable-length records.

7.10.3 Measured data units (MDU)

7.10.3.1 Contents of an MDU

A measured data unit shall contain either a fixed-length record or a variable-length record. An MDU shall comprise an even number of bytes.

7.10.3.2 Relationship to file

Each MDU shall comprise a set of successive bytes of the file. The first or only MDU shall begin at the first byte of the file. Each successive MDU shall begin at the byte in the file immediately following the last byte of the preceding MDU.

7.10.4 Fixed-length records

A fixed-length record shall be a record contained in a file that is assigned to contain records that must have the same length.

A fixed-length record shall be contained in an MDU. The MDU shall consist of the fixed-length record, immediately followed by a (00) byte if necessary to give the MDU an even length.

The minimum assigned length of a fixed-length record shall be 1.

7.10.5 Variable-length records

A variable-length record shall be a record contained in a file that is assigned to contain records that may have different lengths. The value recorded in the record format field of an extended attribute record for a file containing variable-length records shall contain the same value as that recorded in the record format field of any other extended attribute record of that same file.

A variable-length record shall be contained in an MDU. The MDU shall consist of a record control word (RCW) immediately followed by the variable-length record, immediately followed by a (00) byte if necessary to give the MDU an even length.

The RCW shall specify as a 16-bit number the length of the record. The RCW shall be recorded according to:

— 8.2.2, if the value in the record format field of the extended attribute record associated with the file section is 2; or

— 8.2.3, if the value in the record format field of the extended attribute record associated with the file section is 3.

A maximum record length shall be assigned for a file. The length of any record in the file shall not exceed this value. The assigned maximum record length shall be in the range 1 to 32 767.

The minimum length of a variable-length record shall be 0.

8 Recording of descriptor fields

8.1 8-bit numerical values

8.1.1 General

A numerical value represented in binary notation by an 8-bit number shall be recorded in a field of a descriptor in one of the following two formats. The applicable format is specified in the description of the descriptor fields.

8.1.2 8-bit unsigned numerical values

An unsigned numerical value shall be represented in binary notation by an 8-bit number recorded in a one-byte field.

8.1.3 8-bit signed numerical values

A signed numerical value shall be represented in binary notation by an 8-bit two's complement number recorded in a one-byte field.

8.2 16-bit numerical value

8.2.1 General

A numerical value represented in binary notation by a 16-bit number shall be recorded in a field of a descriptor in one of the following three formats. The applicable format is specified in the description of the descriptor fields.

8.2.2 Least significant byte first

A numerical value represented by the hexadecimal representation (*wx yz*) shall be recorded in a two-byte field as (*yz wx*).

EXAMPLE The decimal number 4 660 has (12 34) as its hexadecimal representation and is recorded as (34 12).

8.2.3 Most significant byte first

A numerical value represented by the hexadecimal representation (*wx yz*) shall be recorded in a two-byte field as (*wx yz*).

EXAMPLE The decimal number 4 660 has (12 34) as its hexadecimal representation and is recorded as (12 34).

8.2.4 Both-byte orders

A numerical value represented by the hexadecimal representation (*wx yz*) shall be recorded in a four-byte field as (*yz wx wx yz*).

EXAMPLE The decimal number 4 660 has (12 34) as its hexadecimal representation and is recorded as (34 12 12 34).

8.3 32-bit numerical value

8.3.1 General

A numerical value represented in binary notation by a 32-bit number shall be recorded in a field of a descriptor in one of the following three formats. The applicable format is specified in the description of the descriptor fields.

8.3.2 Least significant byte first

A numerical value represented by the hexadecimal representation (*st uv wx yz*) shall be recorded in a four-byte field as (*yz wx uv st*).

EXAMPLE The decimal number 305 419 896 has (12 34 56 78) as its hexadecimal representation and is recorded as (78 56 34 12).

8.3.3 Most significant byte first

A numerical value represented by the hexadecimal representation (*st uv wx yz*) shall be recorded in a four-byte field as (*st uv wx yz*).

EXAMPLE The decimal number 305 419 896 has (12 34 56 78) as its hexadecimal representation and is recorded as (12 34 56 78).

8.3.4 Both-byte orders

A numerical value represented by the hexadecimal representation (*st uv wx yz*) shall be recorded in an eight-byte field as (*yz wx uv st st uv wx yz*).

EXAMPLE The decimal number 305 419 896 has (12 34 56 78) as its hexadecimal representation and is recorded as (78 56 34 12 12 34 56 78).

8.4 Character sets and coding

8.4.1 d-characters and a-characters

Except as specified in 8.4.4, the characters in the descriptors shall be coded according to ECMA-6 (see Annex A).

The 37 characters in the following positions of the International Reference Version (IRV) are referred to as d-characters (See A.2 for further details):

3/0 to 3/9

4/1 to 5/10

5/15

The 57 characters in the following positions of the IRV are referred to as a-characters (See A.3 for further details):

2/0 to 2/2

2/5 to 2/15

3/0 to 3/15

4/1 to 4/15

5/0 to 5/10

5/15

The applicable set of characters is specified in the description of the descriptor fields.

8.4.2 c-characters

8.4.2.1 General

The characters of the coded graphic character sets identified by the escape sequences in a supplementary volume descriptor are referred to as c-characters.

8.4.2.2 a1-characters

A subset of the c-characters is referred to as a1-characters. This subset shall be subject to agreement between the originator and the recipient of the volume.

8.4.2.3 d1-characters

A subset of the a1-characters is referred to as d1-characters. This subset shall be subject to agreement between the originator and the recipient of the volume.

8.4.3 Separators and filler

8.4.3.1 Separators

The characters separating the components of a file identifier within a directory hierarchy that is identified in a primary volume descriptor or in a supplementary volume descriptor shall be:

- separator 1 represented by the bit combination (2E);
- separator 2 represented by the bit combination (3B).

A file identifier within a directory hierarchy that is identified in an enhanced volume descriptor shall not be separated into components.

Separators are not specified for a file identifier within a directory hierarchy that is identified in an enhanced volume descriptor.

8.4.3.2 Filler

The character filling any byte position which is specified by this Ecma Standard to be characters shall be referred to as filler.

Within a volume that is identified by a primary volume descriptor or by a supplementary volume descriptor, the bit combination of filler shall be (20).

Within a volume that is identified by an enhanced volume descriptor, the bit combination of filler shall be subject to agreement between the originator and the recipient of the volume.

8.4.4 Use of characters in descriptor fields

The characters in the fields of the following descriptors shall be a-characters or d-characters as specified in Clause 10:

- directory records within a directory hierarchy identified in a primary volume descriptor;
- path table records within a path table group identified in a primary volume descriptor;
- extended attribute records identified in a directory of a directory hierarchy that is identified in a primary volume descriptor.

The characters in the fields of the following descriptors shall be a1-characters or d1-characters as specified in Clause 10:

- directory records within a directory hierarchy that is identified in a supplementary volume descriptor;
- path table records within a path table group identified in a supplementary volume descriptor;
- extended attribute records identified in a directory of a directory hierarchy that is identified in a supplementary volume descriptor.

The character in the fields of the following descriptors is the subject of an agreement between the originator and the recipient of the volume:

- directory records within a directory hierarchy that is identified in an enhanced volume descriptor;
- path table records within a path table group identified in an enhanced volume descriptor;
- extended attribute records identified in a directory of a directory hierarchy that is identified in an enhanced volume descriptor.

8.4.5 Justification of characters

In each fixed-length field, the content of which is specified by this Ecma Standard to be characters, the characters shall be left-justified and any remaining byte positions on the right shall be set to filler.

8.5 File identifier

8.5.1 File identifier format

A file identifier shall consist of the following sequence:

— file name: a sequence of zero or more d-characters or d1-characters;

— separator 1;

— file name extension: a sequence of zero or more d-characters or d1-characters;

— separator 2;

— file version number: Digits representing a number from 1 to 32 767.

Within a directory hierarchy that is identified in a primary volume descriptor or in a supplementary volume descriptor, this sequence shall meet the following requirements:

— if no characters are specified for the file name, then the file name extension shall consist of at least one character;

— if no characters are specified for the file name extension, then the file name shall consist of at least one character;

— the sum of the following shall not exceed 30:

  — if there is a file name, the length of the file name,

  — if there is a file name extension, the length of the file name extension.

Within a directory hierarchy that is identified in an enhanced volume descriptor, this sequence shall meet the following requirements:

— the length of file name shall not exceed 207;

— no separators are specified;

— no file name extension is present;

— no file version number is present.

Within a directory hierarchy that is identified in an enhanced volume descriptor, a file identifier shall not be specified as certain character sequences. These sequences shall be subject to agreement between the originator and recipient of the volume.

NOTE 207 (the maximum length of the file name) is 254 (the maximum directory record length), subtracted by 33 (the minimum length of directory record excluding the file identifier) and by 14 (the CD-ROM XA system use extension information). See Reference [3].

8.5.2 File identifier length

The length of the file identifier shall be the sum of the following:

— if there is a file name, the length of the file name;

— if there is a file name extension, the length of the file name extension;

- if there is a file version number, the number of digits in the file version number;

- if separators are specified, 2 (the number of separators).

8.6 Directory identifier

8.6.1 Directory identifier format

A directory identifier shall consist of a sequence of one or more d-characters or d1-characters (see 8.4.4), except as specified in 8.6.2.

Within a volume identified by an enhanced volume descriptor, a directory identifier can be a sequence of characters rather than d-characters or d1-characters. The sequence shall be subject to agreement between the originator and recipient of the volume.

8.6.2 Reserved directory identifiers

The root directory shall be identified by a directory identifier consisting of a single (00) byte.

As specified in 7.8.2.2, certain directory identifiers shall consist of a single (00) byte or a single (01) byte.

8.6.3 Directory identifier length

Within a directory hierarchy that is identified in a primary volume descriptor or in a supplementary volume descriptor, the length of a directory identifier shall not exceed 31.

Within a directory hierarchy that is identified in an enhanced volume descriptor, the length of a directory identifier shall not exceed 207.

9 Volume descriptors

9.1 Format of a volume descriptor

9.1.1 General

The volume descriptors shall identify the volume, the partitions recorded on the volume, the volume creator(s), certain attributes of the volume, the location of other recorded descriptors and the version of the standard which applies to the volume descriptor.

Details of the format of a volume descriptor are given in Table 1.

Table 1 — Volume descriptor

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8 to 2 048	(Depends on volume descriptor type)	(Depends on volume descriptor type)

9.1.2 Volume descriptor type (BP 1)

This field shall specify as an 8-bit number the volume descriptor type:

- number 0 shall mean that the volume descriptor is a boot record;
- number 1 shall mean that the volume descriptor is a primary volume descriptor;
- number 2 shall mean that the volume descriptor is a supplementary volume descriptor or an enhanced volume descriptor;
- number 3 shall mean that the volume descriptor is a volume partition descriptor;
- numbers 4 to 254 are reserved for future standardization;
- number 255 shall mean that the volume descriptor is a volume descriptor set terminator.

This field shall be recorded according to 8.1.2.

9.1.3 Standard identifier (BP 2 to 6)

This field shall specify an identification of this Ecma Standard.

The characters in this field shall be CD001.

9.1.4 Volume descriptor version (BP 7)

This field shall specify as an 8-bit number the version of the specification of the volume descriptor.

The content and the interpretation of this field shall depend on the content of the volume descriptor type field.

This field shall be recorded according to 8.1.2.

9.1.5 Depends on volume descriptor type (BP 8 to 2 048)

The content and the interpretation of this field shall depend on the content of the volume descriptor type field.

