

ECMA

Standardizing Information and Communication Systems

System-Independent Data Format

ECMA

Standardizing Information and Communication Systems

System-Independent Data Format

(SIDF)

Brief History

The System Independent Data Format Association (SIDF) was formed in early 1993 by a consortium of industries. ECMA TC15 adopted the work of this committee in early 1994. The document was transformed into an ECMA Standard by ECMA TC15.

It is contributed to ISO/IEC JTC 1 for adoption as an International Standard under the fast-track procedure.

This ECMA Standard has been adopted by the ECMA General Assembly of December 1994.

Table of contents

| | Page |
|--|-------------|
| Section 1 - General | 1 |
| 1 Scope | 1 |
| 2 Conformance | 1 |
| 2.1 Conformance of Media Volumes | 1 |
| 2.2 Conformance of an originating system | 1 |
| 2.3 Conformance of a receiving system | 1 |
| 3 References | 1 |
| 4 Definitions | 1 |
| 4.1 Blank Space | 1 |
| 4.2 byte | 1 |
| 4.3 Byte Sequence | 2 |
| 4.4 Field | 2 |
| 4.5 Field Identifier (FID) | 2 |
| 4.6 Field Table | 2 |
| 4.7 File | 2 |
| 4.8 file mark | 2 |
| 4.9 File Set | 2 |
| 4.10 Interleaved File Sets | 2 |
| 4.11 Iterated Field Set | 2 |
| 4.12 operating system | 2 |
| 4.13 originating system | 2 |
| 4.14 Relative Byte Position | 2 |
| 4.15 receiving system | 2 |
| 4.16 Sector | 2 |
| 4.17 Sector Number | 2 |
| 4.18 Source | 3 |
| 4.18 standard for recording | 3 |
| 4.20 Volume | 3 |
| 4.21 Volume Set | 3 |
| 4.22 Volume Set Sequence Number | 3 |
| 5 General conventions and notations | 3 |
| 5.1 Representation of numbers | 3 |
| 5.2 Names | 3 |
| 5.3 Structures | 3 |
| 5.4 Fields | 3 |
| 5.5 Field Identifiers (FIDs) | 3 |
| 5.6 Field Tables | 3 |
| 5.7 Headers | 3 |
| 5.8 Indices | 4 |
| 5.9 Structure description schema | 4 |

| | |
|---|----|
| Section 2 - Requirements for the media | 5 |
| 6 Specific conventions and notations for the recorded format | 5 |
| 6.1 Recording of numbers | 5 |
| 6.2 Character sets and coding | 5 |
| 6.3 Character set specification (CHAR SPEC) | 5 |
| 6.3.1 Character Set Type | 5 |
| 6.3.2 Character Set Information | 6 |
| 6.4 CS0 character set | 6 |
| 6.5 CS1 character set | 6 |
| 6.6 CS2 character set | 6 |
| 6.7 CS3 character set | 6 |
| 6.8 CS4 character set | 6 |
| 6.9 CS5 character set | 6 |
| 6.10 CS6 character set | 6 |
| 6.11 CS7 character set | 6 |
| 6.11.1 Code extension characters | 6 |
| 6.12 CS8 character set | 7 |
| 6.13 Source Name Space specification | 7 |
| 6.13.1 Name Space | 7 |
| 6.14 NS0 Name Space | 8 |
| 6.15 NS1 Name Space | 8 |
| 6.16 NS2 Name Space | 8 |
| 6.17 NS3 Name Space | 9 |
| 6.18 NS4 Name Space | 9 |
| 6.19 NS5 Name Space | 9 |
| 6.20 NSFC Name Space | 10 |
| 6.21 NSFEC Name Space | 10 |
| 6.22 Strings | 10 |
| 6.23 Resynchronization Pattern | 10 |
| 7 Timestamp | 10 |
| 7.1 Type and Time Zone (RBP 0) | 10 |
| 7.2 Year (RBP 2) | 11 |
| 7.3 Month (RBP 4) | 11 |
| 7.4 Day (RBP 5) | 11 |
| 7.5 Hour (RBP 6) | 11 |
| 7.6 Minute (RBP 7) | 11 |
| 7.7 Second (RBP 8) | 11 |
| 7.8 Centiseconds (RBP 9) | 11 |
| 7.9 Hundreds of microseconds (RBP 10) | 11 |
| 7.10 Microseconds (RBP 11) | 11 |
| 8 Requirements for a standard for recording | 11 |
| 9 Cyclic Redundancy Check (CRC) | 12 |
| 10 Organization of Information on a Volume | 12 |
| 10.1 Recording of Sectors | 12 |

| | |
|---|-----------|
| 10.2 Organization of a Volume | 12 |
| 10.3 Volume Sets | 12 |
| 10.4 Recording of Fields | 14 |
| 10.4.1 Null Field | 14 |
| 10.5 Recording of Field Tables | 14 |
| 10.6 Recording of Buffers | 14 |
| 10.7 Recording of file marks | 15 |
| 11 Organization of Information within a File Set | 15 |
| 11.1 Organization of a File Set | 15 |
| 11.2 File Sets spanning multiple Volumes | 17 |
| 11.3 Interleaving | 17 |
| 12 Organization of information within a File | 18 |
| 13 Field Table description | 18 |
| 13.1 Volume Header Field Table | 19 |
| 13.2 Volume Trailer Field Table | 20 |
| 13.3 Blank Space Field Table | 21 |
| 13.4 Buffer Header Field Table | 21 |
| 13.5 Volume Index Field Table | 23 |
| 13.6 Volume Subindex Field Table | 24 |
| 13.7 File Set Header Field Table | 24 |
| 13.8 File Set Continuation Header Field Table | 27 |
| 13.9 File Set Trailer Field Table | 27 |
| 13.10 File Set Index Field Table | 28 |
| 13.11 File Set Subindex Field Table | 31 |
| 13.12 File Header Field Table | 31 |
| 13.13 File Continuation Header Field Table | 31 |
| 13.14 File Information Field Table | 32 |
| 13.15 File Data | 33 |
| 13.15.1 Path Field Table | 33 |
| 13.15.2 Characteristics Field Table | 33 |
| 13.15.3 Source volume File Data | 34 |
| 13.15.4 Source directory File Data | 35 |
| 13.15.5 Source file File data | 36 |
| 13.15.6 Transaction Set File data | 37 |
| 13.15.7 Streams | 38 |
| 13.16 Levels of partition interchange | 41 |
| 13.16.1 Level 1 of partition interchange | 41 |
| 13.16.2 Level 2 of partition interchange | 41 |
| 14 Requirements for systems | 41 |
| 14.1 Requirements for the description of systems | 41 |
| 14.2 Requirements for an originating system | 41 |
| 14.2.1 General | 41 |
| 14.2.2 Mandatory access by user | 41 |
| 14.2.3 Optional access by user | 41 |
| 14.2.4 Volume characteristics | 42 |
| 14.2.5 File Set characteristics | 42 |

| | |
|--|-----------|
| 14.2.6 Recording of Bit Data | 42 |
| 14.2.7 Cyclic Redundancy Check (CRC) | 42 |
| 14.3 Requirements for a receiving system | 43 |
| 14.3.1 General | 43 |
| 14.3.2 Levels of conformance | 43 |
| Annex A - Structure of the Field Identifier | 45 |
| Annex B - Field Data Length Structure | 49 |
| Annex C - Field specification | 51 |
| Annex D - Numerical List of the Field Identifiers specified by this ECMA Standard | 81 |
| Annex E - Numerical List of Field Identifiers which are deprecated or not specified by this ECMA Standard | 89 |

Section 1 - General

1 Scope

This ECMA Standard specifies a logical format for information interchange and secondary data storage. The format provides a system-independent data format (SIDF) for the representation of primary file system information. This information includes, among other things, data, attributes and characteristics. This ECMA Standard specifies

- the organization of the information on target media,
- requirements for originating and receiving systems for the processing of the information.

2 Conformance

2.1 Conformance of Media Volumes

A Volume shall be in conformance with this ECMA Standard if all information recorded on it meets the relevant requirements of sections 2 and 3 for the Level of partition claimed.

2.2 Conformance of an originating system

An originating system shall be in conformance with this ECMA Standard if it meets the requirements of 14.2.

2.3 Conformance of a receiving system

A receiving system shall be in conformance with this ECMA Standard if it meets the requirements of 14.3 for the Level of conformance claimed.

3 References

- | | |
|-----------------|---|
| ECMA-6 (1991) | 7-bit Coded Character Set |
| ECMA-35 (1994) | Code Extension Techniques |
| ECMA-94 (1986) | 8-Bit Single-Byte Coded Graphic Character Sets - Latin Alphabets No. 1 to No. 4 |
| ECMA-119 (1987) | Volume and File Structure of CD-ROM for Information Interchange |
| ECMA-167 (1994) | Volume and File Structure of Write-Once and Rewritable Media using Non-Sequential Recording for Information Interchange |
- ISO/IEC 9945-1:1990, *Information technology - Portable operating system interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]*
- ISO/IEC 10646-1:1993, *Information technology - Universal multiple octet coded character set (UCS) - Part 1: Architecture and basic multilingual plane.*
- ISO/IEC 13800:1995, *Information technology - Procedure for the registration of identifiers and attributes for volume and file structure.*
- ITU Rec. X.25: Interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals operating in the Packet Mode and connected to Public Data Networks by dedicated Circuits.

4 Definitions

For the purposes of this ECMA Standard, the following definitions apply.

4.1 Blank Space

Either one or more NULL Fields, or a BLANK SPACE Field Table.

4.2 byte

A string of eight binary digits operated upon as a unit.

4.3 Byte Sequence

An ordered set of bytes.

4.4 Field

The basic logical grouping of data.

4.5 Field Identifier (FID)

A Byte Sequence which identifies a Field.

4.6 Field Table

A set of logically associated Fields.

4.7 File

A collection of data that is stored as a single unit. A File contains a representation of a single primary file system entity, such as a Source directory, file, data base object, or data base.

4.8 file mark

A sector containing a control block used as a delimiter to facilitate positioning, as defined by the relevant standard for recording.

NOTE

Some media may not support file marks.

4.9 File Set

A set of associated Files, sourced from a single Source, recorded within a Volume Set.

4.10 Interleaved File Sets

Two or more File Sets are interleaved if any part of at least one is recorded between any parts of another.

4.11 Iterated Field Set

A sequence of one or more Fields, which may be repeated within a Field Table.

4.12 operating system

The software which controls the file system of the Source.

4.13 originating system

An information processing system which can create a File Set in a Volume Set for the purpose of data interchange with another system.

4.14 Relative Byte Position

Indicates the byte position in a number.

4.15 receiving system

An information processing system which can process a File Set of a Volume Set which has been created by another system for the purpose of data interchange.

4.16 Sector

The data field of the smallest addressable part of the medium that can be accessed independently of other addressable parts of the medium.

NOTE

This Sector is different from the physical sectors of a physical track.

4.17 Sector Number

The ordinal sequence number of a Sector within a Volume, starting with 0 (0, 1, 2, 3, ..., n).

4.18 Source

The primary system (e.g. file system, data base) which is the source of the Files that are recorded as a File Set for secondary storage. The name of the Source is specified for the File Set, and is not included in the Source path information of the individual Files.

4.19 standard for recording

A standard that specifies the recording method and the addressing method for the information recorded on a medium.

4.20 Volume

A set of Sectors which resides within one physical partition on a secondary storage medium.

4.21 Volume Set

One Volume, or multiple logically related Volumes.

4.22 Volume Set Sequence Number

The ordinal sequence number of each Volume within a Volume Set, starting with 1 (1, 2, 3, ..., n).

5 General conventions and notations

5.1 Representation of numbers

- Numbers in decimal notation are shown as decimal digits.
- Numbers in hexadecimal notation are shown as a sequence of hexadecimal digits preceded by #.
- The setting of bits is denoted by ZERO or ONE.
- Numbers in binary notation and bit combinations are represented by sequences of ZEROs or ONES.
- Bit combinations are shown with the most significant bit to the left.
- Bits are indicated with the small letter b.
- Bytes are indicated with the capital letter B.

5.2 Names

The names of formally defined entities, such as a Volume, are shown with a capital initial.

The name of Fields, such as VOLUME HEADER, are written in capitals.

5.3 Structures

Data structures are shown in tables indicating the Relative Byte Position (RBP), purpose and size of the entities that comprise the structure.

5.4 Fields

Fields are denoted by names shown in capital letters.

5.5 Field Identifiers (FIDs)

A Byte Sequence in hexadecimal notation uniquely identifying a Field.

5.6 Field Tables

The name of a Field Table is that of its first Field, it is shown with capital initials. Field Tables list each Field with its name, its FID and a brief description of the data. The normative definition of each Field is specified in annex C. The Data Description reads "empty" if no Data part is included in the Field. Iterated Field Sets within a Field Table are shown indented, with a bar alongside. Nested Iterated Field Sets are shown with each inner level of nesting further indented than the outer level(s).

5.7 Headers

The term "File Set (Continuation) Header" is used to express: File Set Header and/or File Set Continuation Header. The term "File Record (Continuation) Header" is used to express: File Record Header and/or File Record Continuation Header.

5.8 Indices

The term "Volume (Sub)Index" is used to express: Volume Index and/or Volume Subindex. The term "File Set (Sub)Index" is used to express: File Set Index and/or File Set Subindex.

5.9 Structure description schema

Requirements for recording the different elements of the format are summarized by means of a structure description.

A structure shall be a sequence of terms. A term shall be either:

- a name, enclosed by [], followed by a definition enclosed by {}, or
- a name, enclosed by <>, of a term defined in the structure or of a descriptor defined elsewhere in this ECMA Standard in the text associated with the structure.

Within a definition enclosed by {}, the Or Operator and Repetition Operators may be applied to the terms therein.

The Or Operator is represented by the character VERTICAL LINE: |. It is placed between the terms to which it applies and indicates that either term, but not both simultaneously, is valid at that point in the definition.

There are three Repetition Operators, which shall be interpreted as shown in figure 1, where *n* and *m* are decimal digits.

| Repetition Operator | Interpretation |
|---------------------|--|
| <i>n</i> | <i>n</i> occurrences of the preceding term |
| <i>n</i> + | <i>n</i> or more occurrences of the preceding term |
| <i>n</i> + <i>m</i> | <i>n</i> to <i>m</i> occurrences of the preceding term |

Figure 1 - Repetition Operators

The Repetition Operators shall apply only to the preceding term.

The operators are listed in increasing order of precedence:

| repetition operator [] ()

Higher precedence operators shall be applied before lower precedence operators.

As an example, the schema shown in figure 2 specifies that the entity "Concert" shall be organized as zero or one Introduction, followed by one or more Pieces, an Intermission, one or more Pieces, and zero or more Encores. A Piece consists either of one Song or three to five Movements.

```

[Concert] {
    <Introduction> 0+1
    [Piece] {
        <Song>
        | <Movement> 3+5
    } 1+
    <Intermission>
    <Piece> 1+
    <Encore> 0+
}
    
```

Figure 2 - Example of the syntax of the structure description schema

Section 2 - Requirements for the media

6 Specific conventions and notations for the recorded format

6.1 Recording of numbers

- In each Field the information is recorded so that the least significant byte (denoted byte B0) is recorded first. Within each byte the least significant bit is denoted bit b0, the most significant bit is denoted b5, and is recorded first. This order of recording also applies to the data input of the cyclic redundancy code (CRC) and to its output (CRC bits).
- Negative values are recorded in TWO's complement notation.

6.2 Character sets and coding

Except as specified in this clause and in 6.13, the characters used in the Fields specified by this ECMA Standard shall be coded according to the International Reference Version (IRV) of ECMA-6. The CS4 character set described in 6.8 shall be the default character set, and shall be used if no other is explicitly specified. The specification of the characters allowed in these Fields and the method of recording shall be recorded as a CHAR SPEC Field (see 6.3).

NOTE

Support for a variety of character sets is a requirement of this ECMA Standard. Ideally, there should be only one character standard used. In practice, several standards, including ECMA-6, ECMA-35, ECMA-94 and ISO/IEC 10646-1 are used. This ECMA Standard accommodates current practice by specifying several character sets and providing a mechanism for specifying other character sets.

As an example, CS2 (see 6.6) uses the IRV of ECMA-6 as the basic character set but restricts Fields containing characters to a widely usable subset of this character set.

6.3 Character set specification (CHAR SPEC)

The data part of the CHAR SPEC Field consists of two parts, Character Set Type and Character Set Information, recorded as a sequence of bytes. The Character Set Type shall be recorded as the first byte of the sequence; the Character Set Information shall consist of any remaining bytes.

The set of characters allowed in certain Fields and/or Field Tables shall be specified by a one-byte Character Set Type, and additional Character Set Information appended immediately after as needed.

6.3.1 Character Set Type

The Character Set Type identifies the allowed set of characters as shown in figure 3.

| Type | Allowed characters |
|----------|---|
| 0 | The CS0 coded character set (see 6.4). |
| 1 | The CS1 coded character set (see 6.5). |
| 2 | The CS2 coded character set (see 6.6). |
| 3 | The CS3 coded character set (see 6.7). |
| 4 | The CS4 coded character set (see 6.8). |
| 5 | The CS5 coded character set (see 6.9). |
| 6 | The CS6 coded character set (see 6.10). |
| 7 | The CS7 coded character set (see 6.11). |
| 8 | The CS8 coded character set (see 6.12). |
| 9 to 255 | Reserved for future standardization |

Figure 3 - Sets of allowed characters

6.3.2 Character Set Information

The interpretation of the contents of this field, if any, is specified by the value of the Character Set Type as described in 6.4 to 6.12.

6.4 CS0 character set

The CS0 character set shall be subject to agreement between the originator and recipient of the medium.

The character set is undefined for interchange. However, the agreed character set can be identified in the Character Set Information, if so desired.

6.5 CS1 character set

The CS1 character set shall be the graphic characters of the character sets specified by the Character Set Information Field.

The Character Set Information Field shall specify one or more escape sequences, to be used in an 8-bit environment according to ECMA-35 that designate and implicitly invoke graphic character sets specified in ISO/IEC 10646-1. These sequences shall be recorded contiguously from the start of the Field.

6.6 CS2 character set

The CS2 character set shall be the 38 characters in positions 02/14, 03/00 to 03/09, 04/01 to 05/10, and 05/15 of the IRV of ECMA-6.

NOTE

These characters are: FULL STOP, DIGITs, LATIN CAPITAL LETTERs and LOW LINE.

6.7 CS3 character set

The CS3 character set shall be the 65 characters in positions 02/13 to 02/14, 03/00 to 03/09, 04/01 to 05/10, 05/15, and 06/01 to 07/10 of the IRV of ECMA-6.

NOTE

These characters are: HYPHEN-MINUS, FULL STOP, DIGITs, LATIN CAPITAL LETTERs, LATIN SMALL LETTERs and LOW LINE.

6.8 CS4 character set

The CS4 character set shall be the 95 characters in positions 02/00 to 07/14 of the IRV of ECMA-6.

6.9 CS5 character set

The CS5 character set shall be the 191 characters in positions 02/00 to 07/14 and 10/00 to 15/15 of Latin Alphabet No. 1 in ECMA-94.

6.10 CS6 character set

The CS6 character set shall be the graphic characters of the character sets specified by the Character Set Information Field.