9.2 Boot record

9.2.1 General

The boot record shall identify a system which can recognize and act upon the content of the field reserved for boot system use in the boot record, and shall contain information which is used to achieve a specific state for a system or for an application, as shown in Table 2.

Table 2 — Boot record

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8 to 39	Boot system identifier	a-characters
40 to 71	Boot identifier	a-characters
72 to 2 048	Boot system use	Not specified

9.2.2 Volume descriptor type (BP 1)

This field shall specify an 8-bit number indicating that the volume descriptor is a boot record.

The number in this field shall be 0.

This field shall be recorded according to 8.1.2.

9.2.3 Standard identifier (BP 2 to 6)

This field shall specify an identification of this Ecma Standard.

The characters in this field shall be CD00l.

9.2.4 Volume descriptor version (BP 7)

This field shall specify as an 8-bit number the version of the specification of the boot record structure.

1 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.2.5 Boot system identifier (BP 8 to 39)

This field shall specify an identification of a system which can recognize and act upon the content of the boot identifier and boot system use fields in the boot record.

The characters in this field shall be a-characters.

9.2.6 Boot identifier (BP 40 to 71)

This field shall specify an identification of the boot system specified in the boot system use field of the boot record.

The characters in this field shall be a-characters.

9.2.7 Boot system use (BP 72 to 2 048)

This field shall be reserved for boot system use. Its contents are not specified by this Ecma Standard.

9.3 Volume descriptor set terminator

9.3.1 General

The recorded set of volume descriptors shall be terminated by a sequence of one or more volume descriptor set terminators as shown in Table 3.

Table 3 — Volume descriptor set terminator

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8 to 2 048	(Reserved for future standardization)	(00) bytes

9.3.2 Volume descriptor type (BP 1)

This field shall specify an 8-bit number indicating that the volume descriptor is a volume descriptor set terminator.

The number in this field shall be 255.

This field shall be recorded according to 8.1.2.

9.3.3 Standard identifier (BP 2 to 6)

This field shall specify an identification of this Ecma Standard.

The characters in this field shall be CD00l.

9.3.4 Volume descriptor version (BP 7)

This field shall specify as an 8-bit number the version of the specification of the volume descriptor set terminator.

1 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.3.5 Reserved for future standardization (BP 8 to 2 048)

All bytes of this field shall be set to (00).

9.4 Primary volume descriptor

9.4.1 General

The primary volume descriptor shall identify the volume, a system which can recognize and act upon the content of the logical sectors with logical sector numbers 0 to 15, the size of the volume space, the version of the standard which applies to the volume descriptor, the version of the specification which applies to the directory records and the path table records and certain attributes of the volume, as shown in Table 4.

Table 4 — Primary volume descriptor

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8	Unused field	(00) byte
9 to 40	System identifier	a-characters
41 to 72	Volume identifier	d-characters
73 to 80	Unused field	(00) byte
81 to 88	Volume space size	Numerical value
89 to 120	Unused field	(00) bytes
121 to 124	Volume set size	Numerical value
125 to 128	Volume sequence number	Numerical value
129 to 132	Logical block size	Numerical value
133 to 140	Path table size	Numerical value
141 to 144	Location of occurrence of type L path table	Numerical value
145 to 148	Location of optional occurrence of type L path table	Numerical value
149 to 152	Location of occurrence of type M path table	Numerical value
153 to 156	Location of optional occurrence of type M path table	Numerical value
157 to 190	Directory record for root directory	34 bytes
191 to 318	Volume set identifier	d-characters
319 to 446	Publisher identifier	a-characters
447 to 574	Data preparer identifier	a-characters
575 to 702	Application identifier	a-characters
703 to 739	Copyright file identifier	d-characters, separator 1, separator 2
740 to 776	Abstract file identifier	d-characters, separator 1, separator 2
777 to 813	Bibliographic file identifier	d-characters, separator 1, separator 2
814 to 830	Volume creation date and time	Digit(s), numerical value
831 to 847	Volume modification date and time	Digit(s), numerical value
848 to 864	Volume expiration date and time	Digit(s), numerical value
865 to 881	Volume effective date and time	Digit(s), numerical value

Table 4 — Primary volume descriptor (concluded)

BP	Field name	Content
882	File structure version	numerical value
883	(Reserved for future standardization)	(00) byte
884 to 1 395	Application use	not specified
1 396 to 2 048	(Reserved for future standardization)	(00) bytes

9.4.2 Volume descriptor type (BP 1)

This field shall specify an 8-bit number indicating that the volume descriptor is a primary volume descriptor.

The number in this field shall be 1.

This field shall be recorded according to 8.1.2.

9.4.3 Standard identifier (BP 2 to 6)

This field shall specify an identification of this Ecma Standard.

The characters in this field shall be CD00l.

9.4.4 Volume descriptor version (BP 7)

This field shall specify as an 8-bit number an identification of the version of the specification of the primary volume descriptor.

1 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.4.5 Unused field (BP 8)

This field shall be set to (00).

9.4.6 System identifier (BP 9 to 40)

This field shall specify an identification of a system which can recognize and act upon the content of the logical sectors with logical sector numbers 0 to 15 of the volume.

The characters in this field shall be a-characters.

9.4.7 Volume identifier (BP 41 to 72)

This field shall specify an identification of the volume.

The characters in this field shall be d-characters.

9.4.8 Unused field (BP 73 to 80)

All bytes of this field shall be set to (00).

9.4.9 Volume space size (BP 81 to 88)

This field shall specify as a 32-bit number the number of logical blocks in which the volume space of the volume is recorded.

This field shall be recorded according to 8.3.4.

9.4.10 Unused field (BP 89 to 120)

All bytes of this field shall be set to (00).

9.4.11 Volume set size (BP 121 to 124)

This field shall specify as a 16-bit number the assigned volume set size of the volume.

This field shall be recorded according to 8.2.4.

9.4.12 Volume sequence number (BP 125 to 128)

This field shall specify as a 16-bit number the ordinal number of the volume in the volume set of which the volume is a member.

This field shall be recorded according to 8.2.4.

9.4.13 Logical block size (BP 129 to 132)

This field shall specify as a 16-bit number the size in bytes of a logical block.

This field shall be recorded according to 8.2.4.

9.4.14 Path table size (BP 133 to 140)

This field shall specify as a 32-bit number the length in bytes of a recorded occurrence of the path table identified by this volume descriptor.

This field shall be recorded according to 8.3.4.

9.4.15 Location of occurrence of type L path table (BP 141 to 144)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an occurrence of the path table. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the least significant byte first.

This field shall be recorded according to 8.3.2.

9.4.16 Location of optional occurrence of type L path table (BP 145 to 148)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an optional occurrence of the path table. If the value is 0, it shall mean that the extent shall not be expected to have been recorded. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the least significant byte first.

This field shall be recorded according to 8.3.2.

9.4.17 Location of occurrence of type M path table (BP 149 to 152)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an occurrence of the path table. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the most significant byte first.

This field shall be recorded according to 8.3.3.

9.4.18 Location of optional occurrence of type M path table (BP 153 to 156)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an optional occurrence of the path table. If the value is 0, it shall mean that the extent shall not be expected to have been recorded. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the most significant byte first.

This field shall be recorded according to 8.3.3.

9.4.19 Directory record for root directory (BP 157 to 190)

This field shall contain an occurrence of the directory record for the root directory.

This field shall be recorded according to 10.1.

9.4.20 Volume set identifier (BP 191 to 318)

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

The characters in this field shall be d-characters.

9.4.21 Publisher identifier (BP 319 to 446)

This field shall specify an identification of the user who specified what shall be recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the user. This file shall be described in the root directory. The file name shall not contain more than 8 d-characters and the file name extension shall not contain more than 3 d-characters.

If all bytes of this field are set to filler, it shall mean that no such user is identified.

The characters in this field shall be a-characters.

9.4.22 Data preparer identifier (BP 447 to 574)

This field shall specify an identification of the person or other entity which controls the preparation of the data to be recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the data preparer. This file shall be described in the root directory. The file name shall not contain more than 8 d-characters and the file name extension shall not contain more than 3 d-characters.

If all bytes of this field are set to filler, it shall mean that no such data preparer is identified.

The characters in this field shall be a-characters.

9.4.23 Application identifier (BP 575 to 702)

This field shall specify an identification of the specification of how the data are recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the application. This file shall be described in the root directory. The file name shall not contain more than 8 d-characters and the file name extension shall not contain more than 3 d-characters.

If all bytes of this field are set to filler, it shall mean that no such application is identified.

The characters in this field shall be a-characters.

9.4.24 Copyright file identifier (BP 703 to 739)

This field shall specify an identification for a file described by the root directory and containing a copyright statement for those volumes of the volume set the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

The file name of a copyright file identifier shall not contain more than 8 d-characters. The file name extension of a copyright file identifier shall not contain more than 3 d-characters.

The characters in this field shall be d-characters, separator 1 and separator 2.

This field shall be recorded as specified in 8.5.

9.4.25 Abstract file identifier (BP 740 to 776)

This field shall specify an identification for a file described by the root directory and containing an abstract statement for those volumes of the volume set the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

The file name of an abstract file identifier shall not contain more than 8 d-characters. The file name extension of an abstract file identifier shall not contain more than 3 d-characters.

The characters in this field shall be d-characters, separator 1 and separator 2.

This field shall be recorded as specified in 8.5.

9.4.26 Bibliographic file identifier (BP 777 to 813)

This field shall specify an identification for a file described by the root directory and containing bibliographic records interpreted according to standards that are the subject of an agreement between the originator and the recipient of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

The file name of a bibliographic file identifier shall not contain more than 8 d-characters. The file name extension of a bibliographic file identifier shall not contain more than 3 d-characters.

The characters in this field shall be d-characters, separator 1 and separator 2.

This field shall be recorded as specified in 8.5.

9.4.27 Volume creation date and time (BP 814 to 830)

9.4.27.1 General

This field shall specify the date and the time of day at which the information in the volume was created. It shall be recorded according to 9.4.27.2.

9.4.27.2 Date and time format

The date and time shall be represented by a 17-byte field recorded as shown in Table 5.

Table 5 — Date and time format

RBP	Interpretation	Content
1 to 4	Year from 1 to 9999	Digits
5 to 6	Month of the year from 1 to 12	Digits
7 to 8	Day of the month from 1 to 31	Digits
9 to 10	Hour of the day from 0 to 23	Digits
11 to 12	Minute of the hour from 0 to 59	Digits
13 to 14	Second of the minute from 0 to 59	Digits
15 to 16	Hundredths of a second	Digits
17	Offset from Greenwich Mean Time in number of 15 min intervals from -48 (west) to +52 (east) recorded according to 8.1.3	Numerical value

If all characters in RBP 1 to 16 of this field are the digit ZERO and the number in RBP 17 is zero, it shall mean that the date and time are not specified.

9.4.28 Volume modification date and time (BP 831 to 847)

This field shall specify the date and the time of the day at which the information in the volume was last modified.

This field shall be recorded according to 9.4.27.2.

9.4.29 Volume expiration date and time (BP 848 to 864)

This field shall specify the date and the time of day at which the information in the volume may be regarded as obsolete. If the date and time are not specified, then the information shall not be regarded as obsolete.

This field shall be recorded according to 9.4.27.2.

9.4.30 Volume effective date and time (BP 865 to 881)

This field shall specify the date and the time of day at which the information in the volume may be used. If the date and time are not specified, then the information may be used at once.

This field shall be recorded according to 9.4.27.2.

9.4.31 File structure version (BP 882)

This field shall specify as an 8-bit number the version of the specification of the records of a directory and of a path table.