The Character Set Information field shall specify one or more escape sequences according to ECMA-35 that designate and implicitly invoke the graphic character sets to be used in an 8-bit environment according to ECMA-35 or ISO/IEC 10646-1. These sequences shall be recorded contiguously from the start of the Field.

6.11 CS7 character set

The CS7 character set shall be the graphic characters of the character sets specified by the Character Set Information Field and code extension characters (see 6.11.1).

The Character Set Information field shall specify one or more escape sequences according to ECMA-35 that designate and implicitly invoke the graphic character sets to be used in an 8-bit environment according to ECMA-35 or ISO/IEC 10646-1. These sequences shall be recorded contiguously from the start of the field.

6.11.1 Code extension characters

Characters specified by the CS7 Character Set may include one or more of the following, referred to as code extension characters, to allow alternative character sets to be recorded.

- Escape sequences according to ECMA-35 or ISO/IEC 10646-1
- Shift functions according to ECMA-35

6.12 CS8 character set

The CS8 character set shall be the 53 characters in positions 02/01, 02/03 to 02/09, 02/13 to 02/14, 03/00 to 03/09, 04/00 to 05/10, 05/14 to 06/00, 07/11 and 07/13 to 07/14 of the IRV of ECMA-6.

NOTE

These characters are: EXCLAMATION MARK, NUMBER SIGN, DOLLAR SIGN, PERCENT SIGN, AMPERSAND, APOSTROPHE, LEFT PARENTHESIS, RIGHT PARENTHESIS, HYPHEN-MINUS, FULL STOP, DIGITs, LATIN CAPITAL LETTERs, CIRCUMFLEX ACCENT, LOW LINE, GRAVE ACCENT, LEFT CURLY BRACKET, RIGHT CURLY BRACKET, TILDE.

6.13 Source Name Space specification

The Source or originating system shall indicate the properties of path name information that is associated with a Source file or similar entity by the Field NAME SPACE. In some contexts other Fields may also use the same Name Space information, for example RESOURCE NAME SPACE in the File Set Header. The Name Space specifies the following characteristics of the file name:

- **Path element character set restrictions:** The characters used in the Source file path name shall be contained within this set.
- **Path element length restrictions:** Each element of the path name shall consist of at least one character, and shall not be longer than the number of characters specified.
- **Total path length restriction:** The entire path shall not be longer than the number of characters specified.
- **Path ordering (parent-first or child-first):** For complete paths in hierarchical file systems, if the parent-first path ordering is used, the parent node closest to the root shall be listed first, followed by its immediate child, and proceeding so forth to the path element furthest from the root. If the child-first path ordering is used, the node furthest from the root shall be listed first, followed by its immediate parent, and proceeding so forth to the path element nearest to the root.
- **Case-sensitive:** A CAPITAL LETTER is considered different from the corresponding SMALL LETTER.
- **Case retention:** A CAPITAL LETTER and the corresponding SMALL LETTER are stored and displayed distinctly.
- **Separators:** In the full path name, the separator(s) between any two elements of the path name.
- **Special requirements:** Any other characteristics not reflected above. For example, whether restrictions apply to the character FULL STOP (#2E), regarding its position or frequency of use in a path element name.

6.13.1 Name Space

Name Spaces shall be identified by an unsigned hexadecimal number. The Name Spaces specified by this ECMA Standard are shown in figure 4.

| Identifier | Name Space |
|------------|----------------------------------|
| #00 | The NS0 Name Space (see 6.14). |
| #01 | The NS1 Name Space (see 6.15). |
| #02 | The NS2 Name Space (see 6.16). |
| #03 | The NS3 Name Space (see 6.17). |
| #04 | The NS4 Name Space (see 6.18). |
| #05 | The NS5 Name Space (see 6.19). |
| #FFFFFFC | The NSFC Name Space (see 6.20). |
| #FFFFFFE | The NSFEE Name Space (see 6.21). |

Figure 4 - Defined Name Spaces

6.14 NS0 Name Space

Path element character set restrictions: Characters of the NS0 Name Space path elements shall be members of the CS8 Character Set.

Path element length restrictions: A path element shall contain at most 12 characters. A path element shall not contain more than one FULL STOP. A path element shall contain at most 8 characters before a FULL STOP and at most 3 characters after a FULL STOP. If a path element does not contain a FULL STOP, that path element shall contain at most 8 characters.

Total path length restriction: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete (i.e. not relative) path shall be recorded before the child element.

Case-sensitive: FALSE.

Case retention: FALSE.

Separators: The separator between the first two elements of a complete path shall be COLON. The separator between all subsequent elements of the path shall be SOLIDUS.

Special requirements: A path element shall not contain more than one FULL STOP character. The first element of a complete path shall be a Source volume Name.

NOTE

This Name Space may be associated with file systems of many personal computers. In some implementations the total path length is restricted to 256 characters, and the use of longer paths may yield unpredictable results. In some implementation the characters represented by #21, #23 to 29, #2D, #2E, #7D and #7E may not be supported, and their use may yield unpredictable results.

6.15 NS1 Name Space

Path element character set restrictions: Bytes #00 and #3A are not allowed for the NS1 Name Space path elements.

Path element length restriction: Each path element shall contain at most 32 characters.

Total path length restrictions: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete path shall be recorded before the child element.

Case-sensitive: TRUE.

Case retention: TRUE.

Separators: The separator between the first two elements of a complete path shall be two adjacent COLONs. The separator between all subsequent elements of the path shall be COLON.

Special requirements: The first element of a complete path shall be a Source volume Name.

6.16 NS2 Name Space

Path element character set restrictions: Characters of the NS2 Name Space path elements shall be represented by any single byte, except #00, #2F, #3A.

Path element length restriction: Each path element shall contain at most 300 characters.

Total path length restrictions: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete path shall be recorded before the child element.

Case-sensitive: TRUE.

Case retention: TRUE.

Separators: The separator between the first two elements of a fully qualified path shall be COLON. The separator between all subsequent elements of the path shall be SOLIDUS.

Special requirements: The first element of a complete path shall be a Source volume Name.

6.17 NS3 Name Space

Path element character set restrictions: Characters of the NS3 Name Space path elements shall be represented by any single byte within one of the following ranges: #01 to #2E, #30 to #39, #3B to #7E.

Path element length restriction: Each path element shall contain at most 255 characters.

Total path length restrictions: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete path shall be recorded before the child element.

Case-sensitive: TRUE.

Case retention: TRUE.

Separators: The separator between the first two elements of a complete path shall be COLON. The separator between all subsequent elements of the path shall be SOLIDUS.

Special requirements: The first element of a complete path shall be a Source volume Name.

6.18 NS4 Name Space

Path element character set restrictions: Characters of the NS4 Name Space path elements shall be represented by any single byte within one of the following ranges: #21 to #2E, #30 to #5B, #5D to #FF.

Path element length restriction: Each path element shall contain at most 300 characters.

Total path length restrictions: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete path shall be recorded before the child element.

Case-sensitive: FALSE.

Case retention: TRUE.

Separators: The separator between the first two elements of a complete path shall be COLON. The separator between all subsequent elements of the path shall be SOLIDUS.

Special requirements: The first element of a complete path shall be a Source volume Name.

6.19 NS5 Name Space

Path element character set restrictions: Characters of the NS5 Name Space path elements shall be represented by any single byte within one of the following ranges: #30 to #39, #40 to #5A, #5E to #60, #7B, #7D to #FF.

Path element length restriction: Each path element shall contain at most 256 characters.

Total path length restrictions: No total path length restrictions shall apply.

Path ordering: The parent path element(s) of a complete path shall be recorded before the child element.

Case-sensitive: FALSE.

Case retention: TRUE.

Separators: The separator between the first two elements of a complete path shall be COLON. The separator between all subsequent elements of the path shall be SOLIDUS.

Special requirements: The first element of a complete path shall be a Source volume Name.

NOTE

In some implementations the total path length is restricted to 256 characters, and use of longer paths may yield unpredictable results.

6.20 NSFC Name Space

Path element character set restrictions: Characters of the NSFC Name Space path elements shall be represented by any single byte within one of the following ranges: #0 to #2A, #2C to #2D, #2F to #3C, #3E to #5B, #5D to #FF.

Path element length restriction: Each path element shall contain at most 256 characters.

Total path length restrictions: The total path shall contain at most 256 characters.

Path ordering: The child element of a complete path shall be recorded before the parent path element(s).

Case-sensitive: FALSE.

Case retention: TRUE.

Separators: The separator between all path elements shall be FULL STOP.

6.21 NSFE Name Space

Name Space NSFE shall be used to indicate that the characteristics associated with the Name Space shall be determined by the Source, and are not specified by this ECMA Standard.

6.22 Strings

A string shall be recorded as a sequence of zero or more non-NULL characters, followed by a NULL character.

6.23 Resynchronization Pattern

The Resynchronization Pattern shall be the two-byte Byte Sequence #A55A.

7 Timestamp

A Timestamp specifies a date and time recorded in the format shown in figure 5. If the Year field has the value 0 the Timestamp shall be ignored; all other Fields in Timestamp shall be set to all ZEROs. Timestamps are established by the originating system and/or the Source. The accuracy of Timestamps is not specified by this ECMA Standard. If a Timestamp is recorded in a Field the Data part of which is longer than 12 bytes, the Timestamp shall be recorded first, and all subsequent bytes shall be recorded as NULL(s).

| RBP | Name | Representation |
|-----|------------------------------|----------------|
| 0 | Timestamp type and time zone | 16-bit integer |
| 2 | Year | 16-bit integer |
| 4 | Month | 8-bit integer |
| 5 | Day | 8-bit integer |
| 6 | Hour | 8-bit integer |
| 7 | Minute | 8-bit integer |
| 8 | Second | 8-bit integer |
| 9 | Centisecond | 8-bit integer |
| 10 | Hundreds of microseconds | 8-bit integer |
| 11 | Microseconds | 8-bit integer |

Figure 5 - Timestamp format

7.1 Type and Time Zone (RBP 0)

The most significant 4 bits of this integer, interpreted as a 4-bit number, shall specify the interpretation of the Timestamp.

- the value 0 shall indicate that the Timestamp specifies Coordinated Universal Time, timeZone shall be all ZEROs.
- the value 1 shall indicate that the Timestamp specifies local time.

- the value 2 shall indicate that the interpretation of the Timestamp is subject to agreement between the originator and recipient of the medium.
- no other value shall be recorded.

The least significant 12 bits of this field, interpreted as a signed 12-bit number in TWO's complement form, shall be interpreted as follows:

- if the value is in the range -1440 to 1440, then the value specifies the offset, in minutes, of the date and time from Coordinated Universal Time.
- if the value is -2047, this shall mean that no such offset is specified.
- no other values shall be recorded.

7.2 Year (RBP 2)

This field shall specify the year as a number in the range 1 to 9999.

7.3 Month (RBP 4)

This field shall specify the month of the year as a number in the range 1 to 12.

7.4 Day (RBP 5)

This field shall specify the day of the month as a number in the range 1 to 31.

7.5 Hour (RBP 6)

This field shall specify the hour of the day as a number in the range 0 to 23.

7.6 Minute (RBP 7)

This field shall specify the minute of the hour as a number in the range 0 to 59.

7.7 Second (RBP 8)

If the value of the Type field is 2, then this field shall specify the second of the minute as a number in the range 0 to 60. Otherwise, this field shall specify the second of the minute as a number in the range 0 to 59.

7.8 Centiseconds (RBP 9)

This field shall specify the hundredths of the second as a number in the range 0 to 99.

7.9 Hundreds of microseconds (RBP 10)

This field shall specify the hundreds of microseconds as a number in the range 0 to 99.

7.10 Microseconds (RBP 11)

This field shall specify the microseconds as a number in the range 0 to 99.

8 Requirements for a standard for recording

The media for recording shall meet the following requirements of a standard for recording.

The standard for recording shall specify:

- a unique physical address for each sector;
- the length of each sector;
- the means for determining whether a sector is rewritable or not;
- for media where sectors may only be recorded once, a means for detecting whether a sector has not yet been recorded;
- whether sectors or groups of sectors may require pre-processing prior to recording;
- a method for recording and detecting file marks, if file marks are supported.

The standard for recording used in conjunction with this ECMA Standard is subject to agreement between the originator and recipient of the medium.

9 Cyclic Redundancy Check (CRC)

A Cyclic Redundancy Check (CRC), where implemented, shall be computed in accordance with ITU Rec. X.25. CRC shall use the generating polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. CRC shall use the seed value -1. CRC generates a 32-bit output.

10 Organization of Information on a Volume

10.1 Recording of Sectors

The physical sectors of a Volume shall be organized into logical Sectors of equal length. Sectors shall have a length of 2^{n+8} bytes, where n is a positive integer (i.e. Sector lengths of 512, 1024, 2048, 4096, ...). Each Sector of a Volume shall be identified by a unique Sector Number. Sector Number 0 shall be assigned to the Sector at the start of the Volume Header. This is not necessarily the sector with the lowest physical address of the partition.

10.2 Organization of a Volume

A Volume is a set of Sectors. The lowest numbered Sectors of the Volume, referred to as the Volume Preamble, shall contain only a Volume Header and zero or more File Set Continuation Headers. The remaining Sectors of the Volume, up to, but not including, the Volume Postamble, shall be referred to as the Volume Data Space of the Volume. Volume Data Space shall consist of File Sets (see clause 11) and, optionally, Volume Indices (see 13.5).

The Volume Postamble may contain one File Set Subindex and/or one Volume Set Subindex. If it contains both, the File Set Subindex shall precede the Volume Set Subindex. The Volume Postamble shall end with the Volume Terminator. The Volume Terminator shall be either a Volume Trailer, the physical end-of-partition, or the end of recorded data, which ever occurs at the lower numbered Sector. No more than one Volume shall be recorded in one physical partition.

NOTE

Because a Volume, as specified by this ECMA Standard, need not be located at the physical start of the partition, a Volume may be preceded by other information. The presence of such prefixed information is restricted to media at Level 2 of medium interchange (see 14.2).

Figure 6 summarizes the requirements for recording Volumes.

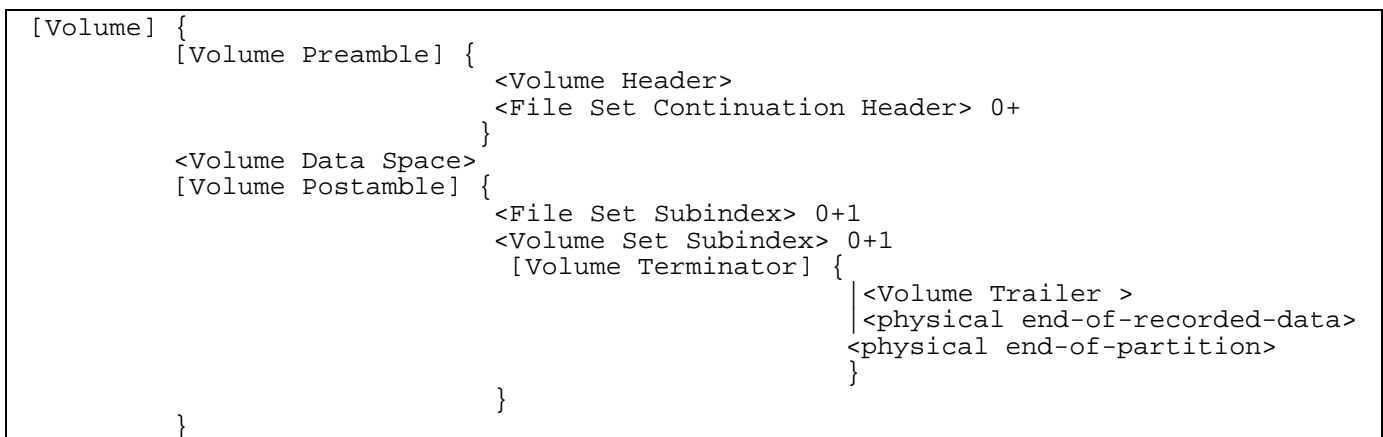


Figure 6 - Schema for Volumes

10.3 Volume Sets

A Volume Set shall consist of one or more Volumes having a Volume Set Label common to all Volumes.

The Volumes in a Volume Set shall be numbered ordinally (1, 2, 3, ..., n), their numbers constitute their Volume Set Sequence Numbers. All occurrences of Volume Set Time within a Volume Set shall be equivalent. The ordered concatenation of the Volume Data Spaces of the Volumes of a Volumes Set constitutes the Volume Set Data Space.

NOTE

Volumes within a Volume Set may be of different physical media types.

File Set(s) shall be recorded within the Volume Set Data Space of a Volume Set. A Volume Index is optional, but if present shall be recorded within the Volume Set Data Space, immediately following a File Set.

Figure 7 summarizes the requirements for recording Volume Set Data Space.

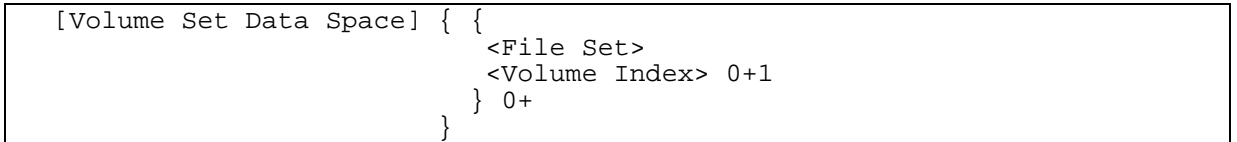


Figure 7 - Schema for Volume Set Data Space

Figure 8 to figure 10 show examples of the layout of the Volume organization.

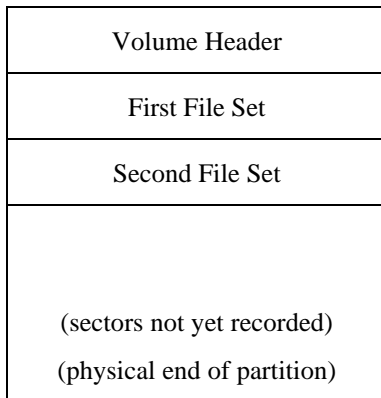


Figure 8 - Simplified view of a Volume

Two File Sets are recorded on the Volume, further File Sets can be appended.

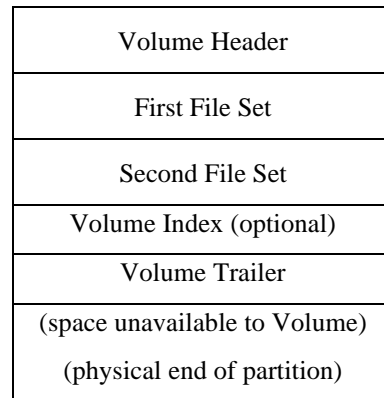


Figure 9 - Simplified view of a Volume

Two File Sets are recorded on the Volume, no further File Sets can be appended without overwriting the Volume Trailer.

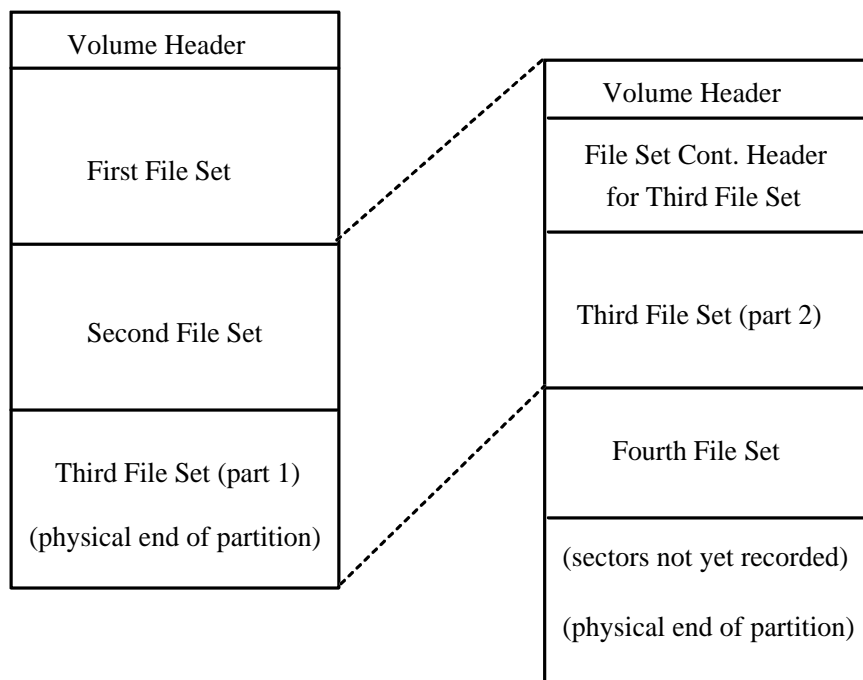


Figure 10 - Simplified view of a Volume Set, the Third File Set of which spans from one Volume to the next.

10.4 Recording of Fields

All data in the Volume, except for Streams (see 13.15.6), shall be expressed in Fields. A Field shall consist of a Field Identifier (FID) (see annex A), and may contain a Data Length part and a Data part.

Certain FIDs specify the length of the data implicitly, in which case the Data Length part shall be omitted. For all other FIDs, the Data Length part shall be present immediately after the FID. If data is present in the Field, the Data part of the Field shall immediately follow the Data Length part if present, or shall follow the FID if the Data Length part is omitted. The formal definition of each type of Field is specified in annex C.

| | | |
|-----|-------------|------|
| FID | Data length | Data |
|-----|-------------|------|

Figure 11 - Parts of a Field

Fields contained in Buffer(s) shall have both their FID and Data length part recorded entirely within a single Buffer.

10.4.1 NULL Field

The NULL Field shall consist only of its FID which consists of the single byte #00. It has no meaning, and functions merely as a placeholder.

10.5 Recording of Field Tables

A Field Table is a sequence of Fields. All Fields shall be recorded in Field Tables, except when the Null Field is used as Blank Space (see 13.3). A Field Table shall start and end with Fields identified by the same FID. The Data part of the first Field shall consist of the Resynchronization Pattern. The Data part of the last Field shall either contain CRC bits or be empty, as indicated by the Data Length part of this last Field. This Field shall not appear elsewhere in the Field Table. Each Field Table is given the name of its first Field. If the OFFSET TO END Field is included in a Field Table, it shall be the second Field.

Any set of Fields specified as an Iterated Field Set may be repeated within the Field Table. Each iteration is identified by the first Field listed in the Iterated Field Set.

Except for the first and last Field, OFFSET TO END, and any restrictions on Iterated Field Sets, or otherwise specified, Fields may be in any order within the Field Table; they need not be in the order shown in examples.

If the CRC bits are computed, they shall be calculated over the whole Field Table, except the last Field.

10.6 Recording of Buffers

All information recorded in a Volume, except

- Volume Header
- File Set Continuation Header(s)
- File Set Header(s)
- File Set Trailer(s)
- file mark(s)
- and Volume Trailer

shall be recorded within Buffers. A Buffer is a set of one or more contiguous Sectors. Buffers shall begin and end at Sector boundaries. A Buffer shall begin with a Buffer Header Field Table immediately followed by one of the following types of data: File, File Set Index, File Set Subindex, Volume Index, Volume Subindex, or implementation-use. If the above data does not occupy the entire Buffer, the remainder shall be padded with Blank Space (see 13.3).

The part of the Buffer between the Buffer Header and any ending Blank Space shall be referred to as Buffer Data Space.

A Buffer shall be contained entirely within one Volume. If a Buffer cannot be recorded in its entirety within the remaining space of a Volume, it shall be recorded again in its entirety on the next Volume of the Volume Set immediately after the Volume Preamble of the next Volume (see 11.2). A Buffer shall not be repeated in any other instance.

Figure 12 summarizes the requirements for recording Buffers.


```

[Buffer] {
  <Buffer Header>
  [Buffer Data Space] {
    [File Data Space]
      {
        [File Continuation Space] {
          <File Continuation Header>
          <File data>
        } 0+1
      }
      [File Space]
        {
          <File Header>
          <File data>
        } 0+
      }
    [Volume or File Set (Sub)Index Data] |
    [implementation use] |
  }
  <Blank Space> 0+1
}

```

Figure 12 - Schema for a Buffer

10.7 Recording of file marks

The use of file marks shall be specified for each Volume by the Volume Header. The FILE MARK USAGE Field of the Volume Header specifies whether file marks shall be recorded

- before File Set Headers (see 11.1), except those that immediately follow either the Volume Preamble or other File Set Header(s) that immediately follow the Volume Preamble, and/or
- before File Set Trailers (see 11.1), and/or
- at specified regular intervals (see annex C: FILE MARK INTERVAL).

Once established by the originating system which records the Volume Header, file mark usage shall be maintained by any system(s) that record subsequent data in the Volume.

A file mark shall occupy one Sector. Contiguous file marks shall not be recorded. Other types of media marks, for example set marks, shall not be recorded. A file mark shall not be recorded within a Volume in any location other than those locations specified in the FILE MARK USAGE Field of the Volume Header.

11 Organization of Information within a File Set

11.1 Organization of a File Set

This ECMA Standard specifies a recording methodology for information interchange and secondary data storage. As such, the data recorded in a File Set may be considered a representation of information of all or part of a primary storage file system, referred to as the Source. Any group of Files from a single Source may be recorded together as a File Set. A File Set shall reside within the Volume Set Data Space. A File Set shall begin with a File Set Preamble, contain zero or more Buffers, and end with a File Set Postamble.

The File Set Preamble shall contain only the File Set Header, and a file mark when required by the Volume Header, as specified in 10.7.

The Volume Header of the Volume containing the start of the File Set Postamble shall specify whether a file mark is to be recorded before the File Set Trailer. If so, the Postamble shall begin with a file mark and the File Set Trailer shall be recorded in the next Sector of the Volume. If not, the Postamble shall begin with the File Set Trailer. The File Set Postamble shall contain nothing after the File Set Trailer, unless the File Set Header specifies that a File Set Index shall be recorded. In this case the File Set Index shall be recorded, within Buffer(s), in the next Sectors of the Volume Set Data Space. If the end of the Volume Data Space containing the File Set Trailer is encountered before the remainder of the File Set Postamble (i.e. File Set Index Buffers) can be completely recorded, the Volume Data Space shall span onto the next Volume in the Volume Set. In that case, the remainder of the File Set Postamble shall be recorded at the beginning of the next Volume Data Space, immediately after its Volume Preamble.

Figure 13 summarizes the requirements for recording File Sets.

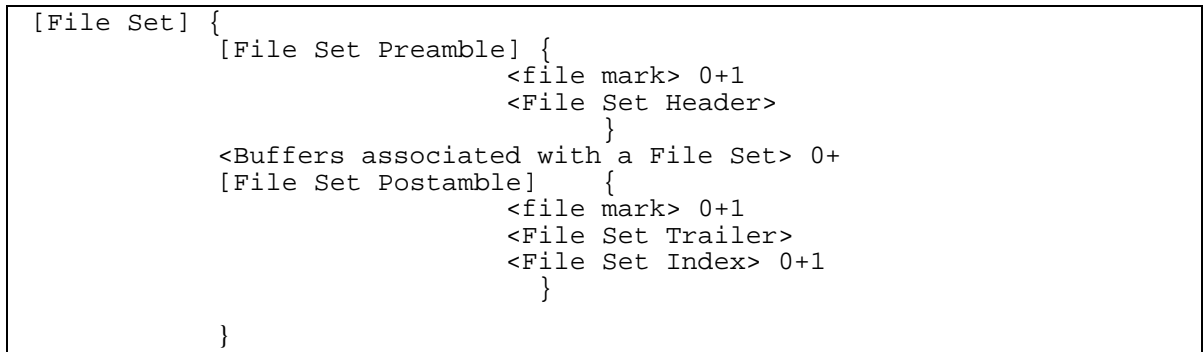


Figure 13 - Schema for File Sets

File Set Indices are optional. A File Set Index shall list all Files contained in the File Set, with their Volume and Sector Number. The File Set Index shall be contained within one or more Buffers.

File Sets that span multiple Volumes shall meet the requirements of 11.2. A File Set Preamble, the File Set Buffer(s) and the File Set Postamble need not be physically adjacent to each other. This allows interleaving of different File Sets.

A File Set Header, Continuation Headers, Buffers and Trailer shall all have the same File Set ID, File Set Time, File Set Label, Source Name Type, Source Name, Source operating system, and Source operating system version information, if recorded. If one or more Registered Identifiers are recorded in a File Set Header, they shall be re-recorded in all File Set Continuation Headers of that File Set.

Figures 14 and 15 show examples of the layout of the File Set organization.

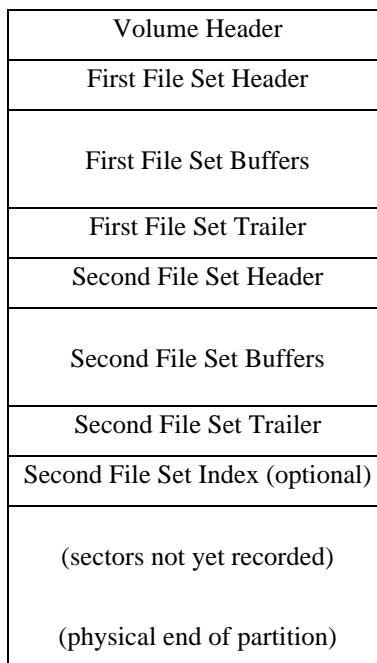


Figure 14 - Simplified view of a Volume containing two File Sets

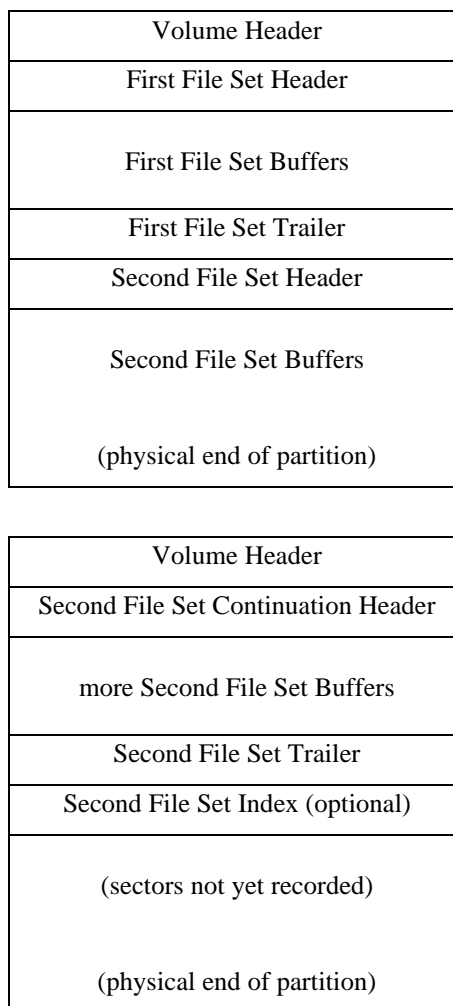


Figure 15 - Simplified view of a Volume Set containing two File Sets. The second File Set spans from the first Volume to the second Volume.

11.2 File Sets spanning multiple Volumes

A File Set shall be considered to span multiple Volumes if parts of the File Set are recorded on two or more Volumes of a Volume Set.

The Volume Set Sequence Numbers of all Volumes of the File Set shall form a consecutive ascending sequence. If the originating system deems that a Buffer of a File Set could not be written in its entirety before the end of a Volume, the entire Buffer shall be recorded immediately after the Volume Preamble of the Volume having the next greater Volume Set Sequence Number.

NOTE

In case of spanned Volumes a Buffer may be completely duplicated on the next Volume. The BUFFER SEQUENCE is the same in both instances. In this case the Buffer on the previous Volume is ignored.

A File Set Continuation Header shall be recorded within the Volume Preamble of all Volumes of the Volume Set after the Volume containing the File Set Header, up to and including the Volume containing the last Sector of the File Set Postamble.

11.3 Interleaving

Interleaving of File Set Preambles, File Set Postambles and File Set Buffers from different File Sets within the Volume Data Space is allowed. The FILE SET ID in each of these File Set entities indicates the File Set to which the entity belongs. The elements within each of these entities shall be logically contiguous.

The Buffers of a Volume Index shall be logically contiguous. A Volume Index shall be logically contiguous with the File Set Postamble that precedes it.

BUFFER SEQUENCE numbers shall be increased on an intra-File Set basis. Thus the interleaving of Buffers from other File Sets does not affect the BUFFER SEQUENCE numbers within a File Set. BUFFER ADDRESSes shall be maintained on an inter-File Set basis. Thus the interleaving of Buffers from other File Sets affects the addresses of subsequent Buffers of all File Sets.

Once a File Set Header has been recorded, the File Set shall be considered pending until the File Set Postamble is recorded. If any interleaved File Sets are pending in a Volume, File Set Continuation Headers for all such File Sets shall be recorded in the Volume Preamble of the next Volume of the Volume Set immediately following the Volume Header.

If a File Set has a File Set Index recorded, the Index shall be recorded immediately after the File Set Trailer within the Volume Set Data Space. Thus, no interleaving occurs within a File Set Postamble.

12 Organization of information within a File

Files and their Headers are contained within Buffer Data Space. A File shall consist of File Information and File Data, pertaining to a single entity from the Source. Each File shall be preceded by a File Header which specifies the type of file and the number of bytes of the File contained in the current Buffer. If the File does not fit entirely within a Buffer, it shall continue in the Buffer Data Space of the subsequent Buffer. In that Buffer, after the Buffer Header, a File Continuation Header shall be recorded, followed by more data of the File. This shall continue as needed until the entire File is recorded.

File Information shall be recorded as a single Field Table, specifying the attributes of the File. File Data (see 13.15) shall be recorded as a series of Field Tables, including path, characteristics, and the various Streams of the File. Additional Field Tables associated with the File may be recorded within the File Data.

Figure 16 summarizes the requirements for recording Files.

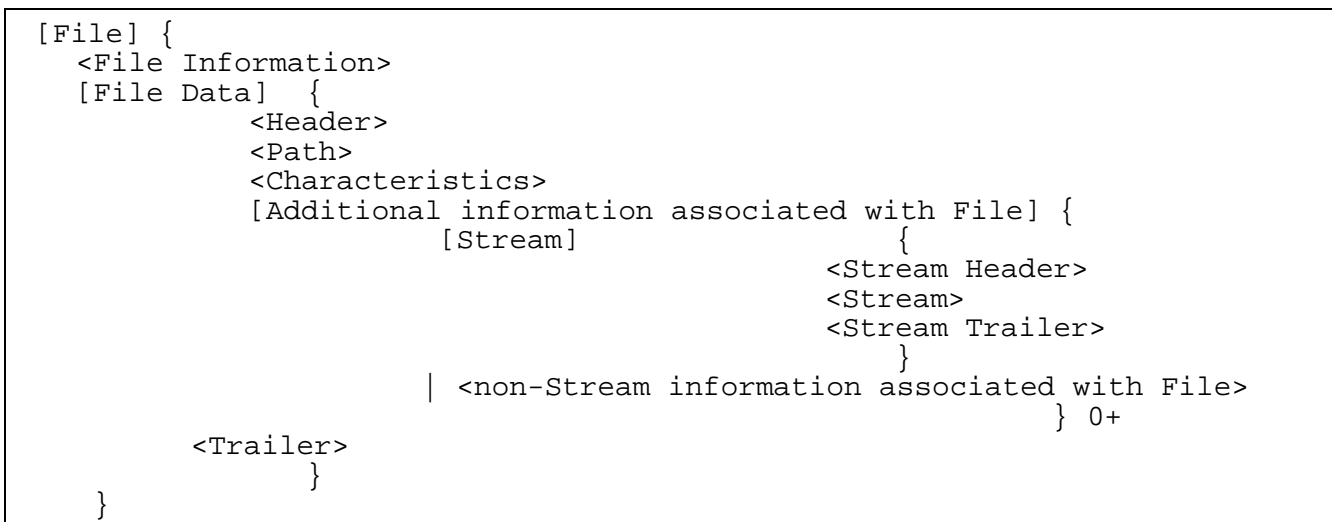


Figure 16 - Schema for Files

13 Field Table description

Field Tables consist of a number of mandatory Fields and any number of additional Fields deemed necessary for the application considered. In the following description, the mandatory Fields are specified. An example of a Field Table with additional Fields is given for illustration purposes. For each Field is indicated whether it is mandatory (Yes), not mandatory (No), or mandatory depending on a condition (Cond.).

13.1 Volume Header Field Table

In each Volume, the Volume Header Field Table is mandatory and shall be recorded beginning at the start of the first Sector of the Volume. This Sector need not be physically the first sector in the media partition. The Sector in which the Volume Header begins shall have Sector Number 0. If the Volume Header Field Table does not end exactly at a Sector boundary, the unused part of its last sector shall be padded with Blank Space.

The following ten Fields are mandatory in the Volume Header Field Table:

- VOLUME HEADER
- OFFSET TO END
- FORMAT VERSION
- SECTOR SIZE
- VOLUME SET TIME
- VOLUME TIME
- VOLUME SET LABEL
- VOLUME SET SEQUENCE
- VOLUME INDEX REQUIRED
- FILE MARK USAGE

If bit b2 in FILE MARK USAGE is set to ONE, signifying that file marks are to be used at regular intervals, the FILE MARK INTERVAL Field shall be present.

Figure 17 shows an example of a Volume Header Field Table.

| Field Name | FID | Mandatory | Data Description |
|---------------------------|---------|-----------|--|
| VOLUME HEADER | #808000 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | Yes | Bytes from start of next Field to start of last Field |
| FORMAT NAME | #8052 | No | The bytes sequence "SIDF" |
| FORMAT VERSION | #8062 | Yes | Major, minor and subminor version number |
| SECTOR SIZE | #80800E | Yes | 512, 1024, 2048, 4096, ... bytes |
| VOLUME SET TIME | #80F400 | Yes | Timestamp of Volume Set creation |
| VOLUME TIME | #80F401 | Yes | Timestamp of Volume creation |
| CHAR SPEC | #808040 | No | Character type specification |
| REGISTERED IDENTIFIER | #808043 | No | ID for extension to Standard, scope: this Volume |
| VOLUME SET LABEL | #808030 | Yes | User-defined string, same for all Volumes in Set |
| VOLUME SET SEQUENCE | #80F100 | Yes | Position of Volume within the Volume Set (1,2,3 ...) |
| CHAR SPEC | #808040 | No 1) | Character type specification |
| VOLUME SET ALIAS | #808041 | No 1) | User-defined string |
| VOLUME LABEL | #808027 | No | Implementation use string |
| PARTITION NUMBER | #808012 | No | Physical partition number of this Volume |
| VOLUME SIZE | #80F201 | No | Capacity in sectors of the Volume (informative) |
| VOLUME INDEX REQUIRED | #80802F | Yes | Whether Volume Index is always at EOD |
| VOLUME INDEX LOCATION | #808042 | No | Sector Number of last Volume Index on Volume |
| FILE MARK USAGE | #808020 | Yes | Where file marks are recorded |
| FILE MARK INTERVAL | #808028 | Cond. | Sectors between file marks recorded at regular intervals |
| VOLUME HEADER WRITE COUNT | #80F104 | No | Number of times a Volume Header has been written to this physical partition. |
| AUTHENTICATION | #8002 | No | Volume password |
| EXPIRATION TIME | #80F404 | No | Timestamp of Volume expiration (informative) |
| DEVICE INFO | #808032 | No | User-defined string |
| VOLUME HEADER | #808000 | Yes | CRC value or empty |

Note to figure 17

1) These Fields form an Iterated Field Set. This set may be used to specify zero or more VOLUME SET ALIASes which may utilize differing Character Specifications. It is recommended that the use of VOLUME SET ALIASes be consistent across a Volume Set.