For a primary volume descriptor or for a supplementary volume descriptor, 1 shall indicate the structure of this Ecma Standard. For an enhanced volume descriptor, 2 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.4.32 Reserved for future standardization (BP 883)

This field shall be set to (00).

9.4.33 Application use (BP 884 to 1 395)

This field shall be reserved for application use. Its content is not specified by this Ecma Standard.

9.4.34 Reserved for future standardization (BP 1 396 to 2 048)

All bytes of this field shall be set to (00).

9.5 Supplementary volume descriptor and enhanced volume descriptor

9.5.1 General

The supplementary volume descriptor or the enhanced volume descriptor shall identify:

— the volume;

— a system which can recognize and act upon the content of the logical sectors with logical sector numbers 0 to 15;

— the size of the volume space;

— the version of the standard which applies to the volume descriptor;

— the version of the specification which applies to the directory records and the path table records;

— certain attributes of the volume;

— the coded graphic character sets used to interpret descriptor fields that contain characters, as shown in Table 6.

Within a volume descriptor set the contents of the fields of this descriptor shall be identical to the contents of the corresponding fields in a primary volume descriptor except for the following fields.

Table 6 — Supplementary volume descriptor and enhanced volume descriptor

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8	Volume flags	8 bits
9 to 40	System identifier	a1-characters*
41 to 72	Volume identifier	d1-characters*
73 to 80	Unused field	(00) byte

Table 6 — Supplementary volume descriptor and enhanced volume descriptor (concluded)

BP	Field name	Content
81 to 88	Volume space size	Numerical value
89 to 120	Escape sequences	32 bytes
121 to 124	Volume set size	Numerical value
125 to 128	Volume sequence number	Numerical value
129 to 132	Logical block size	Numerical value
133 to 140	Path table size	Numerical value
141 to 144	Location of occurrence of type L path table	Numerical value
145 to 148	Location of optional occurrence of type L path table	Numerical value
149 to 152	Location of occurrence of type M path table	Numerical value
153 to 156	Location of optional occurrence of type M path table	Numerical value
157 to 190	Directory record for root directory	34 bytes
191 to 318	Volume set identifier	d1-characters*
319 to 446	Publisher identifier	a1-characters*
447 to 574	Data preparer identifier	a1-characters*
575 to 702	Application identifier	a1-characters*
703 to 739	Copyright file identifier	d1-characters, separator 1, separator 2*
740 to 776	Abstract file identifier	d1-characters, separator 1, separator 2*
777 to 813	Bibliographic file identifier	d1-characters, separator 1, separator 2*
814 to 830	Volume creation date and time	Digit(s), numerical value
831 to 847	Volume modification date and time	Digit(s), numerical value
848 to 864	Volume expiration date and time	Digit(s), numerical value
865 to 881	Volume effective date and time	Digit(s), numerical value
882	File structure version	Numerical value
883	(Reserved for future standardization)	(00) byte
884 to 1 395	Application use	Not specified
1 396 to 2 048	(Reserved for future standardization)	(00) bytes
NOTE '*' denotes case of supplementary volume descriptor.		

9.5.2 Volume descriptor type (BP 1)

This field shall specify an 8-bit number indicating that the volume descriptor is a supplementary volume descriptor or is an enhanced volume descriptor.

The number in this field shall be 2.

This field shall be recorded according to 8.1.2.

9.5.3 Volume descriptor version (BP 7)

This field shall specify an 8-bit number indicating that the volume descriptor is a supplementary volume descriptor or is an enhanced volume descriptor.

For a supplementary volume descriptor, 1 shall indicate the structure of this Ecma Standard. For an enhanced volume descriptor, 2 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.5.4 Volume flags (BP 8)

The bits of this field shall be numbered from 0 to 7 starting with the least significant bit.

This field shall specify certain characteristics of the volume as follows.

Bit 0:

— if set to ZERO, shall mean that the escape sequences field specifies only escape sequences registered according to ISO/IEC 2375;

— if set to ONE, shall mean that the escape sequences field specifies at least one escape sequence not registered according to ISO/IEC 2375.

Bits 1 to 7: These bits are reserved for future standardization and shall all be set to ZERO.

9.5.5 System identifier (BP 9 to 40)

This field shall specify an identification of a system which can recognize and act upon the content of the logical sectors with logical sector numbers 0 to 15 of the volume.

Within a supplementary volume descriptor, the characters in this field shall be a1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.6 Volume identifier (BP 41 to 72)

This field shall specify an identification of the volume.

Within a supplementary volume descriptor, the characters in this field shall be d1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.7 Escape sequences (BP 89 to 120)

This field shall specify one or more escape sequences according to ECMA-35 that designate the GO graphic character set and, optionally, the G1 graphic character set to be used in an 8-bit environment according to ECMA-35 to interpret descriptor fields related to the directory hierarchy identified by this volume descriptor (see 8.4.4). If the G1 set is designated, it is implicitly invoked into columns 10 to 15 of the code table.

These escape sequences shall conform to ECMA-35, except that the escape character shall be omitted from each designating escape sequence when recorded in this field. The first or only escape sequence shall begin at the first byte of the field. Each successive escape sequence shall begin at the byte in the field immediately following the last byte of the preceding escape sequence. Any unused byte positions following the last sequence shall be set to (00).

If all the bytes of this field are set to (00), it shall mean that the set of a1-characters is identical to the set of a-characters and that the set of d1-characters is identical to the set of d-characters. In this case both sets are coded according to ECMA-6.

9.5.8 Path table size (BP 133 to 140)

This field shall specify as a 32-bit number the length in bytes of a recorded occurrence of the path table identified by this volume descriptor.

This field shall be recorded according to 8.3.4.

9.5.9 Location of occurrence of type L path table (BP 141 to 144)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an occurrence of the path table. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the least significant byte first.

This field shall be recorded according to 8.3.2.

9.5.10 Location of optional occurrence of type L path table (BP 145 to 148)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an optional occurrence of the path table. If the value is 0, it shall mean that the extent shall not be expected to have been recorded. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the least significant byte first.

This field shall be recorded according to 8.3.2.

9.5.11 Location of occurrence of type M path table (BP 149 to 152)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an occurrence of the path table. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the most significant byte first.

This field shall be recorded according to 8.3.3.

9.5.12 Location of optional occurrence of type M path table (BP 153 to 156)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent which contains an optional occurrence of the path table. If the value is 0, it shall mean that the extent shall not be expected to have been recorded. Multiple-byte numerical values in a record of this occurrence of the path table shall be recorded with the most significant byte first.

This field shall be recorded according to 8.3.3.

9.5.13 Directory record for root directory (BP 157 to 190)

This field shall contain an occurrence of the directory record for the root directory.

This field shall be recorded according to 10.1.

9.5.14 Volume set identifier (BP 191 to 318)

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

Within a supplementary volume descriptor, the characters in this field shall be d1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.15 Publisher identifier (BP 319 to 446)

This field shall specify an identification of the user who specified what shall be recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the user. This file shall be described in the root directory.

If all bytes of this field are set to filler, it shall mean that no such user is identified.

Within a supplementary volume descriptor, the characters in this field shall be a1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.16 Data preparer identifier (BP 447 to 574)

This field shall specify an identification of the person or other entity which controls the preparation of the data to be recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the data preparer. This file shall be described in the root directory.

If all bytes of this field are set to filler, it shall mean that no such data preparer is identified.

Within a supplementary volume descriptor, the characters in this field shall be a1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.17 Application identifier (BP 575 to 702)

This field shall specify an identification of the specification of how the data are recorded on the volume group of which the volume is a member.

If the first byte is set to (5F), the remaining bytes of this field shall specify an identifier for a file containing the identification of the application. This file shall be described in the root directory.

If all bytes of this field are set to filler, it shall mean that no such application is identified.

Within a supplementary volume descriptor, the characters in this field shall be a1-characters.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

9.5.18 Copyright file identifier (BP 703 to 739)

This field shall specify an identification for a file described by the root directory and containing a copyright statement for those volumes of the volume set the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

Within a supplementary volume descriptor, the characters in this field shall be d1-characters, separator 1 and separator 2.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

The field shall be recorded as specified in 8.5.

9.5.19 Abstract file identifier (BP 740 to 776)

This field shall specify an identification for a file described by the root directory and containing an abstract statement for those volumes of the volume set the sequence numbers of which are less than, or equal to, the assigned volume set size of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

Within a supplementary volume descriptor, the characters in this field shall be d1-characters, separator 1 and separator 2.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

The field shall be recorded as specified in 8.5.

9.5.20 Bibliographic file identifier (BP 777 to 813)

This field shall specify an identification for a file described by the root directory and containing bibliographic records interpreted according to standards that are the subject of an agreement between the originator and the recipient of the volume. If all bytes of this field are set to filler, it shall mean that no such file is identified.

Within a supplementary volume descriptor, the characters in this field shall be d1-characters, separator 1 and separator 2.

Within an enhanced volume descriptor, the characters in this field shall be subject to agreement between the originator and recipient of the volume.

The field shall be recorded as specified in 8.5.

9.5.21 Application use (BP 884 to 1 395)

This field shall be reserved for application use. Its content is not specified by this Ecma Standard.

9.6 Volume partition descriptor

9.6.1 General

The volume partition descriptor shall identify a volume partition within the volume space, a system which can recognize and act upon the content of fields reserved for system use in the volume descriptor, the position and size of the volume partition, and the version of the standard which applies to the volume descriptor, as shown in Table 7. The contents of the volume partition are not specified by this Ecma Standard.

Table 7 — Volume partition descriptor

BP	Field name	Content
1	Volume descriptor type	Numerical value
2 to 6	Standard identifier	CD001
7	Volume descriptor version	Numerical value
8	Unused field	(00) byte
9 to 40	System identifier	a-characters
41 to 72	Volume partition identifier	d-characters
73 to 80	Volume partition location	Numerical value
81 to 88	Volume partition size	Numerical value
89 to 2 048	System use	Not specified

9.6.2 Volume descriptor type (BP 1)

This field shall specify an 8-bit number indicating that the volume descriptor is a volume partition descriptor.

The number in this field shall be 3.

This field shall be recorded according to 8.1.2.

9.6.3 Standard identifier (BP 2 to 6)

This field shall specify an identification of this Ecma Standard.

The characters in this field shall be CD001.

9.6.4 Volume descriptor version (BP 7)

This field shall specify as an 8-bit number an identification of the version of the specification of the volume partition descriptor.

1 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

9.6.5 Unused field (BP 8)

This field shall be set to (00).

9.6.6 System identifier (BP 9 to 40)

This field shall specify an identification of a system which can recognize and act upon the content of the system use field in the volume descriptor.

The characters in this field shall be a-characters.

9.6.7 Volume partition identifier (BP 41 to 72)

This field shall specify an identification of the volume partition. The characters in this field shall be d-characters.

9.6.8 Volume partition location (BP 73 to 80)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the volume partition.

This field shall be recorded according to 8.3.4.

9.6.9 Volume partition size (BP 81 to 88)

This field shall specify as a 32-bit number the number of logical blocks in which the volume partition is recorded.

This field shall be recorded according to 8.3.4.

9.6.10 System use (BP 89 to 2 048)

This field shall be reserved for system use. Its content is not specified by this Ecma Standard.

10 File and directory descriptors

10.1 Format of a directory record

10.1.1 General

Details of the format of a directory record are given in Table 8.