Figure 17 - Example of a Volume Header Field Table

13.2 Volume Trailer Field Table

In each Volume, the Volume Trailer Field Table is not mandatory. If present, it shall be recorded beginning at the start of the last Sector of the Volume. If the Volume Trailer Field Table does not end exactly at a Sector boundary, the unused part of its last sector shall be padded with Blank Space.

The VOLUME TRAILER Field is the only mandatory Field of the Volume Trailer Field Table. Figure 18 shows an example of a Volume Trailer Field Table.

| Field Name | FID | Mandatory | Data Description |
|----------------|---------|-----------|---------------------------------------|
| VOLUME TRAILER | #808003 | Yes | Resynchronization Pattern |
| CLOSE TIME | #80F402 | No | Timestamp when Volume Trailer written |
| VOLUME TRAILER | #808003 | Yes | CRC value or empty |

Figure 18 - Example of a Volume Trailer Field Table

13.3 Blank Space Field Table

In each Volume, the Blank Space Field Table is not mandatory. If present, it shall be used, space permitting:

- after all Field Tables not contained in Buffers, to pad to the next sector boundary, and
- at the end of Buffers.

The BLANK SPACE Field and the OFFSET TO END Field are mandatory in the Blank Space Field Table. The only additional Field allowed in the Blank Space Field Table is the NULL Field. Figure 19 shows an example of a Blank Space Field Table.

| Field Name | FID | Mandatory | Data Description |
|---------------|---------|-----------|---|
| BLANK SPACE | #808019 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | Yes | Bytes from start of next Field to start of last Field |
| NULL Field | #00 | Cond. | Empty |
| BLANK SPACE | #808019 | Yes | CRC value or empty |

Figure 19 - Example of a Blank Space Field Table

13.4 Buffer Header Field Table

In each Volume, the Buffer Header Field Table is mandatory and shall be recorded at the start of each Buffer, it specifies characteristics of that Buffer. The following six Fields are mandatory in the Buffer Header Field Table:

- BUFFER HEADER
- OFFSET TO END
- BUFFER TYPE
- BUFFER SIZE
- BUFFER SEQUENCE
- UNUSED IN THIS BUFFER

If the BUFFER TYPE Field is set to File, the Buffer Header Field Table shall contain the BUFFER ADDRESS Field.

If the BUFFER TYPE Field is set to either File, or File Set (Sub)Index, the Buffer Header Field Table shall contain the FILE SET ID Field and the FILE SET TIME Field.

The BUFFER SIZE shall be no larger than the length specified by BUFFER SIZE in the File Set Header.

The BUFFER SEQUENCE Field shall be set to 1 for the first Buffer of the File Set following the File Set Header. The BUFFER SEQUENCE Field shall be increased by 1 for each subsequent Buffer of that File Set except in the following case: if the originating system deems that a Buffer cannot be recorded in its entirety before the end of the Volume, the Buffer shall be recorded in its entirety at the start of the Volume Set Data Space of the next Volume in the Volume Set (see 11.2). Nothing shall be recorded in a Volume Data Space before a Buffer that is repeated.

If a File Set (Sub)Index exists for the File Set, the BUFFER SEQUENCE Field shall be further increased by 1 for each Buffer of the (Sub)Index. If a File Set Buffer follows a File Set Subindex, the BUFFER SEQUENCE Field shall be further increased by 1 for each File Set Buffer.

If the BUFFER TYPE Field is set to Volume (Sub)Index, it shall be associated only with any other Buffers containing data from the same Volume (Sub)Index, by satisfying the following requirements.

- The BUFFER SEQUENCE Field shall be set to 1 for the first Buffer of the Volume (Sub)Index. Each subsequent Buffer of that Volume (Sub)Index shall contain a BUFFER SEQUENCE Field increased by 1.
- The BUFFER ADDRESS, FILE SET ID, FILE SET TIME, FILE SET LABEL, SOURCE NAME TYPE, SOURCE NAME, SOURCE OPERATING SYSTEM, and SOURCE OPERATING SYSTEM VERSION Fields shall not be present.

Figure 20 shows an example of a Buffer Header Field Table.

| Field Name | FID | Mandatory | Data Description |
|-------------------------------------|---------|-----------|--|
| BUFFER HEADER | #05 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | Yes | Bytes from start of next Field to start of last Field |
| BUFFER TYPE | #60 | Yes | Buffer data type (File, File Set Index ...) |
| BUFFER SIZE | #06 | Yes | Buffer size in bytes, incl. Buffer Header |
| BUFFER SEQUENCE | #07 | Yes | Buffer Sequence number within File Set or Volume Index |
| BUFFER ADDRESS | #08 | Cond. | Number of sectors counting from File Set (Sub) Header |
| BUFFER CRC | #8008 | No | CRC, not including Buffer Header |
| UNUSED IN THIS BUFFER | #8000 | Yes | Length of unused Blank Space |
| FILE SET ID | #8072 | Cond. | ID of associated File Set and Buffer(s) |
| FILE SET TIME | #80F403 | Cond. | Timestamp in Buffer of File Set |
| FILE SET LABEL | #808005 | No | User-defined label of File Set |
| ORIGINATING SYSTEM SOFTWARE NAME | #808006 | No | Originating application name string |
| ORIGINATING SYSTEM SOFTWARE TYPE | #808007 | No | Originating application type string |
| ORIGINATING SYSTEM SOFTWARE VERSION | #808008 | No | Originating application version string |
| SOURCE NAME TYPE | #8009 | No | Source naming convention type |
| SOURCE NAME | #02 | No | Source name string |
| SOURCE OPERATING SYSTEM | #03 | No | Source operating system string |
| SOURCE OPERATING SYSTEM VERSION | #04 | No | Source operating system version string |
| VOLUME SET TIME | #80F400 | No | Timestamp of Volume Set creation |
| VOLUME SET LABEL | #808030 | No | User-defined string |
| VOLUME SET SEQUENCE | #80F100 | No | Volume position within Volume Set |
| BUFFER HEADER | #05 | Yes | CRC value or empty |

Figure 20 - Example of a Buffer Header Field Table

13.5 Volume Index Field Table

In each Volume, the Volume Index Field Table is not mandatory. If present, it shall be recorded within the Volume Set Data Space immediately after a File Set Postamble. A Volume Index stores information about File Sets previously recorded in the Volume or Volume Set. The VOLUME INDEX and NUMBER OF FILE SETS Fields are mandatory in Volume Index Field Tables.

If NUMBER OF FILE SETS is greater than 0, the following eleven additional Fields are also mandatory in the Volume Index Field Table for each File Set:

FILE SET ID
FILE SET TIME
FILE SET LABEL
SOURCE NAME TYPE
SOURCE NAME
SOURCE OPERATING SYSTEM
SOURCE OPERATING SYSTEM VERSION
FSH VOLUME SET SEQUENCE
FILE SET HEADER LOCATION
FST VOLUME SET SEQUENCE
FILE SET TRAILER LOCATION

A Volume Index is optional. If a Volume Index Field Table is written it shall conform to the following requirements.

- A Volume Index Field Table shall be contained in one or more Buffers, which are referred to below as the Volume Index Buffer(s). If the Volume Index Field Table is too large to fit into a single Buffer, it shall be continued into subsequent Buffer(s) which shall be recorded contiguously within the Volume Set Data Space.
- If VOLUME INDEX REQUIRED is set to a bit pattern other than all ZEROs in the Volume Header Field Table, and the Volume is not full, a Volume Index Buffer shall be the last information recorded in the Volume. If VOLUME INDEX REQUIRED is set to all ZEROs, Volume Indices may still be recorded in the location(s) specified above.
- A Volume Index Field Table shall reference all File Sets wholly or partially contained previously in the Volume, and may reference any File Set contained previously in the Volume Set.
- A Volume Index Field Table shall contain all of the File Set-specific information contained in all previous Volume Indices of the Volume.
- The Fields of a Volume Index Field Table shall be associated with either the Volume Index Field Table in general, or a specific File Set. Fields recorded before the first instance of the FILE SET ID Field shall be associated with the Volume Index Field Table in general. Fields in the Table, except the last Field, recorded after the n-th instance of the FILE SET ID Field, shall be associated with the n-th File Set referenced by the Volume Index Field Table.
- The NUMBER OF FILE SETS Field shall specify the number of times that the Iterated Field Set is repeated. An Iteration shall include all Fields starting with the FILE SET ID Field and ending immediately before another occurrence of the FILE SET ID Field or the final VOLUME INDEX Field. All Fields, including non-required Fields, within the nth Iteration shall reference the File Set specified by the nth occurrence of the FILE SET ID Field.
- File Set-specific Iterated Field Sets shall be recorded in the order that their File Set Trailers are recorded within the Volume Set.

Figure 21 shows an example of a Volume Index Field Table.

| Field Name | FID | Mandatory | Data Description |
|---------------------------------|---------|-----------|---|
| VOLUME INDEX | #808011 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| NUMBER OF FILE SETS | #808015 | Yes | Number of File Set entries iterated below |
| FILE SET ID | #8072 | Cond. 1) | File Set 4-byte ID |
| FILE SET TIME | #80F403 | Cond. | File Set Timestamp |
| FILE SET LABEL | #808005 | Cond. | File Set user-defined label string |
| SOURCE NAME TYPE | #8009 | Cond. | Source naming type |
| SOURCE NAME | #02 | Cond. | Source name string |
| SOURCE OPERATING SYSTEM | #03 | Cond. | Source operating system string |
| SOURCE OPERATING SYSTEM VERSION | #04 | Cond. | Source operating system version string |
| FILE SET ABORTED | #808039 | No | Whether File Set has been aborted prematurely |
| RESOURCE NAME SPACE | #808038 | No | Enumeration, see NAME SPACE |
| RESOURCE NAME | #808023 | No | Root entity stored in the File Set |
| RESOURCE TYPE | #808037 | No | Enumeration |
| FSH VOLUME SET SEQUENCE | #80803A | Cond. | Volume Set sequence number of File Set Header |
| FSH PARTITION NUMBER | #80803B | No | Physical partition number of File Set Header |
| FILE SET HEADER LOCATION | #80803C | Cond. | Sector Number of File Set Header |
| FST VOLUME SET SEQUENCE | #80803D | Cond. | Volume Set sequence number of File Set Trailer |
| FST PARTITION NUMBER | #80803E | No | Physical partition number of File Set Trailer |
| FILE SET TRAILER LOCATION | #80803F | Cond. | Sector Number of File Set Trailer |
| VOLUME INDEX | #808011 | Yes | CRC value or empty |

Note to figure 21

1) This Field shall be recorded first within the Iterated Field Set.

Figure 21 - Example of a Volume Index Field Table

13.6 Volume Subindex Field Table

In each Volume, the Volume Subindex Field Table is not mandatory. If present, it shall be recorded only within a Volume Postamble. A Volume Subindex Field Table shall be contained in Buffers, and recorded within the Buffer Data Space. The Volume Index Buffer(s) shall immediately precede the Volume Terminator.

The requirements for the recording of the Volume Subindex Field Table shall be the same as for the Volume Index Field Table, except that the first and last Field shall be the VOLUME SUBINDEX Field (FID = #808031) instead of the VOLUME INDEX Field.

13.7 File Set Header Field Table

In each Volume, the File Set header Field Table is not mandatory, unless the File Set starts within this Volume. If present, it shall be recorded at the start of the first Sector of the File Set. If the File Set Header Field Table does not end exactly at a Sector boundary, the unused part of its last Sector shall be padded with Blank Space.

The following eleven Fields are mandatory in the File Set Header Field Table:

FILE SET HEADER
OFFSET TO END
FILE SET ID
FILE SET TIME
FILE SET LABEL
FILE SET INDEX PRESENT
BUFFER SIZE
SOURCE NAME TYPE
SOURCE NAME
SOURCE OPERATING SYSTEM
SOURCE OPERATING SYSTEM VERSION

SOURCE NAME TYPE shall specify the naming convention used for all subsequent instances of SOURCE NAME and SOURCE ALIAS in this File Set Header Field Table, unless and until another instance of SOURCE NAME TYPE is recorded later in the File Set Header.

SOURCE ALIAS, and SOURCE NAME TYPE if recorded immediately before SOURCE ALIAS, shall form an Iterated Field Set. SOURCE NAME TYPE may be omitted if its data is the same as that of the previous instance of SOURCE NAME TYPE in this File Set Header.

Figure 23 shows an example of a File Set Header Field Table.

| Field Name | FID | Mandatory | Data Description |
|-------------------------------------|---------|-----------|---|
| FILE SET HEADER | #808004 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | Yes | Bytes from start of next Field to start of last Field |
| PREV OBJECT LOCATION | #80802A | No | Sector Number of the previous FS Header or FS Trailer, if in Volume |
| NEXT OBJECT LOCATION | #808029 | No | Sector Number of the next FS Header or FS Trailer, if in Volume |
| CHAR SPEC | #808040 | No | Character Specification default for File Set |
| FILE SET ID | #8072 | Yes | File Set 4-byte ID |
| FILE SET TIME | #80F403 | Yes | File Set Timestamp |
| FILE SET LABEL | #808005 | Yes | User-defined File Set label string |
| FILE SET INDEX PRESENT | #80802D | Yes | Whether the File Set Index is recorded in this File |
| REGISTERED IDENTIFIER | #808043 | No | ID for extension to Standard, scope: this File Set only |
| BUFFER SIZE | #06 | Yes | Maximum Buffer size permissible in the File Set |
| SOURCE NAME TYPE | #8009 | Yes | Source naming convention type |
| SOURCE NAME | #02 | Yes | Source name string |
| SOURCE OPERATING SYSTEM | #03 | Yes | Source operating system string |
| SOURCE OPERATING SYSTEM VERSION | #04 | Yes | Source operating system version string |
| SOURCE NAME TYPE | #8009 | Cond. 1) | Source alias naming convention type |
| SOURCE ALIAS | #808036 | No 1) | Alternate name for Source |
| RESOURCE NAME SPACE | #808038 | No | Enumeration, see NAME SPACE in annex C |
| RESOURCE NAME | #808023 | No | Root File stored in the File Set |
| RESOURCE TYPE | #808037 | No | Enumeration |
| ORIGINATING SYSTEM SOFTWARE NAME | #808006 | No | Originating application name string |
| ORIGINATING SYSTEM SOFTWARE TYPE | #808007 | No | Originating application type string |
| ORIGINATING SYSTEM SOFTWARE VERSION | #808008 | No | Originating application version string |
| FORMAT VERSION | #8062 | No | Major, minor, subminor version: 4 bytes |
| AUTHENTICATION | #8002 | No | File Set Password |
| FILE SET COMMENT | #80802B | No | User-defined string |
| FILE SET HEADER | #808004 | Yes | CRC value or empty |

Note to figure 22

1) These Fields form an Iterated Field Set. SOURCE NAME TYPE is not required unless its data is different from a previous instance in this File Set Header. Zero or more iterations shall be recorded, based on the number of SOURCE ALIASes to write.

Figure 22 - Example of a File Set Header Field Table

13.8 File Set Continuation Header Field Table

In each Volume, the File Set Continuation Header Field Table is not mandatory, unless the File Set spans from the previous Volume. If present, it shall be recorded only within the Volume Preamble, after the Volume Header Field Table. If the File Set Continuation Header Field Table does not end exactly at a Sector boundary, the unused part of its last Sector shall be padded with Blank Space.

The requirements for the recording of the File Set Continuation Header Field Table shall be the same as for the File Set Header Field Table, except that the first and last Field shall be the FILE SET CONTINUATION HEADER Field (FID = #808035), instead of the FILE SET HEADER Field.

13.9 File Set Trailer Field Table

In each Volume, the File Set Trailer Field Table is not mandatory, unless the File Set terminates within this Volume. If present, it shall be recorded after the last Buffer containing the File Set File data. The File Set Trailer Field Table shall begin on a Sector Boundary. If the File Set Header Field Table does not end exactly at a Sector boundary, the unused part of its last Sector shall be padded with Blank Space.

The following nine Fields are mandatory in the File Set Trailer Field Table:

- FILE SET TRAILER
- OFFSET TO END
- FILE SET ID
- FILE SET TIME
- FILE SET LABEL
- SOURCE NAME TYPE
- SOURCE NAME
- SOURCE OPERATING SYSTEM
- SOURCE OPERATING SYSTEM VERSION

Figure 23 shows an example of a File Set Trailer Field Table.

| Field Name | FID | Mandatory | Data Description |
|-------------------------------------|---------|-----------|---|
| FILE SET TRAILER | #808009 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | Yes | Bytes from start of next Field to start of last Field |
| PREV OBJECT LOCATION | #80802A | No | Sector Number of the previous FS Header or FS Trailer, if in Volume |
| NEXT OBJECT LOCATION | #808029 | No | Sector Number of the next FS Header or FS Trailer, if in Volume |
| FILE SET ABORTED | #808039 | No | Whether the File Set is aborted |
| FILE SET ID | #8072 | Yes | File Set 4-byte ID |
| FILE SET TIME | #80F403 | Yes | File Set Timestamp |
| FILE SET LABEL | #808005 | Yes | User-defined File Set label string |
| SOURCE NAME TYPE | #8009 | Yes | Source naming convention type |
| SOURCE NAME | #02 | Yes | Source name string |
| SOURCE OPERATING SYSTEM | #03 | Yes | Source operating system string |
| SOURCE OPERATING SYSTEM VERSION | #04 | Yes | Source operating system version string |
| RESOURCE NAME SPACE | #808038 | No | Enumeration, see NAME SPACE in annex C |
| RESOURCE NAME | #808023 | No | Root File stored in File Set |
| RESOURCE TYPE | #808037 | No | Enumeration |
| ORIGINATING SYSTEM SOFTWARE NAME | #808006 | No | Originating application name string |
| ORIGINATING SYSTEM SOFTWARE TYPE | #808007 | No | Originating application type string |
| ORIGINATING SYSTEM SOFTWARE VERSION | #808008 | No | Originating application version string |
| FILE SET TRAILER | #808009 | Yes | CRC value or empty |

Figure 23 - Example of a File Set Trailer Field Table

13.10 File Set Index Field Table

In each Volume, the File Set Index Field Table is not mandatory. If present for a given File Set, it is recorded in the File Set Postamble, and its presence shall be indicated by the FILE SET INDEX PRESENT Field in the File Set Header Field Table. The following ten Fields are mandatory in the File Set Index Field Table:

FILE SET INDEX
 FILE SET ID
 FILE SET TIME
 FILE SET LABEL
 FILE SET INDEX FIELDS
 SOURCE NAME TYPE
 SOURCE NAME
 SOURCE OPERATING SYSTEM
 SOURCE OPERATING SYSTEM VERSION
 NUMBER OF FILES

If NUMBER OF FILES contains a value greater than 0, the following seven additional Fields are also mandatory in the File Set Index Field Table:

VOLUME SET SEQUENCE NUMBER
BUFFER ADDRESS
BUFFER OFFSET
PARENT
PATH FULLY QUALIFIED
NAME SPACE
PATH NAME

The File Set Index lists every File contained in the File Set, in the order they are recorded. Each File name(s) and location is listed in a schema of four nested levels of Iterated Field Sets (IFS). The location of each File is determined by its VOLUME SET SEQUENCE, BUFFER ADDRESS, and BUFFER OFFSET. Since Volume information, and to a lesser extent Buffer address information, may be highly redundant among the Files, the following scheme shall be used: The first VOLUME SET SEQUENCE shall be recorded (IFS 1) only once, followed by Buffer entries (IFS 2) for each Buffer of that Volume which contains the start of a File. This shall be repeated for all Volumes which contain the start of a File.

The entry for each Buffer shall begin with its BUFFER ADDRESS (address of the first Sector of the Buffer), and contains a sub-entry (IFS 3) for each File that starts in that Buffer. The sub-entry for each File shall begin with the BUFFER OFFSET (byte offset from the start of the Buffer to the File Header), and contains sub-sub-entries (IFS 4) for Name Spaces supported by the File. Each sub-sub-entry shall include the Name Space used, and the File name as expressed in terms of that Name Space. The only sub-sub-entry that is mandatory is that for the Name Space under which the File was created; this sub-sub-entry shall be listed first, those for other namespaces of the File may follow.

If a File Set Index Field Table is written, it shall conform to the following requirements:

- The File Set Index Field Table shall be recorded within Buffer(s). When a File Set Index Field Table overflows a Buffer, it shall continue as an uninterrupted Byte Sequence in the next Buffer immediately after the Buffer Header.
- The File Set Index Field Table shall be recorded beginning in the Sector immediately following the File Set Trailer within the Volume Set Data Space.
- The File Set Index Field Table shall list every File of the File Set, in the order they were recorded.
- The File Set Index Field Table listing shall indicate the logical address of each File as follows: the VOLUME SET SEQUENCE Field shall be recorded for each Volume of the File Set in which a File Header occurs. Before a different VOLUME SET SEQUENCE Field may be recorded, the BUFFER ADDRESS of each Buffer in the Volume that contains one or more File Headers shall be recorded. Before a different BUFFER ADDRESS can be recorded, the BUFFER OFFSET for each File with a File Header beginning in that Buffer shall be recorded. Before a different BUFFER OFFSET can be recorded, the NAME SPACE and PATH NAME information of the File at that offset shall be recorded.
- Fields recorded after VOLUME SET SEQUENCE and before the next instance of BUFFER ADDRESS shall be considered to be associated with the Volume specified by that VOLUME SET SEQUENCE.
- Fields recorded after BUFFER ADDRESS and before the next instance of BUFFER OFFSET shall be considered to be associated with the Buffer specified by that BUFFER ADDRESS.
- The name information associated with each file shall conform to the following requirements: The PATH FULLY QUALIFIED Field shall be recorded before the Name Space entries, and shall not be recorded again for this File. It shall be followed by one or more Name Space entries. Any Fields pertinent to the File in general, and not specific to one of the Name Spaces of the File, shall be recorded after the BUFFER OFFSET and before PATH FULLY QUALIFIED for this File. A Name Space entry shall begin with NAME SPACE, and end with PATH NAME. Any information between these two Fields shall pertain to this Name Space entry. The NAME SPACE used to create the Source entity (for example, file or Source directory) the information of which is stored in this File shall be the Name Space specified by the first Name Space entry.
- For each File, if PARENT is not set to all ZEROS, PATH FULLY QUALIFIED shall not be set to all ZEROS. If PARENT is set to all ZEROS, PATH FULLY QUALIFIED may be set to all ZEROS or not. If PATH FULLY QUALIFIED is not set to all ZEROS, each PATH NAME shall contain, in conformance with its associated

NAME SPACE, the entire path of the Source entity associated with this File. This path information shall not include SOURCE NAME. If PATH FULLY QUALIFIED is set to all ZEROS, each PATH NAME shall contain, in conformance with its associated NAME SPACE, only the terminal name (i.e. relative path) of the Source entity associated with this File, and the complete path shall be considered to be the path of the most immediately preceding File that had PARENT not set to all ZEROS, appended with terminal (i.e. not fully qualified) name of the File.

Figure 24 shows an example of a File Set Field Table.

| Field Name | FID | Mandatory | Data Description |
|---------------------------------|---------|-----------|---|
| FILE SET INDEX | #808010 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| FILE SET ID | #8072 | Yes | File Set 4-byte ID |
| FILE SET TIME | #80F403 | Yes | File Set Timestamp |
| FILE SET ABORTED | #808039 | No | Whether recording of File Set is aborted |
| FILE SET LABEL | #808005 | Yes | User-defined File Set label string |
| FILE SET INDEX FIELDS | #808034 | Yes | List of FIDs used for each File Set entry |
| SOURCE NAME TYPE | #8009 | Yes | Source naming convention type |
| SOURCE NAME | #02 | Yes | Source name string |
| SOURCE OPERATING SYSTEM | #03 | Yes | Source operating system string |
| SOURCE OPERATING SYSTEM VERSION | #04 | Yes | Source operating system version string |
| TOTAL FILE SET SIZE | #808022 | No | Total File Set size |
| NUMBER OF FILES | #808021 | Yes | Number of Files in File Set |
| VOLUME SET SEQUENCE | #80F100 | Cond. 1) | Volume position within Volume Set (1,2,..., n) |
| PARTITION NUMBER | #808012 | No 1) | Physical partition number of specified Volume |
| BUFFER ADDRESS | #08 | Cond. 2) | Sector Number of first Sector in Buffer |
| BUFFER OFFSET | #808014 | Cond. 3) | Bytes from start of Buffer to File Header |
| PARENT | #81F0FD | Cond. 3) | Whether this File is a Source directory |
| FILE IS INVALID | #80F003 | No 3) | Whether this File is valid as recorded |
| PATH FULLY QUALIFIED | #50 | Cond. 3) | Whether path is complete or relative |
| NAME SPACE | #11 | Cond. 4) | Name Space of PATH NAME immediately following |
| PATH NAME | #12 | Cond. 4) | Path name expressed in current Name Space form |
| FILE SET INDEX | #808010 | Yes | CRC value or empty |

Figure 24 - Example of a File Set Index Field Table

Notes to figure 24:

- 1) These Fields form the first Iterated Field Set IFS1. They are nested at the highest level of the four levels of IFSs of the File Set Index. Iterated Field Sets 2, 3 and 4 shall each be recorded one or more times between successive iterations of IFS 1. Iterated Field Set 1 is recorded only when the Volume number changes.
- 2) This Field forms the second Iterated Field Set IFS2. It is at the second of the four nested levels of IFSs of the File Set Index. IFS3 and IFS4 shall each be recorded one or more times before IFS2 is repeated. Additional Fields may be associated with a Buffer Address by recording them between the BUFFER ADDRESS Field for this Buffer and the next occurrence of the BUFFER OFFSET Field.
- 3) These Fields form the third Iterated Field Set IFS3. They are at the third of the four nested levels of IFSs of the File Set Index. IFS4 shall be recorded one or more times before IFS3 is repeated. Additional Fields may be

associated with a File by recording them between the *BUFFER OFFSET* and *PATH FULLY QUALIFIED Fields* for that File

- 4) *These Fields form the fourth Iterated Field Set ISF4. They are at the lowest of the four nested levels of IFSs of the File Set Index. IFS4 shall be recorded one or more times before IFS3 is repeated. Additional Fields may be associated with a specific Name Space of a File by recording them between the NAME SPACE and PATH NAME Fields for that Name Space.*

13.11 File Set Subindex Field Table

A File Set Subindex Field Table is not mandatory. If present, it shall be recorded in the same manner as a File Set Index Field Table, with the following exceptions:

- The first and last Fields shall be the FILE SET SUBINDEX Field (FID = #808033) instead of FILE SET INDEX.
- The Buffer(s) containing the File Set Subindex shall have their Header BUFFER TYPE specify that a File Set Subindex is recorded within.
- The File Set Subindex references only Files which were previously recorded, rather than an entire File Set.
- The File Set Subindex shall be recorded only at the start of a Volume Postamble.

13.12 File Header Field Table

A File Header Field Table shall be recorded within Buffer Data Space, immediately before the File Information Field Table for each Source file. The following three Fields are mandatory in the File Header Table:

FILE HEADER
FILE CHUNK SIZE
FILE TYPE

Figure 25 shows an example of a File Header Field Table.

| Field Name | FID | Required | Data Description |
|-----------------------|---------|----------|---|
| FILE HEADER | #09 | Yes | Resynchronization Pattern |
| FILE CHUNK SIZE | #0B | Yes | Bytes of this File contained in this Buffer |
| FILE TYPE | #70 | No | Type of File (e.g. Source directory, Source file) |
| REGISTERED IDENTIFIER | #808043 | No | ID for extension to Standard, scope: this File only |
| FILE HEADER | #09 | Yes | CRC value or empty |

Figure 25 - Example of a File Header Field Table

13.13 File Continuation Header Field Table

A File Continuation Header Field Table shall be recorded within a Buffer, immediately after the Buffer Header Field Table. The following two Fields are mandatory in the File Continuation Header Field Tables:

FILE CONTINUATION HEADER
FILE CHUNK SIZE

Figure 26 shows an example of a File Continuation Header Field Table.

| Field Name | FID | Required | Data Description |
|--------------------------|---------|----------|---|
| FILE CONTINUATION HEADER | #8001 | Yes | Resynchronization Pattern |
| FILE CHUNK SIZE | #0B | Yes | Bytes of this File contained in this Buffer |
| FILE IS INVALID | #80F003 | No | Whether the File is valid as recorded |
| FILE CONTINUATION HEADER | #8001 | Yes | CRC value or empty |

Figure 26 - Example of a File Continuation Header Field Table

13.14 File Information Field Table

The File Information Field Table shall be recorded within Buffer Data Space, immediately after the File Header Field Table. The following five Fields are mandatory in File Information Field Table:

FILE INFORMATION
 PARENT
 PATH FULLY QUALIFIED
 NAME SPACE
 PATH NAME

Use of the PATH FULLY QUALIFIED Field, and all Fields in the Iterated Field Set beginning with NAME SPACE and ending with PATH NAME shall meet the relevant requirements of 13.10.

Figure 27 shows an example of a File Information Field Table.

| Field Name | FID | Required | Data Description |
|----------------------|---------|----------|--|
| FILE INFORMATION | #813F | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| ATTRIBUTES | #81F2FE | No | File attributes, see annex C |
| CREATOR NAME SPACE | #81F2FC | No | Name Space under which the Source file was created |
| DATA STREAM SIZE | #81F2FB | No | Size in bytes of the data stream |
| TOTAL STREAM SIZE | #81F2FA | No | Size of all streams, in 4096-byte blocks |
| NEEDS ARCHIVE | #16 | No | Whether the Source file was modified since marked as archived |
| DELETED FLAG | #81F0FE | No | Whether the Source file was deleted |
| PARENT | #81F0FD | Yes | Whether the Source file is a Source directory |
| ACCESS TIME | #44 | No | Timestamp when Source file was last accessed |
| CREATION TIME | #64 | No | Timestamp when Source file was created |
| MODIFIED TIME | #74 | No | Timestamp when Source file was last modified |
| ARCHIVE TIME | #54 | No | Timestamp when Source file was last marked as archived |
| PATH FULLY QUALIFIED | #50 | Yes | Whether path is complete or relative |
| NAME SPACE | #11 | Yes | 1) File system naming convention |
| NAME POSITIONS | #27 | No | 1) Beginning position of each path element in the associated PATH NAME Field |
| SEPARATOR POSITIONS | #28 | No | 1) Beginning position of each path separator in the associated PATH NAME Field |
| PATH NAME | #12 | Yes | 1) Source file name, in terms of current Name Space |
| FILE INFORMATION | #813F | Yes | CRC value or empty |

Note to figure 27

1) These Fields form an Iterated Field Set repeated for each File Name Space. The first iteration shall specify the Name Space that was used to create the File. Within each iteration, NAME SPACE shall be the first Field and PATH NAME shall be the last Field, and any Fields recorded between these two shall apply to that specific iteration.

Figure 27 - Example of a File Information Field Table

13.15 File Data

File Data shall be recorded as a series of Field Tables which describe the traits and contents of a single entity of the Source. The following four Field Tables shall be included within File Data: Header Field Table, Path Field Table, Characteristics Field Table and Trailer Field Table. The Header Field Table shall be the first, and the Trailer Field Table the last Field Table recorded in the File Data. Any Streams associated with the File shall be recorded after the Characteristics Field Table in the File Data.

File Types, and the organization of Field Tables comprising the File Data are specified below for the following Source file system entities: Source volume, Source directory, Source file, and transaction set.

13.15.1 Path Field Table

The Path Field Table shall be the second Field Table recorded in the File Data for Source volume, Source file and Source directory types of Files. The following four Fields are mandatory in the Path Field Table:

PATH
 PATH FULLY QUALIFIED
 NAME SPACE
 PATH NAME

Use of the PATH FULLY QUALIFIED Field, and all Fields in the Iterated Field Set beginning with NAME SPACE and ending with PATH NAME shall meet the relevant requirements of 13.14.

Figure 28 shows an example of a Path Field Table.

| Field Name | FID | Required | Data Description |
|----------------------|-----|----------|---|
| PATH | #10 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| PATH FULLY QUALIFIED | #50 | Yes | Whether the following paths are complete or relative |
| NAME SPACE | #11 | Yes 1) | File system naming conventions |
| NAME POSITIONS | #27 | No 1) | Beginning position of each path element in the associate PATH NAME Field |
| SEPARATOR POSITIONS | #28 | No 1) | Beginning position of each path separator in the associated PATH NAME Field |
| PATH NAME | #12 | Yes 1) | File name, in terms of current Name Space |
| PATH | #10 | Yes | CRC value, or empty |

Note to figure 28

These Fields form an Iterated Field Set repeated for each File Name Space. The first iteration shall specify the Name Space that was used to create the File. Within each iteration, NAME SPACE shall be the first Field and PATH NAME shall be the last Field, and any Fields recorded between these two shall apply to that specific iteration.

Figure 28 - Example of a Path Field Table

13.15.2 Characteristics Field Table

The Characteristics Field Table shall be the third Field Table recorded in the File Data for Source volume, Source file and Source directory types of Files. The CHARACTERISTICS Field is the only mandatory Field of the Characteristics Field Table.

Figure 29 shows an example of a Characteristics Field Table.

| Field Name | FID | Required | Data Description |
|-------------------------------|---------|----------|---|
| CHARACTERISTICS | #13 | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| ACCESS TIME | #44 | No | Timestamp when the Source file was last accessed |
| ARCHIVE TIME | #54 | No | Timestamp when the Source file was last archived |
| CANT COMPRESS DATA | #81EFE6 | No | Source indicated data should not be compressed |
| COMPRESS FILE IMMEDIATE | #8116 | No | Source indicated data should be compressed |
| CREATION TIME | #64 | No | Timestamp when the Source file was created |
| SOURCE DIRECTORY | #14 | No | Whether the Source file is a Source directory or a parent |
| DO NOT COMPRESS FILE | #8115 | No | Source file owner indicated do not compress |
| EXECUTE ONLY | #813B | No | Whether the Source file marked execute only |
| HIDDEN | #15 | No | Whether the Source file marked hidden |
| INDEXED | #81EFF8 | No | Whether the Source file marked indexed |
| INHIBITIONS | #813A | No | Whether copy, delete and/or rename allowed or inhibited |
| MODIFIED TIME | #74 | No | Timestamp when the Source file was last modified |
| NEEDS ARCHIVE | #16 | No | Whether the Source file data modified since archived |
| NEEDS ARCHIVE CHARACTERISTICS | #2D | No | Whether the Source file characteristics modified since archived |
| POSIX FILE MODE | #80F203 | No | Source file Posix permissions and properties |
| POSIX FILEID | #80F210 | No | Source file Posix node number |
| POSIX FILE SYSTEM ID | #80F20F | No | Source file Posix file system ID |
| POSIX GROUP ID | #80F204 | No | Source file Posix group owner ID (Posix format) |
| POSIX NUMBER OF LINKS | #80F20D | No | Number of Source files linked to Source file's inode. |
| POSIX RDEVICE | #80F20E | No | Source file Posix raw device number |
| POSIX OWNER ID | #80F209 | No | Source file Posix owner ID |
| PURGE | #8136 | No | Whether the Source file is marked purgable |
| READ ONLY | #17 | No | Whether the Source file is marked read only |
| REMOTE DATA INHIBIT | #81EFE8 | No | Whether the Source file user indicated file is not to be migrated off the Source. |
| REMOTE DATA ACCESS | #81EFE7 | No | Whether the Source indicated file was migrated off the Source. |
| SHAREABLE | #18 | No | Whether the Source file is marked shareable |
| SYSTEM | #19 | No | Whether the Source file is marked system |
| TRANSACTIONAL | #8135 | No | Whether the Source file is marked transactional |
| CHARACTERISTICS | #13 | Yes | CRC value or empty |

Figure 29 - Example of a Characteristics Field Table

13.15.3 Source volume File Data

Source volume File Data shall consist of at least the following Field Tables, recorded in the following order:

- Source volume Header Field Table
- Path Field Table
- Characteristics Field Table
- Source volume Trailer Field Table

If additional Field Tables are to be associated with the Source volume File Data, they shall be recorded after the Characteristics Field Table and before the Source volume Trailer Field Table.

13.15.3.1 Source volume Header Field Table

The Source volume Header Field Table shall be the first Field Table recorded in the Source volume File Data. The SOURCE VOLUME HEADER Field is the only mandatory Field in the Source volume Header Table.

Figure 30 shows an example of a Source volume Header Field Table.

| Field Name | FID | Required | Data Description |
|----------------------|--------|----------|---------------------------|
| SOURCE VOLUME HEADER | #81EFC | Yes | Resynchronization Pattern |
| SOURCE VOLUME HEADER | #81EFC | Yes | CRC value or empty |

Figure 30 - Example of a Source volume Header Field Table

13.15.3.2 Source volume Trailer Field Table

The Source volume Trailer Field Table shall be the last Field Table recorded in the Source volume File Data. The SOURCE VOLUME TRAILER Field is the only mandatory Field in the Source volume Trailer Field Table.

Figure 31 shows an example of a Source volume Trailer Field Table.

| Field Name | FID | Required | Data Description |
|-----------------------|---------|----------|---------------------------|
| SOURCE VOLUME TRAILER | #81EFFB | Yes | Resynchronization Pattern |
| SOURCE VOLUME TRAILER | #81EFFB | Yes | CRC value or empty |

Figure 31 - Example of a Source volume Trailer Field Table

13.15.4 Source directory File Data

Source directory File Data shall consist of at least the following Field Tables, recorded in the following order:

- Source directory Header Field Table
- Path Field Table
- Characteristics Field Table
- Source directory Trailer Field Table

If additional Field Tables are to be associated with the Source directory File Data, they shall be recorded after the Characteristics Field Table and before the Source directory Trailer Field Table.