Table 8 — Format of a directory record

BP	Field name	Content
1	Length of directory record (LEN_DR)	Numerical value
2	Extended attribute record length	Numerical value
3 to 10	Location of extent	Numerical value
11 to 18	Data length	Numerical value
19 to 25	Recording date and time	Numerical value
26	File flags	8 bits
27	File unit size	Numerical value
28	Interleave gap size	Numerical value
29 to 32	Volume sequence number	Numerical value
33	Length of file identifier (LEN_ FI)	Numerical value
34 to (33 + LEN_FI)	File identifier	d-characters, d1-characters, separator 1, separator 2, (00) or (01) byte
(34 + LEN_FI)	Padding field	(00) byte

Table 8 — Format of a directory record (concluded)

BP	Field name	Content
(LEN_DR – LEN_SU + 1) to LEN_DR	System use	LEN_SU bytes
NOTE LEN_SU denotes the length of the system use field.		

10.1.2 Length of directory record (LEN_DR) (BP 1)

This field shall specify as an 8-bit number the length in bytes of the directory record.

This field shall be recorded according to 8.1.2.

10.1.3 Extended attribute record length (BP 2)

This field shall contain an 8-bit number. This number shall specify the assigned extended attribute record length if an extended attribute record is recorded. Otherwise this number shall be zero.

This field shall be recorded according to 8.1.2.

10.1.4 Location of extent (BP 3 to 10)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent.

This field shall be recorded according to 8.3.4.

10.1.5 Data length (BP 11 to 18)

This field shall specify as a 32-bit number the data length of the file section.

This field shall be recorded according to 8.3.4.

NOTE This number does not include the length of any extended attribute record.

10.1.6 Recording date and time (BP 19 to 25)

This field shall indicate the date and the time of the day at which the information in the extent described by the directory record was recorded.

The date and time shall be represented by seven 8-bit numbers each of which shall be recorded according to 8.1.2 as shown in Table 9.

Table 9 — Recording date and time

RBP	Interpretation	Content
1	Number of years since 1900	Numerical value
2	Month of the year from 1 to 12	Numerical value
3	Day of the month from 1 to 31	Numerical value
4	Hour of the day from 0 to 23	Numerical value
5	Minute of the hour from 0 to 59	Numerical value

Table 9 — Recording date and time (concluded)

RBP	Interpretation	Content
6	Second of the minute from 0 to 59	Numerical value
7	Offset from Greenwich Mean Time in number of 15 min intervals from -48 (west) to +52 (east) recorded according to 8.1.3	Numerical value

If all seven numbers are zero, it shall mean that the date and time are not specified.

10.1.7 File flags (BP 26)

The bits of this field shall be numbered from 0 to 7 starting with the least significant bit.

If this directory record identifies a directory, then bit positions 2, 3 and 7 shall be set to ZERO.

If no extended attribute record is associated with the file section identified by this directory record, then bit positions 3 and 4 shall be set to ZERO.

This field shall specify certain characteristics of the file as shown in Table 10.

Table 10 — File flags

BP	Bit name	
0	Existence	If set to ZERO, this shall mean that the existence of the file shall be made known to the user upon an inquiry by the user. If set to ONE, this shall mean that the existence of the file need not be made known to the user.
1	Directory	If set to ZERO, this shall mean that the directory record does not identify a directory. If set to ONE, this shall mean that the directory record identifies a directory.
2	Associated file	If set to ZERO, this shall mean that the file is not an associated file. If set to ONE, this shall mean that the file is an associated file.
3	Record	If set to ZERO, this shall mean that the structure of the information in the file is not specified by the record format field of any associated extended attribute record (see 10.5.9). If set to ONE, this shall mean that the structure of the information in the file has a record format specified by a number other than zero in the record format field of the extended attribute record (see 10.5.9).

Table 10 — File flags (concluded)

BP	Bit name	
4	Protection	If set to ZERO, this shall mean that: — an owner identification and a group identification are not specified for the file (see 10.5.2 and 10.5.3); — any user may read or execute the file (see 10.5.4). If set to ONE, this shall mean that: — an owner identification and a group identification are specified for the file (see 10.5.2 and 10.5.3); — at least one of the even-numbered bits or bit 0 in the permissions field of the associated extended attribute record is set to ONE (see 10.5.4).
5 to 6	Reserved	These bits are reserved for future standardization and shall be set to ZERO.
7	Multi-extent	If set to ZERO, this shall mean that this is the final directory record for the file. If set to ONE, this shall mean that this is not the final directory record for the file.

10.1.8 File unit size (BP 27)

This field shall contain an 8-bit number. This number shall specify the assigned file unit size for the file section if the file section is recorded in interleaved mode. Otherwise this number shall be zero.

This field shall be recorded according to 8.1.2.

10.1.9 Interleave gap size (BP 28)

This field shall contain an 8-bit number. This number shall specify the assigned interleave gap size for the file section if the file section is recorded in interleaved mode. Otherwise this number shall be zero.

This field shall be recorded according to 8.1.2.

10.1.10 Volume sequence number (BP 29 to 32)

This field shall specify as a 16-bit number the ordinal number of the volume in the volume set on which the extent described by this directory record is recorded.

This field shall be recorded according to 8.2.4.

10.1.11 Length of file identifier (LEN_FI) (BP 33)

This field shall specify as an 8-bit number the length in bytes of the file identifier field of the directory record.

This field shall be recorded according to 8.1.2.

10.1.12 File identifier [BP 34 to (33 + LEN_FI)]

The interpretation of this field depends on the setting of the directory bit of the file flags field, as follows.

If set to ZERO, it shall mean:

- the field shall specify an identification for the file;
- the characters in this field shall be d-characters or d1-characters, separator 1, separator 2;
- the field shall be recorded as specified in 8.5.

If set to ONE, it shall mean:

- the field shall specify an identification for the directory;
- the characters in this field shall be d-characters or d1-characters, or only a (00) byte, or only a (01) byte;
- the field shall be recorded as specified in 8.6.

10.1.13 Padding field [BP (34 + LEN_FI)]

This field shall be present in the directory record only if the number in the length of the file identifier field is an even number.

If present, this field shall be set to (00).

10.1.14 System use [BP (LEN_DR - LEN_SU + 1) to LEN_DR]

This field shall be optional. If present, this field shall be reserved for system use. Its content is not specified by this Ecma Standard. If necessary, so that the directory record comprises an even number of bytes, a (00) byte shall be added to terminate this field.

10.2 Consistency of file attributes between directory records of a file

The following fields of each directory record for the same file shall contain the same values:

- existence bit of the file flags field;
- directory bit of the file flags field;
- associated file bit of the file flags field;
- record bit of the file flags field;
- reserved bits of the file flags field;
- length of file identifier field;
- file identifier field;
- padding field.

10.3 Order of directory records

The records of a directory shall be ordered according to the relative value of the file identifier field by the following criteria in descending order of significance:

a) in ascending order according to the relative value of file name, where file names shall be valued as follows:

- if two file names have the same content in all byte positions, then these two file names are said to be equal in value;

- if two file names do not contain the same number of byte positions, the shorter file name shall be treated as if it were padded on the right with all padding bytes set to filler and as if both file names contained the identical number of byte positions;

- after any padding necessary to treat the file names as if they were of equal length, the characters in the corresponding byte positions, starting with the first position, of the file names are compared until a byte position is found that does not contain the same character in both file names. The greater file name is the one that contains the character with the higher code position value in the coded graphic character sets used to interpret the file identifier field of the directory record.

b) in ascending order according to the relative value of file name extension, where file name extensions shall be valued as follows:

- if two file name extensions have the same content in all byte positions, then these two file name extensions are said to be equal in value;

- if two file name extensions do not contain the same number of byte positions, the shorter file name extension shall be treated as if it were padded on the right with all padding bytes set to filler and as if both file name extensions contained the identical number of byte positions;

- after any padding necessary to treat the file name extensions as if they were of equal length, the characters in the corresponding byte positions, starting with the first position, of the file name extensions are compared until a byte position is found that does not contain the same character in both file name extensions. The greater file name extension is the one that contains the character with the higher code position value in the coded graphic character sets used to interpret the file identifier field of the directory record.

c) in descending order according to the relative value of file version number, where file version numbers shall be valued as follows:

- if two file version numbers have the same content in all byte positions, then these two file version numbers are said to be equal in value;

- if two file version numbers do not contain the same number of byte positions, the shorter file version number shall be treated as if it were padded on the left with all padding bytes set to (30) and as if both file version numbers contained the identical number of byte positions;

- after any padding necessary to treat the file version numbers as if they were of equal length, the characters in the corresponding byte positions, starting with the first position, of the file version numbers are compared until a byte position is found that does not contain the same character in both file version numbers. The greater file version number is the one that contains the character with the higher code position value in the coded graphic character sets used to interpret the file identifier field of the directory record.

d) in descending order according to the value of the associated file bit of the file flags field;

e) the order of the file sections of the file.

10.4 Format of a path table record

10.4.1 General

Details of the format of a path table record are given in Table 11.

Table 11 — Path table record

BP	Field name	Content
1	Length of directory identifier (LEN_ DI)	Numerical value
2	Extended attribute record length	Numerical value
3 to 6	Location of extent	Numerical value
7 to 8	Parent directory number	Numerical value
9 to (8 + LEN_DI)	Directory identifier	d-characters, d1-characters, (00) byte
(9 + LEN_DI)	Padding field	(00) byte

10.4.2 Length of directory identifier (LEN_DI) (BP 1)

This field shall specify as an 8-bit number the length in bytes of the directory identifier field of the path table record.

This field shall be recorded according to 8.1.2.

10.4.3 Extended attribute record length (BP 2)

This field shall contain an 8-bit number. This number shall specify the assigned extended attribute record length if an extended attribute record is recorded. Otherwise this number shall be zero.

This field shall be recorded according to 8.1.2.

10.4.4 Location of extent (BP 3 to 6)

This field shall specify as a 32-bit number the logical block number of the first logical block allocated to the extent in which the directory is recorded.

This field shall be recorded according to 8.3.

10.4.5 Parent directory number (BP 7 to 8)

This field shall specify as a 16-bit number the record number in the path table for the parent directory of the directory.

This field shall be recorded according to 8.2.

10.4.6 Directory identifier [BP 9 to (8 + LEN_DI)]

This field shall specify an identification for a directory.

The characters in this field shall be d-characters or d1-characters or only a (00) byte.

This field shall be recorded as specified in 8.6.

10.4.7 Padding field [BP (9 + LEN_ DI)]

This field shall be present in the path table record only if the number in the length of directory identifier field is an odd number.

If present, this field shall be set to (00).

10.5 Format of an extended attribute record

10.5.1 General

If present, an extended attribute record, format details of which are given in Table 12, shall be recorded over at least one logical block.

Table 12 — Extended attribute record

BP	Field name	Content
1 to 4	Owner identification	Numerical value
5 to 8	Group identification	Numerical value
9 to 10	Permissions	16 bits
11 to 27	File creation date and time	Digit(s), numerical value
28 to 44	File modification date and time	Digit(s), numerical value
45 to 61	File expiration date and time	Digit(s), numerical value
62 to 78	File effective date and time	Digit(s), numerical value
79	Record format	Numerical value
80	Record attributes	Numerical value
81 to 84	Record length	Numerical value
85 to 116	System identifier	a-characters, a1-characters
117 to 180	System use	Not specified
181	Extended attribute record version	Numerical value
182	Length of escape sequences (LEN_ESC)	Numerical value
183 to 246	(Reserved for future standardization)	(00) bytes
247 to 250	Length of application use (LEN_AU)	Numerical value
251 to (250 + LEN_AU)	Application use	LEN_AU bytes
(251 + LEN_AU) to (250 + LEN_ESC + LEN_AU)	Escape sequences	LEN_ESC bytes

10.5.2 Owner identification (BP 1 to 4)

This field shall specify as a 16-bit number an identification of the file owner who is a member of the group identified by the group identification field of the extended attribute record.