13.15.4.1 Source directory Header Field Table

The Source directory Header Field Table shall be the first Field Table recorded in the Source directory File Data. The Field SOURCE DIRECTORY HEADER Field is the only mandatory Field in Source directory Field Table.

Figure 32 shows an example of a Source directory Header Field Table.

| Field Name | FID | Required | Data Description |
|-------------------------|-----|----------|---|
| SOURCE DIRECTORY HEADER | #0C | Yes | Resynchronization Pattern |
| EXCLUSION OPTIONS | #29 | No | Whether streams or attributes were excluded |
| SOURCE DIRECTORY HEADER | #0C | Yes | CRC value or empty |

Figure 32 - Example of a Source directory Header Field Table

13.15.4.2 Source directory Trailer Field Table

The Source directory Trailer Field Table shall be the last Field Table recorded in the Source directory File Data. The SOURCE DIRECTORY TRAILER Field is the only mandatory Field in the Source directory Trailer Field Table.

Figure 33 shows an example of a Source directory Trailer Field Table.

| Field Name | FID | Required | Data Description |
|--------------------------|-----|----------|---------------------------|
| SOURCE DIRECTORY TRAILER | #0D | Yes | Resynchronization Pattern |
| SOURCE DIRECTORY TRAILER | #0D | Yes | CRC value or empty |

Figure 33 - Example of a Source directory Trailer Field Table

13.15.5 Source file File data

Source file File data shall consist of at least the following Field Tables, recorded in the following order:

- Source file Header Field Table
- Path Field Table
- Characteristics Field Table
- Source file Trailer Field Table

If additional Field Tables, such as Stream(s), are to be associated with the File data, they shall be recorded after the Characteristics Field Table and before the File Trailer Field Table.

13.15.5.1 Source file Header Field Table

The Source file Header Field Table shall be the first Field Table recorded in the Source file File data. The SOURCE FILE HEADER Field is the only mandatory Field in Source file Header Field Tables.

In addition an Iterated Field Set shall be recorded for each Stream type of the File the Stream of which is not recorded in clear data Stream format (see 13.15.7.3). In each Iterated Field Set, STREAM TYPE shall be the first and STREAM FORMAT the second Field recorded. If STREAM FORMAT indicates that the Stream is compressed, the Iterated Field Set shall contain the STREAM COMPRESS TYPE Field.

Figure 34 shows an example of a Source file Header Field Table.

| Field Name | FID | Required | Data Description |
|----------------------|-------|----------|--|
| SOURCE FILE HEADER | #0E | Yes | Resynchronization Pattern |
| EXCLUSION OPTIONS | #29 | No | Which, if any, Streams or attributes were excluded |
| STREAM TYPE | #2B | Cond. 1) | Type of stream (e.g. data, resource, security ...) |
| STREAM FORMAT | #2C | Cond. 1) | Physical Stream format (e.g. compressed, sparse ...) |
| STREAM COMPRESS TYPE | #8005 | Cond. 1) | Compression algorithm used for the stream |
| SOURCE FILE HEADER | #0E | Yes | CRC value or empty |

Note to figure 34.

1) These Fields form an Iterated Field Set.

Figure 34 - Example of a Source file Header Field Table

13.15.5.2 Source file Trailer Field Table

The Source file Trailer Field Table shall be the last Field Table recorded in the Source file File data. The SOURCE FILE TRAILER is the only mandatory Field in Source file Trailer Field Tables.

Figure 35 shows an example of a Source file Trailer Field Table.

| Field Name | FID | Required | Data Description |
|---------------------|-----|----------|---------------------------|
| SOURCE FILE TRAILER | #0F | Yes | Resynchronization Pattern |
| SOURCE FILE TRAILER | #0F | Yes | CRC value or empty |

Figure 35 - Example of a Source file Trailer Field Table

13.15.6 Transaction Set File data

A transaction set is a group of entities from the Source, such as Source files, that must be manipulated as a group.

Transaction Set File data shall begin with a Transaction Set Header Field Table, and end with a Transaction Set Trailer Field Table. Between these, for each of the entities of the transaction set the following group of Field Table shall be recorded in the order specified.

- Transaction Set Header

The following Field Tables, up to, and not including, the Transaction Set Trailer, represent an iterated set of Field Tables. They shall be recorded once for each entity of the Transaction Set. Such an entity shall have a unique path name within the Transaction Set; otherwise the entity is defined by the originating system.

- Header Field Table appropriate for the entity type (for example, a Source file Header)
- Path Field Table
- Characteristics Field Table
- Trailer Field Table of type matching the Header for this entity.
- Transaction Set Trailer Field Table

If additional Field Tables, such as a Streams Field Table, are to be associated with a transaction set entity, they shall be recorded after the Characteristics Field Table and before the Trailer Field Table for that entity.

If additional Field Tables are to be associated with the Transaction Set File data, they shall be recorded after the Trailer Field Table for the last entity and before the Transaction Set Trailer Field Table.

13.15.6.1 Transaction Set Header Field Table

The Transaction Set Header Field Table shall be the first Field Table recorded in the Transaction Set File data. The TRANSACTION SET HEADER Field is the only mandatory Field in Transaction Set header Field Tables.

Figure 36 shows an example of a Transaction Set header Field Table.

| Field Name | FID | Required | Data Description |
|------------------------|---------|----------|--|
| TRANSACTION SET HEADER | #81EFF3 | Yes | Resynchronization Pattern |
| TRANSACTION SET TYPE | #81EFEE | No | Transaction Set type, as defined by Source |
| TRANSACTION SET NAME | #81EFEF | No | Transaction Set name, as defined by Source |
| TRANSACTION SET HEADER | #81EFF3 | Yes | CRC value or empty |

Figure 36 - Example of Transaction Set Header Field Table

13.15.6.2 Transaction Set Trailer Field Table

The Transaction Set Trailer Field Table shall be the last Field Table recorded in the Transaction Set File data. The TRANSACTION SET TRAILER Field is the only mandatory Field in the Transaction Set Trailer Field Table.

Figure 37 shows an example of a Transaction Set Trailer Field Table.

| Field Name | FID | Required | Data Description |
|-------------------------|---------|----------|---------------------------|
| TRANSACTION SET TRAILER | #81EFF2 | Yes | Resynchronization Pattern |
| TRANSACTION SET TRAILER | #81EFF2 | Yes | CRC value or empty |

Figure 37 - Example of Transaction Set Trailer Field Table

13.15.7 Streams

A Stream is a set of logically related bytes in a File. Streams shall be recorded after the Characteristics Field Table, and before the associated trailer Field Table for their File. A Stream consists of a Stream Header Field Table, followed by the Stream data, followed by a Stream Trailer Field Table.

13.15.7.1 Stream Header Field Table

The following four Fields are mandatory in the Stream Header Field Table:

STREAM HEADER
 STREAM TYPE
 STREAM FORMAT
 STREAM SIZE

In addition the Stream Header Field Table shall meet to the following requirements:

- If STREAM FORMAT indicates that the Stream is compressed, STREAM COMPRESSED TYPE shall be recorded.
- If STREAM FORMAT indicates that the Stream is compressed, sparse or delta block, STREAM EXPANDED SIZE shall be recorded.
- If STREAM FORMAT indicates that the Stream is sparse or delta block format, BLOCK SIZE and BLOCK MAP shall be recorded.
- If STREAM FORMAT indicates that the Stream is delta block or delta extent format, DELTA BASE TIME shall be recorded.
- If STREAM FORMAT indicates that the Stream is delta extent format, DELTA EXTENT OFFSET and DELTA EXTENT OLD SIZE shall be recorded.
- If STREAM TYPE indicates that the Stream contains extended attributes, EA KEY shall be recorded.
- The STREAM SIZE value shall be the number of bytes of the Stream as recorded on the Volume.
- If multiple Streams of the same STREAM TYPE are recorded in a File, each such Stream Header shall contain STREAM TYPE SEQUENCE. The STREAM TYPE SEQUENCES for a File Streams sharing the same STREAM TYPE shall form an ascending ordinal sequence (1, 2, 3, ..., *n*) matching the order in which those Streams are recorded.

Figure 38 shows an example of a Stream Header Field Table.

| Field Name | FID | Required | Data Description |
|-----------------------|-------|----------|---|
| STREAM HEADER | #1D | Yes | Resynchronization Pattern |
| OFFSET TO END | #01 | No | Bytes from start of next Field to start of last Field |
| STREAM TYPE | #2B | Yes | Type of Stream (e.g. data, resource, security ...) |
| STREAM FORMAT | #2C | Yes | Physical Stream format (e.g. compressed, sparse ...) |
| STREAM TYPE SEQUENCE | #61 | Cond. | Sequence number of the Stream for a given Stream type |
| STREAM SIZE | #20 | Yes | Bytes in Stream as recorded |
| STREAM COMPRESS TYPE | #8005 | Cond. | Compression algorithm used for the Stream |
| STREAM EXPANDED SIZE | #8006 | Cond. | Logical bytes in Stream (e.g. uncompressed) |
| BLOCK SIZE | #24 | Cond. | Block size of block map, in bytes |
| BLOCK MAP | #25 | Cond. | Which blocks are used in Stream |
| EA KEY | #1B | Cond. | The extended attribute key value |
| DELTA BASE TIME | #8044 | Cond. | Timestamp of base of Source delta |
| DELTA EXTENT OFFSET | #800A | Cond. | Stream byte offset to start of delta extent |
| DELTA EXTENT OLD SIZE | #800B | Cond. | Size of area replaced by delta extent |
| STREAM HEADER | #1D | Yes | CRC value or empty |

Figure 38 - Example of a Stream Header Field Table

13.15.7.2 Stream Trailer Field Table

The STREAM TRAILER Field is the only mandatory Field in the Stream Trailer Field Table.

Figure 39 shows an example of a Stream Trailer Field Table.

| Field Name | FID | Required | Data Description |
|-------------------|-----|----------|--|
| STREAM TRAILER | #1E | Yes | Resynchronization Pattern |
| STREAM IS INVALID | #21 | No | Whether the stream is deemed invalid by the originating system |
| STREAM CRC | #22 | No | CRC of the stream as recorded |
| STREAM TRAILER | #1E | Yes | CRC value or empty |

Figure 39 - Example of Stream Trailer Field Table

13.15.7.3 Stream format

Stream format is the physical representation in which a Stream is recorded. Stream format is specified by the STREAM FORMAT Field in the Stream Header Field Table. This ECMA Standard specifies the following five Stream formats:

- clear data
- sparse
- compressed
- delta block
- delta extent

If multiple Streams with the same STREAM TYPE are recorded for a File, these Streams must all be recorded using the same Stream format.

Sparse, delta block and delta extent Stream formats shall record a subset of the data of the corresponding stream of the Source, as specified below. Delta block and delta extent Stream formats represent changes that

have occurred to a Source Stream specified by DELTA BASE TIME. Changes represented are relative to the state of the Source Stream at the DELTA BASE TIME, so only blocks or extents that have been changed are recorded.

NOTE

Delta block and delta extent Stream may not be fully usable by the receiving system unless an image of the Source Stream at the DELTA BASE TIME is available.

13.15.7.4 Block-mapped Stream formats

Sparse and delta block Stream formats use a block mapping scheme to represent the Stream. A block-mapped Stream shall specify its EXPANDED STREAM SIZE, BLOCK SIZE, and BLOCK MAP in its Stream Header. The term expanded stream image shall be used to represent a sequence of bytes of the size indicated by EXPANDED STREAM SIZE, containing the data explicitly or implicitly represented by the Source Stream (for example, the uncompressed image of a compressed Source Stream). This expanded Stream image is handled as a series of equally sized, contiguous blocks of the size indicated by BLOCK SIZE. The BLOCK MAP shall be a bit map indicating which blocks of the expanded Stream image are recorded as this Stream and which blocks are not recorded. The bits in the bit map shall have an ordered one-to-one correspondence to the blocks of the expanded Stream image. The bit map shall have the first bit recorded reflect the presence of the first block of the expanded Stream image, second bit describe the second block, and so forth. A bit shall be set to ONE if its corresponding block of the expanded Stream image is recorded. A bit shall be set to ZERO if its corresponding block of the expanded Stream image is not recorded.

13.15.7.4.1 Clear data Stream format

The clear data Stream format shall represent the Stream data as a sequence of bytes, without data compression performed by the Source or the originating system application, such that the number of bytes recorded, and their contents, are the same as is available to a user of the Source file.

13.15.7.4.2 Compressed Stream format

The compressed Stream format shall represent Stream data as a sequence of bytes submitted to compression performed or maintained by the Source or the originating system application. The type of data compression used shall be indicated by the STREAM COMPRESS TYPE Field in the Stream Header.

13.15.7.4.3 Sparse Stream format

Stream data using the sparse Stream format shall be recorded as a block-mapped Stream format. Blocks of the expanded stream image that are not recorded should be considered to contain only NULL bytes.

13.15.7.4.4 Delta block Stream format

Stream data using the delta block Stream format shall be recorded as a block-mapped Stream format. Blocks of the expanded stream image that are not recorded should be considered unchanged from their pre-delta value (i.e. unchanged from the state of the Source file Stream as it existed at the Source at the DELTA BASE TIME recorded in the Stream Header).

13.15.7.4.5 Delta extent Stream format

An extent is defined as a contiguous set of bytes within a Stream. The original stream is defined as the Source stream as it existed at the Source at the DELTA BASE TIME specified in the Stream Header. Per agreement between the originating and receiving systems, the delta extent Stream replaces a specified extent within the original stream. This latter extent shall be referred to as the replaced extent. The expanded stream image represented by the delta stream shall be defined as the concatenation of the part of the original stream before the replaced extent, the delta extent, and the part of the original stream after the replaced extent.

DELTA EXTENT OFFSET shall indicate the byte offset from the start of the original Stream to the start of the replaced extent. DELTA EXTENT SIZE shall indicate the size of the replaced extent. STREAM SIZE shall indicate the size of the delta extent, so that the delta extent may be of different size than the replaced extent. No combination of delta extent and/or delta block Streams shall be recorded such that the resulting Stream is ambiguous. If a Stream has a greater DELTA BASE TIME than a previous Stream sequence, the deltas are based on the Source Stream with the previous sequences changes applied.

13.15.7.5 Stream sequence

STREAM SEQUENCE shall be recorded in the Stream Header of all Streams of a File that are of the same STREAM TYPE as another Stream of the same File. Streams of the same type within a File shall be sequenced ordinally (1, 2, 3, ..., n) in the order they are recorded on the media. STREAM SEQUENCE shall increment independently for different STREAM TYPE values within a File.

13.16 Levels of partition interchange

Two levels of partition interchange are specified.

13.16.1 Level 1 of partition interchange

At level 1, the following special restrictions shall apply:

- A File Set shall not be interleaved.
- For media which support file marks, the Volume Header file mark usage value shall be 3, indicating that file marks shall be written before all File Set Trailers, and before all File Set Headers recorded after the first Buffer of a Volume.
- Character use shall be limited to the CS4 character set (see 6.8).
- A File Set shall not span physical partitions or media.
- The Buffer Size shall not be greater than 65 536 bytes.
- The Buffer Size shall be a fixed size per File Set, and that size shall match the value recorded in the File Set Header.
- Numeric data values shall be less than 2^{32} .
- Volume Headers and Trailers, and File Set (Continuation) Headers and Trailers shall not be larger than the Sector Size specified in the Volume Header.
- The Volume Header shall be recorded in the first user-accessible physical sector of the partition.

13.16.2 Level 2 of partition interchange

At level 2, no restrictions shall apply.

14 Requirements for systems

14.1 Requirements for the description of systems

This ECMA Standard specifies that certain information shall be communicated between a primary file system and/or its associated application, and a medium.

14.2 Requirements for an originating system

14.2.1 General

The originating system shall be capable of recording primary file system information as a Volume Set according to the normative specifications of this ECMA Standard in conformance with one of the partition interchange levels specified in 13.16.

14.2.2 Mandatory access by user

The originating system shall obtain from the requesting application the information that constitutes the Source files to be recorded on the Volume Set.

14.2.3 Optional access by user

The originating system shall permit the requesting application to supply other information to be recorded in any of the following Field Table fields, and shall record such information as supplied by the requesting application. The originating system shall supply the information for such a Field if the requesting application does not supply it. The applicable Fields are listed for the following Field Tables:

Volume Header Field Table

- VOLUME SET LABEL

File Set Header Field Table

- FILE SET LABEL
- SOURCE NAME

14.2.4 Volume characteristics

Whenever an originating system records a File Set on a Volume, the characteristics of the following Fields of the Volume Header shall be preserved, independent of which originating system recorded the Volume Header:

FILE MARK USAGE, and
VOLUME INDEX PRESENT

The following Fields, if recorded, shall have consistent data values within the Headers, Trailers and (Sub)Indices of a Volume Set:

VOLUME SET LABEL
VOLUME SET TIME.

14.2.5 File Set characteristics

The following Fields, if recorded, shall have consistent data values within the (Continuation) Headers, Buffers, Trailer and (Sub)Indices of a File Set:

FILE SET ID
FILE SET TIME
FILE SET LABEL
SOURCE NAME
SOURCE OPERATING SYSTEM
SOURCE OPERATING SYSTEM VERSION

If one or more Registered Identifiers are recorded in a File Set Header, they shall be re-recorded in all File Set Continuation Headers of that File Set.

14.2.6 Recording of Bit Data

Fields the Data Description of which (annex C) specifies that their data part shall be recorded as Bit Data, shall be recorded according to B.3 in annex B.

14.2.7 Cyclic Redundancy Check (CRC)

This ECMA Standard specifies CRC use for

- Field Tables
- Buffers
- Streams

Recording of CRC values is optional. CRC shall be calculated according to clause 9. The size of the data part of a Field containing CRC bits shall be 4 bytes.

If calculated for, and recorded in, a Field Table, the CRC bits shall be recorded as the data part of the last Field of the Field Table. The data used as input for the CRC shall be the entire Field Table except the last Field.

If calculated for, and recorded in, a Buffer, the CRC bits shall be recorded as the data part of the BUFFER CRC Field of the Buffer Header. The data used as input for the CRC shall be the entire Buffer except the Buffer Header.

If calculated for, and recorded in the Trailer of, a Stream, the CRC bits shall be recorded as the data part of the STREAM CRC Field of the Stream Trailer Field Table. The data used as input for the CRC shall be the entire Stream as recorded on the Volume, independent of any source stream or Stream Format interpretation. The Stream is all bytes within Buffer Data Space following the Stream Header and preceding the Stream Trailer.

14.3 Requirements for a receiving system

14.3.1 General

The receiving system shall be capable of reading the Files and associated information from a Volume Set that has been recorded in conformance with this ECMA Standard in one of the partition interchange levels specified in 13.16. The receiving system shall make available to the requesting application or file system the information that constitutes the recorded Files of the Volume Set. The receiving system shall make available to the requesting application or file system the information that is descriptive of the Source files and was supplied by the Source to the originating system and recorded on the Volume Set.

The originating system may record Source file associated information that a conforming receiving system is unable to interpret. When processing such information, the receiving system shall nevertheless make this information available to the requesting application and/or file system.

14.3.2 Levels of conformance

This ECMA Standard specifies two levels of conformance for receiving systems. These are known as Level 1 and Level 2. A statement of conformance of a receiving system shall specify the level to which the receiving system conforms.