If the number in this field is 0, this shall indicate that there is no owner identification specified for the file. In this case, the group identification field shall contain zero.

This field shall be recorded according to 8.2.4.

10.5.3 Group identification (BP 5 to 8)

This field shall specify as a 16-bit number an identification of the group of which the file owner is a member.

The values for this number, from 1 to a number subject to agreement between the data preparer and receiving system, shall identify the group as belonging to the class of user referred to as system.

If the number in this field is 0, this shall indicate that there is no group identification specified for the file. In this case, the owner identification field shall contain zero.

This field shall be recorded according to 8.2.4.

10.5.4 Permissions (BP 9 to 10)

The bits of this 16-bit field shall be numbered from 0 to 15 starting with the least significant bit of the byte recorded in byte position 10.

Bits 0 to 3 may be ignored in interchange.

If requested by the owner, bits 4 to 7 may be ignored in interchange.

This field shall specify access permission for certain classes of users as described in Table 13.

Table 13 — Permissions

Bit 0	If set to ZERO, this shall mean that an owner who is a member of a group of the system class of user may read the file. If set to ONE, this shall mean that an owner who is a member of a group of the system class of user may not read the file.
Bit 1	Shall be set to ONE.
Bit 2	If set to ZERO, this shall mean that an owner who is a member of a group of the system class of user may execute the file. If set to ONE, this shall mean that an owner who is a member of a group of the system class of user may not execute the file.
Bit 3	Shall be set to ONE.
Bit 4	If set to ZERO, this shall mean that the owner may read the file. If set to ONE, this shall mean that the owner may not read the file.
Bit 5	Shall be set to ONE.
Bit 6	If set to ZERO, this shall mean that the owner may execute the file. If set to ONE, this shall mean that the owner may not execute the file.
Bit 7	Shall be set to ONE.
Bit 8	If set to ZERO, this shall mean that any user who is a member of the group specified by the group identification field may read the file. If set to ONE, this shall mean that of the users who are members of the group specified by the group identification field, only the owner may read the file.
Bit 9	Shall be set to ONE.
Bit 10	If set to ZERO, this shall mean that any user who is a member of the group specified by the group identification field may execute the file. If set to ONE, this shall mean that of the users who are members of the group specified by the group identification field, only the owner may execute the file.
Bit 11	Shall be set to ONE.

Table 13 — Permissions (concluded)

Bit 12	If set to ZERO, this shall mean that any user may read the file. If set to ONE, this shall mean that a user not a member of the group specified by the group identification field may not read the file.
Bit 13	Shall be set to ONE.
Bit 14	If set to ZERO, this shall mean that any user may execute the file. If set to ONE, this shall mean that a user not a member of the group specified by the group identification field may not execute the file.
Bit 15	Shall be set to ONE.

10.5.5 File creation date and time (BP 11 to 27)

This field shall specify the date and the time of the day at which the information in the file was created.

This field shall be recorded according to 9.4.27.2.

10.5.6 File modification date and time (BP 28 to 44)

This field shall specify the date and the time of the day at which the information in the file was last modified.

This field shall be recorded according to 9.4.27.2.

10.5.7 File expiration date and time (BP 45 to 61)

This field shall specify the date and the time of the day at which the information in the file may be regarded as obsolete. If the date and time are not specified then the information shall not be regarded as obsolete.

This field shall be recorded according to 9.4.27.2.

10.5.8 File effective date and time (BP 62 to 78)

This field shall specify the date and the time of the day at which the information in the file may be used. If the date and time are not specified then the information may be used at once.

This field shall be recorded according to 9.4.27.2.

10.5.9 Record format (BP 79)

This field shall contain an 8-bit number specifying the format of the information in the file:

— number 0 shall mean that the structure of the information recorded in the file is not specified by this field;

— number 1 shall mean that the information in the file is a sequence of fixed-length records (see 7.10.4);

— number 2 shall mean that the information in the file is a sequence of variable-length records (see 7.10.5), in which the RCW is recorded according to 8.2.2;

— number 3 shall mean that the information in the file is a sequence of variable-length records (see 7.10.5), in which the RCW is recorded according to 8.2.3;

— numbers 4 to 127 are reserved for future standardization;

— numbers 128 to 255 are reserved for system use.

This field shall be recorded according to 8.1.2.

10.5.10 Record attributes (BP 80)

This field shall contain an 8-bit number specifying certain processing of the records in a file when they are displayed on a character-imaging device:

— number 0 shall mean that each record shall be preceded by a line feed character and followed by a carriage return character;

— number 1 shall mean that the first byte of a record shall be interpreted as specified in ISO/IEC 1539-1 for vertical spacing;

— number 2 shall mean that the record contains the necessary control information;

— numbers 3 to 255 are reserved for future standardization.

If the record format field contains zero, then the record attributes field shall be ignored in interchange.

This field shall be recorded according to 8.1.2.

10.5.11 Record length (BP 81 to 84)

This field shall specify a 16-bit number as follows:

— if the record format field contains the number 0, the record length field shall contain zero;

— if the record format field contains the number 1, the record length field shall specify the length in bytes of each record in the file;

— if the record format field contains the number 2 or 3, the record length field shall specify the maximum length in bytes of a record in the file.

This field shall be recorded according to 8.2.4.

10.5.12 System identifier (BP 85 to 116)

This field shall specify an identification of a system which can recognize and act upon the content of the system use fields in the extended attribute record and associated directory record.

The characters in this field shall be a-characters or a1-characters.

10.5.13 System use (BP 117 to 180)

This field shall be reserved for system use. Its content is not specified by this Ecma Standard.

10.5.14 Extended attribute record version (BP 181)

This field shall specify as an 8-bit number the version of the specification of the extended attribute record.

1 shall indicate the structure of this Ecma Standard.

This field shall be recorded according to 8.1.2.

10.5.15 Length of escape sequences (BP 182)

This field shall specify as an 8-bit number the length in bytes of the escape sequences field in the extended attribute record.

This field shall be recorded according to 8.1.2.

10.5.16 Reserved for future standardization (BP 183 to 246)

All bytes of this field shall be set to (00).

10.5.17 Length of application use (BP 247 to 250)

This field shall specify as a 16-bit number the length in bytes of the application use field in the extended attribute record.

This field shall be recorded according to 8.2.4.

10.5.18 Application use [BP 251 to (250 + LEN_AU)]

This field shall be reserved for application use. Its content is not specified by this Ecma Standard.

10.5.19 Escape sequences [BP (251 + LEN_AU) to (250 + LEN_ESC + LEN_AU)]

This field shall be optional. If present, this field shall contain escape sequences that designate the coded character set to be used to interpret the contents of the file. These escape sequences shall conform to ECMA-35, except that the escape character shall be omitted from each escape sequence.

The first or only escape sequence shall begin at the first byte of the field. Each successive escape sequence shall begin at the byte in the field immediately following the last byte of the preceding escape sequence. Any unused positions following the last escape sequence shall be set to (00).

10.6 Consistency of file attributes between extended attribute records of a file

The following fields of the extended attribute record associated with the file sections of a file shall contain the same values:

- record format field;
- record attributes field;
- record length field, if the records are fixed-length records (see 7.10.4).

11 Levels of interchange

11.1 General

This Ecma Standard specifies three nested levels of interchange.

11.2 Level 1

At level 1, the following restrictions shall apply to a volume identified by a primary volume descriptor or by a supplementary volume descriptor:

- each file shall consist of only one file section;
- a file name shall not contain more than 8 d-characters or 8 d1-characters;
- a file name extension shall not contain more than 3 d-characters or 3 d1-characters;

— a directory identifier shall not contain more than 8 d-characters or 8 d1-characters.

At level 1 the following restrictions shall apply to a volume identified by an enhanced volume descriptor:

— each file shall consist of only one file section.

11.3 Level 2

At level 2 the following restriction shall apply to a volume identified by a primary volume descriptor, by a supplementary volume descriptor, or by an enhanced volume descriptor:

— each file shall consist of only one file section.

11.4 Level 3

At level 3 no restrictions shall apply.

12 Requirements for systems

The volume and file structure of CD-ROM are specified from the following two perspectives:

— requirements for the medium;

— requirements for systems.

The requirements for systems are defined and specified in Clauses 13 to 15.

13 Description of systems

Certain information shall be communicated between a user and an implementation (see Clauses 14 and 15).

An information processing system that conforms to this Ecma Standard shall be the subject of a description which identifies the means by which the user may supply such information or may obtain it when it is made available, as specified in this Ecma Standard.

14 Requirements for an originating system

14.1 General

The implementation shall be capable of recording a set of files, and all descriptors that are specified in this Ecma Standard, on a volume set in accordance with one of the interchange levels specified in this Ecma Standard.

14.2 Files

The implementation shall obtain from the data preparer the information that constitutes the set of files to be recorded.

14.3 Descriptors

14.3.1 Primary volume descriptor, path table record and directory record

The implementation shall allow the data preparer to supply the information that is to be recorded in each of the descriptor fields listed below, and shall supply the information for a field if the data preparer does not supply it.

For the primary volume descriptor:

- system identifier;
- volume identifier;
- logical block size;
- location of occurrence of type L path table;
- location of optional occurrence of type L path table;
- location of occurrence of type M path table;
- location of optional occurrence of type M path table;
- volume set identifier;
- publisher identifier;
- data preparer identifier;
- application identifier;
- copyright file identifier;
- abstract file identifier;
- bibliographic file identifier;
- volume creation date and time;
- volume modification date and time;
- volume expiration date and time;
- volume effective date and time;
- application use.

For each path table record:

- extended attribute record length;
- location of extent;
- parent directory number;
- directory identifier.

For each directory record:

- extended attribute record length;
- location of extent;
- data length;
- recording date and time;
- existence bit of the file flags field;
- directory bit of the file flags field;
- associated file bit of the file flags field;
- record bit of the file flags field;
- protection bit of the file flags field;
- file unit size;
- interleave gap size;
- volume sequence number;
- file name of a file identifier;
- file name extension of a file identifier;
- file version number of a file identifier;
- system use.

14.3.2 Supplementary volume descriptor and enhanced volume descriptor

The implementation shall allow the data preparer to supply the information that is to be recorded in the descriptor fields listed below, and shall not record the supplementary volume descriptor or the enhanced volume descriptor if the data preparer does not supply the information.

For each supplementary volume descriptor or enhanced volume descriptor:

- system identifier;
- volume identifier;
- logical block size;
- location of occurrence of type L path table;
- location of optional occurrence of type L path table;
- location of occurrence of type M path table;
- location of optional occurrence of type M path table;
- bit 0 of the volume flags field;
- escape sequences;

- volume set identifier;
- publisher identifier;
- data preparer identifier;
- application identifier;
- copyright file identifier;
- abstract file identifier;
- bibliographic file identifier;
- volume creation date and time;
- volume modification date and time;
- volume expiration date and time;
- volume effective date and time;
- application use.

14.3.3 Volume partition descriptor

The implementation shall allow the data preparer to supply the information that is to be recorded in the descriptor fields listed below, and shall not record the volume partition descriptor if the data preparer does not supply the information.

For each volume partition descriptor:

- system identifier;
- volume partition identifier;
- volume partition location;
- volume partition size;
- system use.

14.3.4 Boot record

The implementation shall allow the data preparer to supply the information that is to be recorded in the descriptor fields listed below, and shall not record the boot record if the data preparer does not supply the information.

For each boot record:

- boot system identifier;
- boot identifier;
- boot system use.