14.3.2.1 Level 1 conformance of a receiving system

A receiving system shall conform to Level 1 if it meets all the requirements of 14.3.2.1 and 14.3.2.2 when reading a Volume from a medium partition that conforms to the requirements of 13.16.1.

14.3.2.2 Level 2 conformance of a receiving system

A receiving system shall conform to Level 2 if it meets all the requirements of 14.3.2.1 and 14.3.2.2 when reading a Volume from a medium partition that conforms to the requirements of 13.16.2.

Annex A
(normative)

Structure of the Field Identifier

A.1 Overview

Each Field Identifier (FID) shall consist of a sequence of one to four bytes, uniquely identified by the setting of the bits of these bytes. Various bits in these bytes may have a specific meaning.

A.1.1 Bits specifying an operating system

Bits specifying an operating system shall be a set of six adjacent bits which denote that certain FIDs pertain to a specific operating system. These bits are used in certain 2-byte and 3-byte FIDs as indicated below. An operating system specified by these bits shall be registered in accordance with ISO/IEC 13800.

A.1.2 Bits specifying a developer

Bits specifying a developer shall be a set of thirteen adjacent bits which denote that certain FIDs pertain to a specific developer. These bits are used in certain 3-byte and 4-byte FIDs as indicated below. A developer specified by these bits shall be registered in accordance with ISO/IEC 13800.

A.2 General Structure of 1-byte FIDs

1-byte FIDs shall be identified by their high-order bit b7 set to ZERO. The interpretation of bits b5 to b0 depends on the setting of b6.

If b6 is set to ZERO, the Data part of the Field shall be of variable length, and bits b5 to b0 shall be in the range #00 to #3F, thus identifying 63 such FIDs. In this case, the length of the Data part of the Field shall be specified by the Data Length part of the Field.

If b6 is set to ONE, the Data part of the Field shall be of fixed length specified by bits b2 to b0, and the Data Size part of the Field shall be omitted. Bits b2 to b0 shall specify a number N , such that 2^N expresses the length of the data. Bits b5, b4 and b3 allow eight different FIDs to be specified for each of these fixed data length.

| 1-byte FIDs | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Variable Data Length | 0 | 0 | x | x | x | x | x | x |
| Fixed Data Length | 0 | 1 | x' | x' | x' | <i>n</i> | <i>n</i> | <i>n</i> |

xxxxxx: Bits of the FID, identifying 63 different FIDs
 x'x'x': Bits of the FD identifying 8 different FIDs for each data length
nnn: Length of the Data part of Field equals 2^N

Figure A.1 - Structure of 1-byte FIDs

A.2.1 Special case of 1-byte FIDs: NULL Field/ NULL FID

A 1-byte FID with all bits set to ZERO is designated the NULL FID, it identifies the NULL Field, which has no Data Length part and no Data part.

A.3 General Structure of 2-byte FIDs

2-byte FIDs shall be identified by the bits b7 and b6 of the first byte set to ONE and ZERO, respectively, and bit b7 of the second byte set to ZERO.

If the first byte is set to #80, the FID shall not specify an operating system. If the first byte is in the range #81 to #BF, the FID shall specify the operating system identified by bits b5 to b0. In either case, the second byte shall be interpreted as specified for 1-byte FIDs.

| | First Byte | Second Byte |
|----------------------|--------------------------------|--------------------------------|
| 2-byte FIDs | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 |
| Variable Data Length | 1 0 s s s s s s | 0 0 x x x x x x |
| Fixed Data Length | 1 0 s s s s s s | 0 1 x' x' x' n n n |

ssssss: set to all ZEROs, or to an operating system identification number in the range #01 to #3F
 xxxxxx: Bits of the FID identifying 6 different FIDs
 x'x'x': Bits of the FID identifying 8 different FIDs for each data length for each operating system
 nnn: Length of the Data part of Field equals 2^N

Figure A.2 - Structure of 2-byte FIDs

A.4 General Structure of 3-byte FIDs

3-byte FIDs shall be identified by either:

Case A: Bits b7 and b6 of the first byte set to ONE and ZERO, respectively, and bit b7 of the second byte set to ONE; or

Case B: Bits b7 and b6 of the first byte set to ONE, and bit b7 of the third byte set to ZERO.

In Case A, bits b5 to b0 of the first byte shall either be bits specifying an operating system, or set to all ZEROs. In Case B, bits b5 to b0 of the first byte, and all bits of the second byte shall specify a developer.

Case A: In this case, if bits b6, b5 and b4 of the second byte are not all set to ONE, the Data part of the Field shall be of variable length. This length shall be specified by the Data Length part of the Field. Bits b6 to b0 of the second byte, and all bits of the third byte are simply part of the FID, and have no specific meaning.

If bits b6, b5 and b4 of the second byte are all set to ONE, the Data part of the Field shall be of fixed length. Bits b2 to b0 of the second byte shall define a number N such that the length of the Data part of the Field is 2^N . Bit b3 of the second byte, together with all the bits of the third byte, shall specify up to 512 different FIDs for each data length.

| | First Byte | Second Byte | Third Byte |
|----------------------|--------------------------------|--------------------------------|--------------------------------|
| 3-byte FIDs | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 |
| Variable Data Length | 1 0 s s s s s s | 1 x x x x x x x | x x x x x x x |
| Fixed Data Length | 1 0 s s s s s s | 1 1 1 1 x' n n n | x' x' x' x' x' x' x' |

ssssss: set to all ZEROs, or to an operating system identification number in the range #01 to #3F
 xxxxxxxxxxxxxxxx: Bits of the FID identifying 32 768 different FIDs
 x'x'x'x'x'x'x'x': Bits of the FID identifying 512 different FIDs for each data length for each operating system; the high-order bit of these is bit b3 of the second byte.
 nnn: Length of the Data part of Field equals 2^N

Figure A.3 - Structure of 3-byte FIDs: Case A

Case B: In this case bits b5 to b0 of the first byte and all bits of the second byte shall either specify a developer or be set to all ZEROs. The third byte shall be interpreted as specified for 1-byte FIDs.

| | First Byte | Second Byte | Third Byte |
|----------------------|--------------------------------|--------------------------------|--------------------------------|
| 3-byte FIDs | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 |
| Variable Data Length | 1 1 0 d d d d d | d d d d d d d d | 0 0 x x x x x x |
| Fixed Data Length | 1 1 0 d d d d d | d d d d d d d d | 0 1 x' x' x' n n n |

dddddddddddd: Developer identification number in the range #0000 to #1FFF
 xxxxxx: Bits of the FID identifying 64 different FIDs
 x'x'x': Bits of the FID identifying 8 different FIDs for each data length for each developer
 nnn: Length of the Data part of Field equals 2^N

Figure A.4 - Structure of 3-byte FIDs - Case B

A.5 General Structure of 4-byte FIDs

4-byte FIDs shall be identified by bits b7 and b6 of the first byte set to ONE and bit b7 of the third byte set to ONE. The first and second bytes shall be interpreted as specified for Case B of the 3-byte FIDs. The third and the fourth bytes shall be interpreted as specified for Case A of the 3-byte FIDs.

| First Byte | Second Byte | Third Byte | Fourth Byte |
|--------------------------------|--------------------------------|--------------------------------|---------------------------------------|
| b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 |
| Variable Data Length: | d d d d d d d d | 1 x x x x x x x | x x x x x x x x |
| Fixed Data Length: | 1 1 0 d d d d d | d d d d d d d d | 1 1 1 1 x' n n n x' x' x' x' x' x' x' |

dddddddddddd: Developer identification number in the range #0000 to #1FFF
 xxxxxxxxxxxxxxxx: Bits of the FID identifying 32 768 FIDs
 x'x'x'x'x'x'x'x': Bits of the FID identifying 512 different FIDs for each length for each developer; the high-order bit of these is from bit b3 of the third byte.
 nnn: Length of the Data part of Field equals 2^N

Figure A.5 - Structure of 4-byte FIDs

A.6 FIDs not reserved by this ECMA Standard

FIDs with bits b7, b6 and b5 of the first byte set to ONE are not defined by this ECMA Standard, and are available for registration via ISO/IEC 13800. However, the use of these FIDs shall follow the conventions of this annex as follows.

- If bit b7 of the third byte of such a FID is set to ZERO, the FID shall be a 3-byte FID, and the interpretation of the third byte shall be the same as for Case B for 3-byte FIDs as specified above.
- If bit b7 of the third byte of such a FID is set to ONE, the FID shall be a 4-byte FID, and the interpretation of the third and fourth bytes shall be the same as for the 4-byte FIDs as specified above.

NOTE

Bits b4 to b0 of the first byte and all bits of the second byte of FIDs not defined by this ECMA Standard are intended to be associated arbitrarily to different Registered Identifiers of ISO/IEC 13800.

A.7 Allocation of FIDs by range

| FID range | FID type |
|----------------------|--|
| #00-3F | One byte, variable data length, generic |
| #40-7F | One byte, fixed data length, generic |
| | |
| #8000-803F | Two bytes, variable data length, generic |
| #8040-807F | Two bytes, fixed data length, generic |
| #8100-813F | Two bytes, variable data length, 1st operating system |
| #8140-817F | Two bytes, fixed data length, 1st operating system |
| #8100-813F | Two bytes, variable data length, 2nd operating system |
| #8140-817F | Two bytes, fixed data length, 2nd operating system |
| ⋮ | ⋮ |
| #BF00-BF3F | Two bytes, variable data length, 63rd operating system |
| #BF40-BF7F | Two bytes, fixed data length, 63rd operating system |
| | |
| #808000-#80EFFF | Three bytes, variable data length, generic |
| #80F000-#80FFFF | Three bytes, fixed data length, generic |
| #818000-#81EFFF | Three bytes, variable data length, 1st operating system |
| #81F000-#81FFFF | Three bytes, fixed data length, 1st operating system |
| #828000-#82EFFF | Three bytes, variable data length, 2nd operating system |
| #82F000-#82FFFF | Three bytes, fixed data length, 2nd operating system |
| ⋮ | ⋮ |
| #BF8000-#BFEFFF | Three bytes, variable data length, 63rd operating system |
| #BFF000-#BFFFFFFF | Three bytes, fixed data length, 63rd operating system |
| #C00000-#C0003F | Three bytes, variable data length, 0-th developer |
| #C00040-#C0007F | Three bytes, fixed data length, 0-th developer |
| #C00100-#C0013F | Three bytes, variable data length, 1st developer |
| #C00140-#C0017F | Three bytes, fixed data length, 1st developer |
| ⋮ | ⋮ |
| #DFFF00-#DFFF3F | Three bytes, variable data length, 8191st developer |
| #DFFF40-#DFFF7F | Three bytes, fixed data length, 8191st developer |
| #E00000-#FFFF7F | Three bytes, not defined by this ECMA Standard, available for registration according to ISO/IEC 13800 |
| | |
| #C0008000-#C000EFFF | Four bytes, variable data length, 0-th developer |
| #C000F000-#C000FFFF | Four bytes, fixed data length, 0-th developer |
| #C0018000-#C001EFFF | Four bytes, variable data length, 1st developer |
| #C001F000-#C001FFFF | Four bytes, fixed data length, 1st developer |
| ⋮ | ⋮ |
| #DFFF8000-#DFFF8FFF | Four bytes, variable data length, 8191st developer |
| #DFFF8F00-#DFFFFFFF | Four bytes, fixed data length, 8191st developer |
| #E0018000-#FFFFFFF7F | Four bytes, not defined by this ECMA Standard, available for registration according to ISO/IEC 13800 |

Figure A.6 - Summary of the FID ranges

Annex B
(normative)

Specification of the length of data in Fields

All Fields with variable data length (i.e. length not specified as fixed by the FID) shall have a Data Length part. The Data Length part shall be presented in one of three different formats.

B.1 Direct length format

The Direct length format shall be identified by bit b7 of the first byte of the Data Length part of the Field set to ZERO. The Direct length format shall occupy one byte. Bits b6 to b0 shall represent in binary notation the data length from 0 to 127.

| | | | | | | | |
|----|----------|----------|----------|----------|----------|----------|----------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 0 | <i>n</i> | <i>n</i> | <i>n</i> | <i>n</i> | <i>n</i> | <i>n</i> | <i>n</i> |

nnnnnnn: Length of the Data part of Field, from 0 to 127

Figure B.1 - Structure of a Direct 1-byte Data Length part of a Field

B.2 Indirect length format

The Indirect length format shall be identified by bit b7 of the first byte of the Data Length part of the Field set to ONE, and bits b6 to b2 set to ZERO. Bits b1 and b0 shall specify a number N , such that the 2^N bytes immediately following the first byte of the Data Length part specify the length in bytes of the data part. These bytes specifying the data length shall be recorded as a number (with low-order byte first). The Indirect length format shall occupy between two and nine bytes. Lengths of 0 to $18 \cdot 10^{18}$ bytes can be represented by this format.

| | | | | | | | |
|------------|----|----|----|----|----|----------|----------|
| First Byte | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 0 | 0 | 0 | 0 | <i>n</i> | <i>n</i> |

nn: 2^N = number of bytes specifying the data length, to follow immediately

Figure B.2 - Structure of the first byte of the Indirect Data length part of a Field

| First Byte (indirect:2 bytes follow) | Second Byte | Third Byte |
|--------------------------------------|-------------------------|-------------------------|
| b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 | b7 b6 b5 b4 b3 b2 b1 b0 |
| 1 0 0 0 0 0 0 1 | 0 1 1 0 0 0 0 0 | 0 0 0 0 1 0 1 0 |

Figure B.3 - Example of Indirect length: bits b1 and b0 of first byte set to 01, specifying a length recorded in the following $2^1 = 2$ bytes. These bytes specify a data length #600A = 24586 bytes.

B.3 Bit Data length format

The Bit Data length format shall be identified by bits b7 and b6 of the first byte of the Data Length part of the Field set to ONE. This data length consists of one byte. The data is expressed by 1 to 6 bits from b0 to b5. In each FID definition (see annex C), the relevant bit or bits are specified. Bits not specified shall be ignored. Because the Field data is stored in the Data Length part, no Data Part is recorded.

| | | | | | | | |
|----|----|----------|----------|----------|----------|----------|----------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 1 | <i>d</i> | <i>d</i> | <i>d</i> | <i>d</i> | <i>d</i> | <i>d</i> |

dddddd Data bits, specifying a range of data values from 0 to 63

Figure B.4 - Structure of the Data Length part of a Field in Bit Data format

Annex C

(normative)

Field specification

This annex lists alphabetically all Fields defined in this ECMA Standard. The Fields are presented as follows:

- The name of the Field
- The FID: presented and recorded high-order byte first (i.e. big endian).
- The Data length: The length of the Data Length part of the Field. If the Field has a fixed Data Length, the word "Fixed" is shown, followed by the length in bytes. If the Field has a variable data size, the word "Variable" is shown. If the Data Length is shown as Bit Data, the data shall be recorded according to annex B, B.3.
- Data description: A description of the data and the setting of bits, where applicable.
- Unless otherwise specified, the absence of a Field which is not required shall be interpreted as if the Field is present and set to all ZEROs.
- All reserved bits/bytes shall be set to ZERO/all ZEROs. This includes bytes B12 to B15 of Timestamps.

ACCESS TIME

FID: #44

Data length: Fixed, 16 bytes

Data description: The Timestamp from the Source when the file was last accessed.

ARCHIVE TIME

FID: #54

Data length: Fixed, 16 bytes

Data description: The Timestamp from the Source when the file was last marked as archived.

ATTRIBUTES

FID: #81F2FE

Data length: Variable

Data description: The following attribute(s) shall be associated with a Source entity (for example, a Source file) if, and only if, the corresponding bit(s) is set to ONE. Bits denoting attributes not associated with a Source entity shall be set to ZERO.

| Bit | Meaning |
|------------|--|
| b0 | Read only |
| b1 | Hidden from normal Source directory access |
| b2 | Intended for use only by the operating system (system) |
| b3 | Execute only, shall not be read or written to |
| b4 | Source directory |
| b5 | One or more of the file Streams have been modified since the archive bit (this bit) was last cleared (archive) |
| b6 | Reserved, no meaning |
| b7 | May be accessed simultaneously by multiple users (shareable) |
| b8 | Reserved, no meaning |
| b9 | Reserved, no meaning |
| b10 | Reserved, no meaning |
| b11 | Do not suballocate file block(s) with other files |
| b12 | File is transactional |
| b13 | Reserved, no meaning |
| b14 | Reserved, no meaning |
| b15 | Reserved, no meaning |
| b16 | File to be purged immediately after deletion |
| b17 | File shall not be renamed |
| b18 | File shall not be deleted |
| b19 | File shall not be copied |
| b20 | Reserved, no meaning |
| b21 | Reserved, no meaning |
| b22 | File or Source directory was migrated to secondary storage |
| b23 | File or Source directory shall not be migrated |
| b24 | Reserved, no meaning |
| b25 | File streams shall be compressed onto target file system |
| b26 | File streams were compressed |
| b27 | File owner specified file Streams should not be compressed |
| b28 | Reserved, no meaning |
| b29 | File deemed not effectively compressible by file system |
| b30 | File information other than Stream data has been modified since bit b5 was last cleared |

AUTHENTICATION

FID: #8002

Data length: Variable

Data description: This string shall contain a password for the Volume or File Set created by, or for, the originating system. A receiving system accessing the Volume or File Set shall validate the user-supplied password against the Volume or File Set Authentication value before access to the Volume or File Set is granted to the user.

The string shall be XORed on a 4-byte basis with the Byte Stream #53694466 (characters "SiDf").

BLANK SPACE

FID: #808019
Data length: Variable
Data description: Resynchronization data, CRC or empty.

BLOCK MAP

FID: #25
Data length: Variable
Data description: Bit map for the interpretation of a Data Stream.

BLOCK SIZE

FID: #24
Data length: Variable
Data description: The size of the blocks defined by the BLOCK MAP of a Data Stream.

BUFFER ADDRESS

FID: #08
Data length: Variable
Data description: The distance in Sectors from the File Set (Continuation) Header which is associated with, and in, the same Volume as the Buffer, to the start of the Buffer. A File Set (continuation) Header and a Buffer are associated if they have the same FILE SET ID.

BUFFER CRC

FID: #8008
Data length: Variable
Data description: The CRC of the Buffer excluding the Buffer Header, but including Blank Space at end of Buffer counted in UNUSED IN THIS BUFFER.

BUFFER HEADER

FID: #05
Data length: Variable
Data description: Resynchronization data, CRC or empty.

BUFFER OFFSET

FID: #808014
Data length: Variable
Data description: The offset in bytes to a location within a Buffer.

BUFFER SEQUENCE

FID: #07

Data length: Variable

Data description: The ordinal sequence number of a Buffer within its File Set or other usage set (1, 2, 3, ..., n).

BUFFER SIZE

FID: #06

Data length: Variable

Data description: In the File Set Header: This Field shall contain the maximum Buffer size in bytes for the File Set. In the Buffer Header: This Field shall contain the size in bytes of this Buffer. The Buffer size shall be an integer multiple of the Sector size.

BUFFER TYPE

FID: #60

Data length: Fixed, 1 byte

Data description: This Field shall specify which of the 6 types of data the Buffer contains.

| Value | Data type |
|--------------|--------------------|
| 0 | Implementation Use |
| 1 | File |
| 2 | File Set Index |
| 3 | File Set Subindex |
| 4 | Volume Index |
| 5 | Volume Subindex |

No other value shall be recorded.