14.3.5 Extended attribute record

The implementation shall allow the data preparer to supply the information that is to be recorded in the descriptor fields listed below, and need not record the extended attribute record if the data preparer does not supply the information for any of the descriptor fields listed below. If the extended attribute record is recorded, the implementation shall supply the information for a field if the data preparer does not supply it.

For each extended attribute record:

- owner identification;
- group identification;
- permissions;
- file creation date and time;
- file modification date and time;
- file expiration date and time;
- file effective date and time;
- record format;
- record attributes;
- record length;
- system identifier;
- system use;
- length of escape sequences;
- length of application use;
- application use;
- escape sequences.

14.4 System area

The implementation shall allow the data preparer to supply the information that is to be recorded on the logical sectors with logical sector numbers 0 to 15.

15 Requirements for a receiving system

15.1 General

The implementation shall be capable of reading the files and the recorded descriptors from a volume set that has been recorded in accordance with one of the interchange levels specified in this Ecma Standard, except associated files.

15.2 Files

The implementation shall make available to the user the information that constitutes the recorded files, except any associated file.

If the implementation allows the user to specify that the information constituting a file is to be interpreted according to 7.10, the implementation shall make available to the user the length of each record in the file.

15.3 Descriptors

15.3.1 Implementation

The implementation shall allow the user to supply sufficient information to enable the implementation to locate the files required by the user, and to locate the volumes on which these are recorded.

15.3.2 Descriptor information

The implementation shall make available to the user the information that is recorded in each of the descriptor fields listed below.

For the primary volume descriptor:

- volume identifier;
- volume set identifier;
- copyright file identifier;
- abstract file identifier;
- bibliographic file identifier.

For each supplementary volume descriptor or enhanced volume descriptor:

- volume identifier;
- bit 0 of the volume flags field;
- escape sequences;
- volume set identifier;
- copyright file identifier;
- abstract file identifier;
- bibliographic file identifier.

For each path table record:

- parent directory number;
- directory identifier.

For each directory record:

- file name of a file identifier;

— file name extension of a file identifier;

— directory bit of the file flags field.

15.4 Restrictions

The implementation may impose a limit on the length of a record to be made available to the user. The implementation is not required to make available to the user any byte beyond the first *n* bytes of a record, where *n* is the value of the imposed limit.

15.5 Levels of implementation

15.5.1 General

This Ecma Standard specifies two nested levels of implementation.

15.5.2 Level 1

At level 1 the implementation is permitted not to make available to the user:

— the information that constitutes the files identified in a directory hierarchy that is identified in a supplementary volume descriptor or in an enhanced volume descriptor;

— the information that is recorded in the descriptor fields of a supplementary volume descriptor or the descriptor fields of an enhanced volume descriptor and of the associated path table records, associated directory records, and extended attribute records identified by the associated directory records.

15.5.3 Level 2

At level 2 no such restrictions shall apply.

Annex A
(informative)

ECMA-6: International reference version (IRV)

A.1 General

Primarily for the interchange of information among data processing systems and associated equipment, and within data communication systems, ECMA-6 specifies a set of 128 characters (control characters and graphic characters such as letters, digits, and symbols) with their coded representation. Most of these characters are mandatory and unchangeable, but provision is made for some flexibility to accommodate national and other requirements.

ECMA-6 specifies a 7-bit coded character set with a number of options, and also provides guidance on how to exercise the options to define specific national versions and application-oriented versions.

The international reference version (IRV) is available for use when there is no requirement to use a national or an application-oriented version. In information interchange, the IRV is assumed unless an agreement exists between sender and recipient of the data.

A.2 d-characters

The d-characters are those which are not shaded in Table A.1 showing the IRV.

Table A.1 — d-characters

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	**0**	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	**1**	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	**2**	STX	DC2	”	2	B	R	b	r
0	0	1	1	**3**	ETX	DC3	#	3	C	S	c	s
0	1	0	0	**4**	EOT	DC4	$	4	D	T	d	t
0	1	0	1	**5**	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	**6**	ACK	SYN	&	6	F	V	f	v
0	1	1	1	**7**	BEL	ETB	’	7	G	W	g	w
1	0	0	0	**8**	BS	CAN	(	8	H	X	h	x
1	0	0	1	**9**	HT	EM	)	9	I	Y	i	y
1	0	1	0	**10**	LF	SUB	*	:	J	Z	j	z
1	0	1	1	**11**	VT	ESC	+	;	K	[	k	{
1	1	0	0	**12**	FF	IS4	,	<	L	\	l	|
1	1	0	1	**13**	CR	IS3	-	=	M	]	m	}
1	1	1	0	**14**	SO	IS2	.	>	N	^	n	~
1	1	1	1	**15**	SI	IS1	/	?	O	_	o	DEL

A.3 a-characters

The a-characters are those which are not shaded in Table A.2 showing the IRV.

Table A.2 — a-characters

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	**0**	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	**1**	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	**2**	STX	DC2	”	2	B	R	b	r
0	0	1	1	**3**	ETX	DC3	#	3	C	S	c	s
0	1	0	0	**4**	EOT	DC4	$	4	D	T	d	t
0	1	0	1	**5**	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	**6**	ACK	SYN	&	6	F	V	f	v
0	1	1	1	**7**	BEL	ETB	’	7	G	W	g	w
1	0	0	0	**8**	BS	CAN	(	8	H	X	h	x
1	0	0	1	**9**	HT	EM	)	9	I	Y	i	y
1	0	1	0	**10**	LF	SUB	*	:	J	Z	j	z
1	0	1	1	**11**	VT	ESC	+	;	K	[	k	{
1	1	0	0	**12**	FF	IS4	,	<	L	\	l	|
1	1	0	1	**13**	CR	IS3	-	=	M	]	m	}
1	1	1	0	**14**	SO	IS2	.	>	N	^	n	~
1	1	1	1	**15**	SI	IS1	/	?	O	_	o	DEL

Annex B
(informative)

ECMA-119 2nd edition and Joliet Specification: Modifications

B.1 Changes between 2nd edition and this edition

B.1.1 Main changes

The enhanced volume descriptor has been added, and the following specifications have been introduced for the enhanced volume descriptor.

a) The volume descriptor version has been modified to indicate the new structure.

- Version number 2 indicated the new specification.

b) The file structure version has been modified to indicate the new structure.

- Version number 2 indicated the new specification.

c) The limitation in the depth of hierarchy is lifted.

- The levels in the hierarchy can exceed eight.

d) The file identifier is not separated into components.

- The separators are not specified.
- A complex file name can be expressed.

e) The file identifier does not have file version numbers.

- File with and without version numbers should not exist in the same directory.

f) The character used for filling byte positions which are specified to be characters is subject to agreement between the originator and the recipient of the volume.

g) The length of file identifier is limited to 207.

- Long file names can be expressed.

h) The length of a directory identifier is limited to 207.

- Long directory names can be expressed.

NOTE Without using the enhanced volume descriptor, this Ecma Standard can be read as in the 2nd edition.

B.1.2 Details of the changes

a) The limitation in the depth of hierarchy is lifted by using the enhanced volume descriptor (see 7.8.2.2).

b) Filler was defined (8.4.3.2), and (20) changed to filler (7.9.2, 8.4.5, 9.4.21, 9.4.22, 9.4.23, 9.4.24, 9.4.25, 9.4.26, 9.5.15, 9.5.16, 9.5.17, 9.5.18, 9.5.19, 9.5.20, 10.3).

c) Within the volume that identified by enhanced volume descriptor, the separators are not specified (see 8.4.3.1).

d) The enhanced volume descriptor was added to the use of characters (see 8.4.4, 9.5.5, 9.5.6, 9.5.14, 9.5.15, 9.5.16, 9.5.16, 9.5.17, 9.5.18, 9.5.19, 9.5.20).

e) File identifier format for enhanced volume descriptor was added (see 8.5.1).

f) Directory identifier for enhanced volume descriptor was added (see 8.6.1, 8.6.3).

g) File structure version number for enhanced volume descriptor was specified (9.4.31).

h) Enhanced volume descriptor was added to volume descriptors (see 7.7, 7.7.2 c), 9.5, Table 6).

i) Enhanced volume descriptor was added in volume descriptor type and volume descriptor version (9.5.2, 9.5.3).

j) Interchange level 1 for a volume identified by an enhanced volume descriptor was added (11.2).

k) Other changes related to the enhanced volume descriptor are found in 7.8.2, 7.8.3, 8.5.2, 9.1.2, 14.3.2, 15.3.2 and 15.5.2.

l) Other changes which are not related to the enhanced volume descriptor are listed in Clause B.1.

B.2 Joliet Specification

The “Joliet Specification” defines a volume space, which is almost same concept to volume space identified by the enhanced volume descriptor of this Ecma Standard.

The “Joliet Specification” defines an extended volume space which is identified by the supplementary volume descriptor.

Change details for the “Joliet Specification” are as follows.

a) In 7.8.2.2:

- the number of levels in the hierarchy may exceed eight;
- for each file recorded, the sum of the following shall not exceed 240 bytes:
  - the length of the file identifier (see 8.5.2),
  - the length of the directory identifiers (see 8.6.3) of all relevant directories,
  - the number of relevant directories.

b) In 9.5.4:

- volume flags field of supplementary volume descriptor bit 0: ZERO.

c) In 9.5.7:

- escape sequences field shall be set to (25 2F 40), (25 2F 43) or (25 2F 45), by UCS-2 level.

d) In 8.4.2.1:

- c-characters shall be recorded in the most significant byte first;
- c-characters are all UCS-2 code points, except for the following UCS-2 code points:
    - all code points between (00 00) and (00 1F), inclusive (control characters),
    - (00 2A) '*' (asterisk),
    - (00 2F) '/' (solidus),
    - (00 3A) ':' (colon),
    - (00 3B) ';' (semicolon),
    - (00 3F) '?' (question mark),
    - (00 5C) '\' (reverse solidus),
- c-characters shall be recorded in most significant byte first (see 8.2.3);
- a1-characters and d1-characters are not specified.

e) In 8.4.3.1:

- separator 1 is represented by the bit combination (00 2E);
- separator 2 is represented by the bit combination (00 3B).

f) In 8.4.3.2:

- the bit combination of filler shall be (00).

g) In 8.5.1:

- file version number shall be recoded in c-characters (UCS-2);
- the sum of following may be up to 128 bytes (64 UCS-2 characters) in length:
    - if there is a file name, the length of the file name,
    - if there is a file name extension, the length of file name extension.

h) In 8.6.3:

- due to the d1-characters not specified, the directory identifiers may contain a file name extension;
- the directory identifiers may be up to 128 bytes (64 UCS-2 characters) in length;
- the length of the directory identifiers shall be calculated in the same was as for the file identifier length (see 8.5.2).

i) In 10.3 c):

- padding byte for file version number is (00).

The original “Joliet Specification” designed for ISO 9660:1988, which ECMA-119 2nd edition is technically identical with, is described in Annex C.

Annex C
(informative)

Joliet Specification

C.1 General

The following clauses of this Annex are a faithful copy of

- Joliet Specification: CD-ROM Recording Spec ISO 9660:1988, Extensions for Unicode Version 1, 1995

on the template of an Ecma Standard. Therefore, the following readings need to be changed in this Annex:

- this standard to this Annex;
- ISO 2022 to ISO 2022:1986;
- ISO 9660 to ISO 9660:1988;
- ISO 10646 to ISO/IEC 10646-1:1993.

In Annex C.6, the documents, System Use Sharing Protocol and RockRidge Interchange Protocol, are listed as the references. The document of System Use Sharing Protocol specifies an extension to the ISO 9660 format for CD-ROM which enables the shared utilization of the System Use Areas provided by ISO 9660:1988 for recording system-specific extensions to ISO 9660 defined by multiple independent parties. The document of Rock Ridge Interchange Protocol specifies an extension to the ISO 9660 format for CD-ROM which enables the recording of POSIX File System semantics. Both of the documents were developed as IEEE standards by IEEE CDROM File Systems Working Group and are generally described as follows.

- IEEE P1281, System Use Sharing Protocol Version 1.12, 1994
- IEEE P1282, Rock Ridge Interchange Protocol Version 1.12, 1994

C.2 Preface

C.2.1 Purpose and Scope

While the CD-ROM media provides for cost-effective software distribution, the existing ISO 9660 file system contains a number of restrictions which interfere with simple and efficient distribution of files on a CD-ROM.

The read-only nature of CD-ROM media has led content authors to continue to use traditional magnetic media as their main avenue for creating applications. Each of the existing file systems for magnetic media contain various features which cannot be represented on CD-ROM media using an unenhanced version of ISO 9660.

As content authors attempt to transfer their applications to the CD-ROM, they are likely to find that some of their work cannot be distributed on the CD-ROM media due to restrictions in the ISO 9660 format. This frustrates some content authors.

Because the CD-ROM media is mainly a distribution media, rather than a creative (read/write) media, it is necessary for the CD-ROM file system to support a superset of the creative media features. This fundamental flaw in the design of ISO 9660 has prompted several operating systems vendors to extend ISO 9660 in several ways. Some examples are Rock Ridge Interchange Protocol and Apple's use of the System Use Area to store finder flags.

Some of the ISO 9660 problems which are addressed by this specification include:

- Character Set limitations.
- File Name Length limitations
- Directory Tree Depth limitations
- Directory Name Format limitations
- Wide Character (16-bit character) ambiguities

The general design approach used in the Joliet specification is to relax restrictions and resolve ambiguities in the ISO 9660:1988 specification so the practical goals can be met.

C.2.2 Overview

The Joliet specification utilizes the supplementary volume descriptor (SVD) feature of ISO 9660 to specify a set of files recorded within the Unicode character set.

The ISO 10646 character set specification may be identified by an ISO 2022 escape sequence. By recording this escape sequence in an ISO 9660 SVD, this technique for identifying the Unicode SVD is compliant with the ISO 9660 specification. It also retains interchange by not disrupting the files referenced through the primary volume descriptor (PVD).

All that remains is to resolve minor technical ambiguities within ISO 9660 which arise as the result of the use of wide characters.

Because the use of this particular escape sequence in an ISO 9660 SVD is unprecedented up to this time, several of the restrictions which are imposed by ISO 9660 may be relaxed without significantly disrupting information interchange between existing systems from a practical standpoint.

This design approach has several benefits. For instance, the use of the existing ISO 9660 standard allows for straightforward integration with existing extensions to ISO 9660. The designs for the System Use Sharing Protocol, Rock Ridge extensions for POSIX semantics, CD-XA System Use Area Semantics, Apple's Finder Flags and Resource Forks, all port in a straightforward manner to the Joliet specification.

Also, the use of a new SVD eliminates the danger of breaking software compatibility with existing ISO 9660 systems. Existing software will simply ignore the Unicode SVD, and will simply use the PVD instead. This compatibility "safety-valve" makes the goal of relaxing the file system's restrictions easier.

This document describes how a CD-ROM may be constructed so that names on the volume can be recorded in Unicode while remaining in compliance with ISO 9660. The particular ISO 10646 character sets used here are UCS-2 Level 1, UCS-2 Level 2, and UCS-2 Level 3.

The basic strategy of CD-ROM volume recognition is the Volume Recognition Sequence, which is a sequence of volume descriptors, recorded one per sector, starting at Sector 16 in the first track of the last session on the disc. A receiving system reads these sectors and chooses a particular volume descriptor from the sequence. This volume descriptor acts as a kind of anchor upon which the remainder of the volume is constructed.

C.2.3 Terminology and Notation

Joliet is based on the ISO 9660:1988 standard. Unless defined in this document, the terminology used shall be as defined in ISO 9660:1988.

The following notation is used in this document.

a) Decimal and Hexadecimal Notation

- Numbers in decimal notation are represented by decimal digits, namely 0 to 9.

- Numbers in hexadecimal notation are represented by hexadecimal digits, namely 0 to 9 and A to F, shown in parentheses. For instance, the hexadecimal number D0 shall be written as (D0).

b) A literal sequence of ASCII characters will be represented by those characters within single quotes. For instance, 'ABC' means the byte sequence (41)(42)(43).

c) References to characters in the ISO 2022 escape sequence will be given in comma-separated decimal nibble/nibble format, in hexadecimal format, and as ASCII characters, with equal signs between each format, all enclosed within parenthesis. For instance, the 3-byte ISO 2022 escape sequence for Shift-JIS is (2/4, 2/11, 3/10 = (24)(2B)(3A)= '$+:').

C.3 Joliet Recording Specification

C.3.1 Change Summary

The Joliet specification resolves the following ISO 9660 ambiguities for UCS-2 volumes:

- Use a SVD with a UCS-2 (UNICODE) Escape Sequence.

- The UCS-2 escape sequences used are: (25)(2F)(40), (25)(2F)(43), or (25)(2F)(45).

- The default setting of bit 0 of the SVD "Volume Flags Field" is ZERO.

- The Unicode Wide characters shall be recorded in "Big Endian" (Motorola) format.

- Special Directory Identifiers are recorded as single byte names containing (00) or (01).

- SEPARATOR 1 and SEPARATOR 2 are encoded using an equivalent 16-bit code point.

- Sort ordering is unchanged, except that all justification pad bytes are to be set to (00).

The Joliet specification recommends that several ISO 9660 restrictions be lifted on UCS-2 volumes. The Joliet specification allows for the following interchange rules:

- The File or Directory Identifiers may be up to 128 bytes (64 unicode characters) in length.

- Directory Identifiers may contain file name extensions.

- The Directory Hierarchy may be recorded deeper than 8 levels.

- The Volume Recognition Sequence supports multisession. This is compatible with the CD-Bridge specification.

The Joliet specification may be extended through the use of the following specifications:

- Mode 2 Form 2 extents and CD-DA extents, ("System Description CD-ROM XA")

- System Use Sharing Protocol (not explicitly specified here)

- RockRidge Interchange Protocol (not explicitly specified here)

- Other future CD-ROM file system formats

C.4 Identifying an ISO 9660 SVD as Unicode (UCS-2)

C.4.1 SVD Escape Sequences Field

The Escape Sequences field of an ISO 9660 Supplementary Volume Descriptor (ISO 9660 clause 8.5.6) shall identify the character set used to interpret descriptor fields related to the Directory Hierarchy identified by the Volume Descriptor.

If the Escape Sequences field of an ISO 9660 SVD identifies any of the following UCS-2 escape sequences, then the descriptor fields related to the Directory Hierarchy identified by that Volume Descriptor shall be interpreted according to the identified UCS-2 character set.

Table C.1 — ISO 2022 UCS-2 Escape Sequences

Standard	Level	Decimal	Hex bytes	ASCII
UCS-2	Level 1	2/5, 2/15, 4/0	(25)(2F)(40)	'%\@'
UCS-2	Level 2	2/5, 2/15, 4/3	(25)(2F)(43)	'%\C
UCS-2	Level 3	2/5, 2/15, 4/5	(25)(2F)(45)	'%\E'
NOTE ISO 2022 Escape Sequence as recorded in the ISO 9660 SVD				

A "Unicode Volume" refers to the Volume Descriptor and Directory Hierarchy identified by a Supplementary Volume Descriptor containing an Escape Sequences field which identifies any of the above UCS-2 character sets.

C.4.2 SVD Volume Flags Field

The UCS-2 Level 1, UCS Level 2, and UCS-2 Level 3 escape sequences are considered to be registered according ISO 2735 for purposes of setting bit 0 of the Volume Flags field of the SVD.

The nominal value of Bit 0 of the Volume Flags field for a Unicode SVD shall be ZERO.

C.4.3 Resolution of ISO 9660 Ambiguities for Wide Characters

This specification resolves ISO 9660 ambiguities with respect to wide (16-bit) character sets, such as the UCS-2 character set.

C.4.4 Wide Character Byte Ordering

All UCS-2 characters shall be recorded according to ISO 9660:1988 clause 7.2.2, 16-bit numerical value, most significant byte first ("Big Endian").

C.4.5 Allowed Character Set

All UCS-2 code points shall be allowed except for the following UCS-2 code points:

— All code points between (00)(00) and (00)(1F), inclusive. (Control Characters)

— (00)(2A) '*'(Asterisk)

— (00)(2F) '/' (Forward Slash)

— (00)(3A) ':' (Colon)

— (00)(3B) ';' (Semicolon)

- (00)(3F) '?' (Question Mark)

- (00)(5C) '\' (Backslash)

C.4.6 Special Directory Identifiers

Clause 7.6 of ISO 9660 describes the recording of reserved directory identifiers for the root, current, and parent directory identifiers as single (00) or single (01) bytes.

In a wide character set, it is not possible to represent a character in a single byte. The following portions of the ISO 9660:1988 specification referring to reserved directory identifiers are ambiguous.

The ISO 9660:1988 clauses in question are as follows:

- 6.8.2.2 Identification of directories

- 7.6.2 Reserved Directory Identifiers

- 9.1.11 File Identifier

- 9.4.5 Directory Identifier.

These special case directory identifiers are not intended to represent characters in a graphic character set. These characters are placeholders, not characters. Therefore, these definitions remain unchanged on a volume recorded in Unicode.

Simply put, Special Directory Identifiers shall remain as 8-bit values, even on a UCS-2 volume, where other characters have been expanded to 16-bits.

a) Root Directory

   The Directory Identifier of a Directory Record describing the Root Directory shall consist of a single (00) byte.

b) Current Directory

   The Directory Identifier of the first Directory Record of each directory shall consist of a single (00) byte.

c) Parent Directory

   The Directory Identifier of the second Directory Record of each directory shall consist of a single (01) byte.

C.4.7 Separator Characters

The separator characters SEPARATOR 1 and SEPARATOR 2 are specified as 8-bit characters, which cannot be represented in a wide character set, so the ISO 9660:1988 specification clauses referring to SEPARATOR 1 and SEPARATOR 2 are ambiguous.

The ISO 9660:1988 clauses in question are as follows:

- 7.4.3 Separators

- 7.5.1 File Identifier format

- 7.5.2 File Identifier length

- 8.4.24 Abstract File Identifier

- 8.4.25 Bibliographic File Identifier
- 8.5.17 Copyright File Identifier
- 8.5.19 Bibliographic File Identifier
- 9.1.11 File Identifier

The values SEPARATOR 1 and SEPARATOR 2 shall be represented differently depending on the d1 character set.

In the case of an SVD identifying a UCS-2 character set, the values of SEPARATOR 1 and SEPARATOR 2 shall be recorded as a UCS-2 character with an equivalent code point value.

Otherwise, the definitions of SEPARATOR 1 and SEPARATOR 2 shall be recorded according to clause 7.4.3 of ISO 9660:1988.

Simply put, SEPARATOR 1 and SEPARATOR 2 shall be expanded to 16-bits.

Table C.2 — Separator Representations

Separator	ISO 9660:1988 volume	Unicode volume
	Bit combination	UCS-2 codepoint
SEPARATOR 1	(2E)	(00)(2E)
SEPARATOR 2	(3B)	(00)(3B)

C.4.8 Sort Ordering

C.4.8.1 General

ISO 9660 specifies the order of path table records within a path table, and specifies the order of directory records within a directory. These sorting algorithms assume an 8-bit character set is used. These sorting algorithms are ambiguous when used with wide characters.

The ISO 9660:1988 clauses in question are as follows:

- 6.9.1 Order of Path Table Records
- 9.3 Order of Directory Records

The only change required is to redefine the value of the sort justification pad byte to zero (00).

Simply put, comparing the byte contents in all positions remains a suitable sorting algorithm for the descriptor fields recorded in a UCS-2 SVD Directory Hierarchy. This is one of the primary reasons for selecting the Big Endian format to represent all UCS-2 characters.

C.4.8.2 Natural Language Sorting

On a Unicode volume, the 16-bit UCS-2 code points are used to determine the Order of Path Table Records and the Order of Directory Records.

No attempt will be made to provide natural language sorting on the media. Natural language sorting may optionally be provided by a display application as desired.

C.4.8.3 Justification Pad Bytes

The sort ordering algorithms as specified in ISO 9660:1988 clauses 6.9.1 and 9.3 are acceptable except for the value of the justification "pad byte".

The value of the justification "pad byte" as specified in ISO 9660:1988 clause 6.9.1 shall be (00). This is changed from a value of (20) as specified in that same clause.

The value of the justification "pad byte" as specified in ISO 9660:1988 clause 9.3 a) and b) shall be (00). This is changed from a value of (20) as specified in those same clauses.

The value of the justification "pad byte" as specified in ISO 9660:1988 clause 9.3 c) shall be (00). This is changed from a value of (30) as specified in that same clause.

Simply put, set all the justification "pad bytes" to zero to simplify sorting.

C.4.8.4 Mandatory Sort Ordering

Correct sort ordering is mandatory on UCS-2 volumes.

C.4.8.5 Descriptor Fields affected by the UCS-2 Escape Sequence

If a UCS-2 escape sequence is detected in a supplementary volume descriptor, the following descriptor fields referenced from that supplementary volume descriptor shall contain UCS-2 characters.

- ISO 9660:1988 clause 8.5.4 System Identifier
- ISO 9660:1988 clause 8.5.5 Volume Identifier
- ISO 9660:1988 clause 8.5.13 Volume Set Identifier
- ISO 9660:1988 clause 8.5.14 Publisher Identifier
- ISO 9660:1988 clause 8.5.15 Data Preparer Identifier
- ISO 9660:1988 clause 8.5.16 Application Identifier
- ISO 9660:1988 clause 8.5.17 Copyright File Identifier
- ISO 9660:1988 clause 8.5.18 Abstract File Identifier
- ISO 9660:1988 clause 8.5.19 Bibliographic File Identifier
- ISO 9660:1988 clause 9.1.11 File Identifier
- ISO 9660:1988 clause 9.4.5 Directory Identifier
- ISO 9660:1988 clause 9.5.11 System Identifier (of Extended Attribute Record)

C.4.9 Relaxation of ISO 9660 Restrictions on UCS-2 Volumes

C.4.9.1 General

Several ISO 9660 restrictions will be relaxed to achieve a more useful recording specification. Joliet receiving systems shall be capable of receiving media recorded with restrictions which have been relaxed relative to ISO 9660.

C.4.9.2 Maximum File Identifier Length Increased

Joliet receiving systems shall receive directory hierarchies recorded with file identifiers longer than those allowed by ISO 9660 receiving systems.

ISO 9660 (clause 7.5.1) states that the sum of the following shall not exceed 30:

- If there is a file name, the length of the file name.

- If there is a file name extension, the length of the file name extension.

On Joliet compliant media, however, the sum as calculated above shall not exceed 128, to allow for longer file identifiers.

The above lengths shall be expressed as a number of bytes.

C.4.9.3 Maximum Directory Identifier Length Increased

Joliet receiving systems shall receive directory hierarchies recorded with file names longer than those allowed by ISO 9660 receiving systems.

ISO 9660 (clause 7.6.3) states that the length of a directory identifier shall not exceed 31.

On Joliet compliant media, however, the length of a directory identifier shall not exceed 128, to allow for longer directory identifiers.

The above lengths shall be expressed as a number of bytes.

C.4.9.4 Directory Names May Have File Name Extensions

ISO 9660 does not allow directory identifiers to contain file name extensions.

On Joliet compliant media, however, directory identifiers may contain file name extensions.

The Joliet directory identifier format shall be calculated according to ISO 9660 clause 7.5.1 "File Identifier format", with the exception that the length of a directory identifier may exceed 31, but shall not exceed 128.

In addition, the Joliet directory identifier format shall comply with ISO 9660 clause 7.6.2 "Reserved Directory Identifiers".

The directory identifier length shall be calculated according to ISO 9660 clause 7.5.2 "File Identifier length".

The above lengths shall be expressed as a number of bytes.

C.4.9.5 Maximum Directory Hierarchy Depth May Exceed 8 Levels

ISO 9660 (clause 6.8.2.1) specifies restrictions regarding the Depth of Directory Hierarchy. This clause of ISO 9660 specifies that this number of levels in the hierarchy shall not exceed eight.

On Joliet compliant media, however, the number of levels in the hierarchy may exceed eight.

Joliet compliant media shall comply with the remainder of ISO 9660 clause 6.8.2.1, so that for each file recorded, the sum of the following shall not exceed 240:

- the length of the file identifier;

- the length of the directory identifiers of all relevant directories;

— the number of relevant directories.

The above lengths shall be expressed as a number of bytes.

C.5 Extensions to Joliet

C.5.1 Joliet for Multisession Media

C.5.1.1 Multisession recordings are Received

When provided with CD-ROM reader hardware with multisession capability, Joliet receiving systems shall receive media recorded using the multisession recording technique.

The details of this technique are provided below

C.5.1.2 Logical Sector Addressing on Multisession Recordings

Each sector on the media is assigned a unique Logical Sector Address.

Logical Sector Addresses zero and above increase linearly across the surface of the disc, regardless of session boundaries.

Logical Sector Address zero references the sector with Minute:Second:Frame address 00:02:00 in the first session. All other Logical Sector Addresses are relative to Minute:Second:Frame address 00:02:00 in the first session.

The conversion between Logical Sector Addresses and Minute:Second:Frame addresses is Logical Sector Address = (((Minute*60)+Seconds)*75) - 150.

Simply put, the Logical Sector Address on a multisession disc describes a flat address space.

C.5.1.3 Multisession Addressability

The data area for a volume may span multiple sessions.

For example, if a disc is recorded with 3 sessions, the directory hierarchy described by a volume descriptor in session 3 may reference logical sectors recorded in session 1, 2, or 3.

C.5.1.4 Multisession Volume Recognition Sequence

The Volume Recognition Sequence shall begin at the 16th logical sector of the first track of the last session on the disc.

This Volume Recognition Sequence supersedes all other Volume Recognition Sequences on the disc. The interpretation of the Volume Recognition Sequence is otherwise unchanged.

For example, consider a disc that contains 3 sessions, where session 1 starts at 00:00:00, session 2 starts at 10:00:00, and session 3 starts at 20:00:00. The Volume Recognition Sequence for this disc would start at Minute:Second:Frame address 20:00:16.

This technique is compatible with the CD-Bridge multisession technique.

C.5.1.5 Track Modes and Sector Forms

The data area for a Joliet volume on a CD-ROM shall be comprised of either Mode 1 or Mode 2 Form 1 sectors. CD-ROM media utilizing the multisession recording techniques outlined above may not contain any Mode 1 sectors anywhere on the media. Mode 1 sectors are allowed only on single-session media.

Mode 2 Form 2 sectors and CD-Digital Audio tracks may be recorded on the same media as a Joliet volume. In this case, the CD-XA extensions to Joliet may be utilized to identify Mode 2 Form 2 extents and CD-Digital Audio extents.

CD-Digital Audio tracks may not be recorded in sessions 2 and higher. If any CD-Digital Audio tracks are recorded, all the CD-Digital Audio tracks shall be recorded in the first session.

C.5.2 CD-XA Extensions to Joliet

CD-ROM discs utilizing the Joliet extensions to ISO 9660 and which also identify mode 2 form 2 extents or CD-Digital Audio extents shall be marked with a CD-ROM XA Label as specified in "System Description CD-XA" section 2.1.

The CD-ROM XA Label shall be located at offset 1024 (byte position 1025) in the Joliet Supplementary Volume Descriptor. The identifying signature 'CD-XA001' shall be recorded starting at offset 1024 in the Joliet Supplementary Volume Descriptor. This identifying signature is equivalent to the hex bytes (43)(44)(2D)(58)(41)(30)(30)(31).

Mode 2 form 2 extents shall be identified using recording rules outlined in "System Description CD-XA", section 2.7. In this case, bit 12 of the Attributes field of the "XA System Use Information" shall be set to one to identify that the file contains mode 2 form 2 sectors. See below for additional information regarding Data Length.

CD-Digital Audio extents shall be identified using recording rules outlined in "System Description CD-XA", section 2.7. In this case, bit 14 of the Attributes field of the "XA System Use Information" shall be set to one to identify that the file is comprised of an extent of CD-Digital Audio. See below for additional information regarding Data Length.

If a file is marked such that either bit 12 is set to one or bit 14 is set to one in the Attributes field of the "XA System Use Information", then the Data Length field of the Directory Record shall be set to 2048 times the number of sectors contained in the extent.

See ISO 9660:1988 clause 9.1.4.

C.5.3 Other Extensions to Joliet

The Joliet Extensions to ISO 9660 are designed to coexist with other extensions such as the "System Use Sharing Protocol" and "RockRidge Interchange Protocol". However, these protocols are not an integral part of the Joliet specification.

The method used to integrate these other protocols into Joliet is not defined here.

C.6 Bibliography

— ISO 2022 - Information processing — ISO 7-bit and 8-bit coded character sets - Code extension techniques, International Organization for Standardization,

— ISO 9660 - Information processing — Volume and file structure of CD-ROM for information interchange, International Organization for Standardization, 1988-04-15

— ISO 10149 : 1989 (E) - Information technology — Data interchange on read-only 120mm optical data discs (CD-ROM) "YellowBook", International Organization for Standardization, 1989-09-01

- ISO 10646 - Information technology — Universal Multiple-Octet Coded Character Sets (UCS), International Organization for Standardization,
- The Unicode Standard - Worldwide Character Encoding Version 1.0, The Unicode Consortium, Addison-Wesley Publishing Company, Inc, 1990-1991 Unicode, Inc., Volume 1
- Orangebook, N. V. Philips and Sony Corporation, November 1990
- System Description CD-XA, N. V. Philips and Sony Corporation, March 1991
- System Use Sharing Protocol
- RockRidge Interchange Protocol

Bibliography

[1] ECMA-167, Volume and file structure for write-once and rewritable media using non-sequential recording for information interchange

[2] ISO/IEC 646, Information technology — ISO 7-bit coded character set for information interchange

[3] ISO/IEC 2022, Information technology — Character code structure and extension techniques

[4] ISO 7665:1983, Information processing — File structure and labelling of flexible disk cartridges for information interchange

[5] ISO/IEC 9293:1994, Information technology — Volume and file structure of disk cartridges for information interchange

[6] ISO/IEC 9660, Information processing — Volume and file structure of CD-ROM for information interchange

[7] ISO/IEC 10149, Information technology — Data interchange on read-only 120 mm optical data disks (CD-ROM)

[8] N. V. Philips and Sony Corporation. Orange Book, 1990

[9] N. V. Philips and Sony Corporation. CD-ROM XA, 1991

[10] Microsoft Corporation. Joliet Specification: Extensions for Unicode Version 1, 1995