CANT COMPRESS DATA

FID: #81EFE6

Data length: Bit Data

Data description: This Field shall specify whether the Source has indicated that it could not effectively compress the data associated with this entity.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | No indication on effective compression |
| Bit b0 set to ONE: | Effective compression impossible |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

CHARACTERISTICS

FID: #13

Data length: Variable

Data description: Resynchronization data, CRC or empty.

CHAR SPEC

FID: #808040

Data length: Variable

Data description: The first byte shall indicate the character specification type. Additional bytes, as needed, shall indicate character set information.

CLOSE TIME

FID: #80F402

Data length: Fixed, 16 bytes

Data description: The Timestamp from the originating system when the Volume Trailer was recorded.

COMPRESS FILE IMMEDIATE

FID: #8116

Data length: Bit Data

Data description: This Field shall specify whether the user has requested that the Source information associated with the File be compressed when transferred by the receiving system to a primary file system.

| Setting | Meaning |
|---------------------|------------------------------------|
| Bit b0 set to ZERO: | No immediate compression requested |
| Bit b0 set to ONE: | Immediate compression requested |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

CREATION TIME

FID: #64

Data length: Fixed, 16 bytes

Data description: The Timestamp when the Source entity was created.

CREATOR NAME SPACE

FID: #81F2FC

Data length: Fixed, 4 bytes

Data description: Name Space used to create the Source entity.

DELETED FLAG

FID: #81F0FE

Data length: Fixed, 1 byte

Data description: This Field shall specify whether the Source entity was deleted.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Source entity not deleted |
| Bit b0 set to ONE: | Source entity deleted |
| Bits b1 to b7 | shall be set to ZERO, no meaning |

DELTA BASE TIME

FID: #8044

Data length: Fixed, 16 bytes

Data description: Timestamp of base of Source delta.

DELTA EXTENT OFFSET

FID: #800A

Data length: Variable

Data description: Within the Source stream at DELTA BASE TIME, the byte offset from the start of the stream to the start of delta extent

DELTA EXTENT OLD SIZE

FID: #800B

Data length: Variable

Data description: The number of bytes of the pre-delta Source stream to be replaced by the delta extent.

DEVICE INFO

FID: #808032

Data length: Variable

Data description: Implementation use information.

SOURCE DIRECTORY

FID: #14

Data length: Bit Data

Data description: This Field shall specify whether the Source entity is a Source directory or a parent.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | Source entity is no Source directory or parent |
| Bit b0 set to ONE: | Source entity is Source directory or parent |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

SOURCE DIRECTORY HEADER

FID: #0C
Data length: Variable
Data description: Resynchronization data, CRC or empty.

SOURCE DIRECTORY TRAILER

FID: #0D
Data length: Variable
Data description: Resynchronization data, CRC or empty.

DO NOT COMPRESS FILE

FID: #8115
Data length: Bit Data
Data description: This Field shall specify whether the user has requested that the Source entity shall not be compressed.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | No request for no compression |
| Bit b0 set to ONE: | Request for no compression |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

EXCLUSION OPTIONS

FID: #29
Data length: Bit Data
Data description: A bit field indicating that when the originating system recorded the data, the specified information was intended to be excluded from the File(s).

| Setting | Meaning |
|--------------------|----------------------------------|
| Bit b0 set to ZERO | Data stream not excluded |
| Bit b0 set to ONE | Data stream excluded |
| Bit b1 set to ZERO | Extended attributes not excluded |
| Bit b1 set to ONE | Extended attributes excluded |
| Bits b2 to b5 | Set to ZERO, no meaning |

EXECUTE ONLY

FID: #813B
Data length: Bit Data
Data description: This Field shall specify whether the entity is execute only.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | No execute only |
| Bit b0 set to ONE: | Execute only |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

EXPIRATION TIME

FID: #80F404

Data length: Fixed, 16 bytes

Data description: Timestamp of the expiration of the recorded entity, per agreement between the originating and receiving system.

FILE CHUNK SIZE

FID: #0B

Data length: Variable

Data description: The number of bytes contained by the following File within the current Buffer.

FILE CONTINUATION HEADER

FID: #8001

Data length: Variable

Data description: Resynchronization data, CRC or empty.

FILE HEADER

FID: #09

Data length: Variable

Data description: Resynchronization data, CRC or empty.

FILE INFORMATION

FID: #813F

Data length: Variable

Data description: Resynchronization data, CRC or empty.

FILE IS INVALID

FID: #80F003

Data length: Bit Data

Data description: This Field shall specify whether the originating systems deemed that the File as recorded is invalid.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | File not invalid |
| Bit b0 set to ONE: | File invalid |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

FILE MARK INTERVAL

FID: #808028
Data length: Variable
Data description: The distance in sectors between two consecutive file marks recorded at regular intervals.

FILE MARK USAGE

FID: #808020
Data length: Bit Data
Data description: This Field shall specify where physical file marks shall be recorded.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | No file marks before File Set Headers |
| Bit b0 set to ONE: | File marks before File Set Headers |
| Bit b1 set to ZERO: | No file marks before File Set Trailers |
| Bit b1 set to ONE: | File marks before File Set Trailers |
| Bit b2 set to ZERO: | No file marks at regular intervals |
| Bit b2 set to ONE: | File marks at the Sector Address interval indicated by FILE MARK INTERVAL. |
| Bits b3 to b5 | shall be set to ZERO, no meaning |

Note: Interval file marks are intended to provide a seek-performance improvement mechanism for devices which do not natively support direct seeking: the location of the interval file marks can facilitate high speed but measured travel of large media distances. Interval file mark use is strongly discouraged for devices that support direct seeking, because its use adds difficulty in processing non-interval file marks.

FILE SET ABORTED

FID: #808039
Data length: Bit Data
Data description: This Field shall specify whether the originating system aborted recording of the File Set prematurely.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | File not aborted prematurely |
| Bit b0 set to ONE: | File aborted prematurely |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

FILE SET COMMENT

FID: #80802B
Data length: Variable
Data description: An additional string associated with the File Set.

FILE SET CONTINUATION HEADER

FID: #808035
Data length: Variable
Data description: Resynchronization data, CRC or empty.

FILE SET HEADER

FID: #808004
Data length: Variable
Data description: Resynchronization data, CRC or empty.

FILE SET HEADER LOCATION

FID: #80803C
Data length: Variable
Data description: The Sector Number of the File Set Header.

FILE SET ID

FID: #8072
Data length: Fixed, 4 bytes
Data description: A non-zero 4-byte value assigned to a File Set by the originating system. This value shall be unique among Interleaved File Sets and among the File Sets of the Volume Set.

FILE SET INDEX

FID: #808010
Data length: Variable
Data description: Resynchronization data, CRC or empty.

FILE SET INDEX PRESENT

FID: #80802D
Data length: Bit Data
Data description: This Field shall specify whether a File Set Index is recorded in this File Set.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | File Set Index not necessarily present |
| Bit b0 set to ONE: | File Set Index present |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

FILE SET INDEX FIELDS

FID: #808034
Data length: Variable

Data description: A list of FIDs represented as a sequence of 32-bit numbers (i.e. little endian). This list specifies which Fields (including optional Fields) are included in the Iterated Field Set for each File listed in the File Set Index. It shall begin with the BUFFER OFFSET FID and end with the PATH FULLY QUALIFIED FID. Except for the first and last FID listed, no particular order is required. If a Field is not recorded for every File entry in the Index, its FID shall not be recorded in this list.

Example

The Data part of FILE SET INDEX FIELDS for a File Set Index which includes MODIFIED TIME for each entry is:

#14 #80 #80 #00 #74 #00 #00 #00 #50 #00 #00 #00

FILE SET LABEL

FID: #808005

Data length: Variable

Data description: A string generated by the originating system intended to describe the contents of the File Set.

FILE SET SUBINDEX

FID: #808033

Data length: Variable

Data description: Resynchronization data, CRC or empty.

FILE SET TIME

FID: #80F403

Data length: Fixed, 16 bytes

Data description: The Timestamp when the originating system began recording the File Set.

FILE SET TRAILER

FID: #808009

Data length: Variable

Data description: Resynchronization data, CRC or empty.

FILE SET TRAILER LOCATION

FID: #80803F

Data length: Variable

Data description: The Sector Number of the File Set Trailer.

FILE TYPE

FID: #70

Data length: Fixed, 1 byte

Data description: In a File Header Field Table, specifies which of the following types of File is contained in the ensuing File record and File Continuation record(s).

| Value | Type |
|-------|--------------------|
| 0 | Reserved |
| 1 | Implementation use |
| 2 | Logical volume |
| 3 | Source directory |
| 4 | File |
| 5 | Transaction set |

No other values shall be recorded

FORMAT NAME

FID: #8052

Data length: Fixed, 4 bytes

Data description: The Bytes Stream #53 #49 #44 #46 (“SIDF” not NULL-terminated). This identifies the Volume with a defined sequence to be SIDF compliant.

FORMAT VERSION

FID: #8062

Data length: Fixed, 4 bytes

Data description: Version of the Logical Format. A Byte Sequence, which specifies the version as shown below:

| Byte | Meaning |
|------|---|
| B0 | Major version number. |
| B1 | Minor version number. |
| B2 | Subminor version number. |
| B3 | Undefined, this byte shall be set to all ZEROS. |

FSH PARTITION NUMBER

FID: #80803B

Data length: Variable

Data description: Specifies the physical partition number of the File Set Header.

FSH VOLUME SET SEQUENCE

FID: #80803A

Data length: Variable

Data description: Specifies the Volume Set sequence number of the File Set Header.

FST PARTITION NUMBER

FID: #80803E
Data length: Variable
Data description: Specifies the physical partition number of the File Set Trailer.

FST VOLUME SET SEQUENCE

FID: #80803D
Data length: Variable
Data description: Specifies the Volume Set sequence number of the File Set Header.

HEADER DEBUG STRING

FID: #81EFFF
Data length: Variable
Data description: Implementation use: recorded after the OFFSET TO END Field to put the Field Table name coded according to the IRV of ECMA-6 in the data for debugging.

HIDDEN

FID: #15
Data length: Bit Data
Data description: This Field shall specify whether the Source file is marked by the Source as hidden.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Source file is not hidden |
| Bit b0 set to ONE: | Source file is hidden |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

INDEXED

FID: #81EFF8
Data length: Bit Data
Data description: This Field shall specify whether the Source file is marked by the Source as index.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Source file is not indexed |
| Bit b0 set to ONE: | Source file is indexed |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

INHIBITIONS

FID: #813A

Data length: Bit Data

Data description: This Field shall specify the functionality of the file which is prohibited by the Source. If this Field is not part of the Characteristics Field Table, the Source file shall be considered as having none of the inhibitions specified by bits b0 to bit b2.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Copying of this file allowed |
| Bit b0 set to ONE: | Copying of this file prohibited |
| Bit b1 set to ZERO: | Deletion of this file allowed |
| Bit b1 set to ONE: | Deletion of this file prohibited |
| Bit b2 set to ZERO: | Rename of this file allowed |
| Bit b2 set to ONE: | Rename of this file prohibited |
| Bits b3 to b5 | shall be set to ZERO, no meaning |

MODIFIED TIME

FID: #74

Data length: Fixed, 16 bytes

Data description: The Timestamp when the Source file was last modified.

NAME POSITIONS

FID: #27

Data length: Variable

Data description: An array of 16-bit numbers containing the beginning position of each path element in the associated PATH NAME Field. Position counting shall be in terms of bytes, and shall begin with 0. All occurrences of NAME POSITIONS shall be associated with a specific PATH NAME Field. Certain properties of this array are associated with the Name Space of the path name.

Example

The Data Length part of NAME POSITION for file name "volume2:bin\foo" is:
#00 #00 #08 #00 #0D #00.

NAME SPACE

FID: #11

Data length: Variable

Data description: The Source file Name Space type value. Name Space specifies several characteristics of Source file name use. When this Field is part of an Iterated Field Set, the first instance of this Field shall indicate the Name Space that was used to create the file, and the Fields containing the other Name Spaces shall follow.

NEEDS ARCHIVE

FID: #16

Data length: Bit Data

Data description: This Field shall specify whether at least one of the Source file Streams has been modified since the last time the Source file was marked as archived.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | No Source file Stream modified |
| Bit b0 set to ONE: | One or more Source file Streams modified |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

NEEDS ARCHIVE CHARACTERISTICS

FID: #2D

Data length: Bit Data

Data description: This Field shall specify whether Source file information other than Stream(s) has been modified since the last time the Source file was marked as archived.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | No information other than Stream(s) modified |
| Bit b0 set to ONE: | Information other than Stream(s) modified |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

NEXT OBJECT LOCATION

FID: #808029

Data length: Variable

Data description: The Sector Number of the Object that most immediately follows the current Field Table. In this context, Objects are all File Set Headers and Trailers. If no Object is recorded on the Volume after the current Field Table, the location shall be recorded as 0.

NUMBER OF FILES

FID: #808021

Data length: Variable

Data description: The number of Files in a File Set.

NUMBER OF FILE SETS

FID: #808015

Data length: Variable

Data description: The number of File Sets the Volume Index references.

OFFSET TO END

FID: #01

Data length: Variable

Data description: The distance in bytes from the beginning of the Field following this OFFSET TO END Field, to the beginning of the last Field in the Field table. A value of 0 shall mean that this Field shall be ignored.

ORIGINATING SYSTEM SOFTWARE NAME

FID: #808006

Data length: Variable

Data description: An implementation use string intended to identify the originating system software which recorded the File Set.

ORIGINATING SYSTEM SOFTWARE TYPE

FID: #808007

Data length: Variable

Data description: An implementation use string intended to identify the type of the originating system software which recorded the File Set.

ORIGINATING SYSTEM SOFTWARE VERSION

FID: #808008

Data length: Variable

Data description: An implementation use string intended to identify the version of the originating system software which recorded the File Set.

PARENT

FID: #81F0FD

Data length: Fixed, 1 byte

Data description: This Field shall specify whether the Source is a parent.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Source is not a parent |
| Bit b0 set to ONE: | Source is a parent |
| Bits b1 to b7 | shall be set to ZERO, no meaning |

PARTITION NUMBER

FID: #808012

Data length: Variable

Data description: The physical partition number that contains the referenced entity.

PATH

FID: #10
Data length: Variable
Data description: Resynchronization data, CRC or empty.

PATH FULLY QUALIFIED

FID: #50
Data length: Fixed, 1 byte
Data description: This Field shall specify whether the associated path is a complete path.

If the associated path for the referenced entity is a relative path, the complete path shall be constructed as follows: the concatenation of the path of the closest previous File the PARENT Field bit b0 of which is not set to ZERO, and the relative path of the referenced entity.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Associated path is relative |
| Bit b0 set to ONE: | Associated path is complete |
| Bits b1 to b7 | shall be set to ZERO, no meaning |

PATH NAME

FID: #12
Data length: Variable
Data description: Path name string.

POSIX FILE MODE

FID: #80F203

Data length: Fixed, 4 bytes

Data description: Contains the POSIX file access mode permissions and file type information. If a bit is set to ONE, the corresponding permission or property indicated below is implied by the Source, else the bit(s) shall be set to ZERO.

| Bit | Meaning |
|-----|--|
| b0 | Execute by others (i.e. other than group or owner) |
| b1 | Write by others |
| b2 | Read by others |
| b3 | Execute by group |
| b4 | Write by group |
| b5 | Read by group |
| b6 | Execute by owner |
| b7 | Write by owner |
| b8 | Read by owner |
| b9 | Reserved no meaning |
| b10 | Set group ID |
| b11 | Set user ID |
| b12 | FIFO |
| b13 | Character special file |
| b14 | Source directory |

Bits b15 to b31 shall be set to ZERO, no meaning.

POSIX FILE ID

FID: #80F210

Data length: Fixed, 4 bytes

Data description: The POSIX inode number of the file. Used by the receiving system to rebuild links.

POSIX FILE SYSTEM ID

FID: #80F20F

Data length: Fixed, 4 bytes

Data description: The file system ID, used with the POSIX FILE ID to uniquely identify the Source file.

POSIX GROUP ID

FID: #80F204

Data length: Fixed, 4 bytes

Data description: Contains the Source file POSIX group ID.

POSIX NUMBER OF LINKS

FID: #80F20D
Data length: Fixed, 4 bytes
Data description: The number of hard links for the Source file.

POSIX OWNER ID

FID: #80F209
Data length: Fixed, 4 bytes
Data description: The POSIX user ID of the Source file owner.

POSIX RDEVICE

FID: #80F20E
Data length: Fixed, 4 bytes
Data description: Raw device number, as indicated by the Source

PREV OBJECT LOCATION

FID: #80802A
Data length: Variable
Data description: The Sector Number of the Object that immediately precedes the current Field Table. In this context, Objects are all File Set Headers and Trailers. If no Object is recorded on the Volume before the current Field Table, the location shall be recorded as 0.

PURGE

FID: #8136
Data length: Bit Data
Data description: This Field shall specify whether a Source file shall be immediately purged after deletion.

| Setting | Meaning |
|---------------------|--|
| Bit b0 set to ZERO: | Purging not required |
| Bit b0 set to ONE: | File to be purged immediately after deletion |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

READ ONLY

FID: #17

Data length: Bit Data

Data description: This Field shall specify whether the Source file is read only.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Source file is not read only |
| Bit b0 set to ONE: | Source file is read only |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

REGISTERED IDENTIFIER

FID: #808043

Data length: Variable

Data description: A Byte Sequence of 23 bytes containing a registered identifier in conformance with ISO/IEC 13800.

A registered identifier is a construct used to specify that a set of extensions to this ECMA Standard are to be applied to some part of the recorded media. That part of the media is known as the scope of the registered identifier. The identifier indicates which specific extensions are to be applied.

This Field shall be recorded only in a Volume Header Field Table, File Set Header Field Table or a File Header Field Table. If recorded in a Volume Header Field Table, its scope is that Volume. If recorded in a File Set Header Field Table, its scope is that entire File Set, even if the File Set spans multiple Volumes. If recorded in a File Header Field Table, its scope is that entire File, even if the File spans multiple Buffers.

Extensions to this ECMA Standard invoked by a registered identifier shall be limited to the following. The creating body associated with the registered identifier shall supply any needed supplemental usage specifications for the created entities.

- Creation of new FIDs, which shall be constructed in accordance with annex A of this ECMA Standard, and shall be unique and registered in compliance with ISO/IEC 13800.
- Creation of new data value definitions for existing Fields, provided they create no conflicts with this ECMA Standard or use associated with other registered identifiers.
- Establishment of Field ordering conventions, including the extended use of Iterated Field Sets.

REMOTE DATA ACCESS

FID: #81EFE7

Data length: Bit Data

Data description: This Field shall specify whether the Source has indicated that its copy of the data stream has been stored remotely.

| Setting | Meaning |
|---------------------|---|
| Bit b0 set to ZERO: | Copy of data stream not stored remotely |
| Bit b0 set to ONE: | Copy of data stream stored remotely |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

REMOTE DATA INHIBIT

FID: #81EFE8

Data length: Bit Data

Data description: This Field shall specify whether the Source indicated that this entity shall not be stored remotely.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Remote storage allowed |
| Bit b0 set to ONE: | Remote storage not allowed |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

RESOURCE NAME

FID: #808023

Data length: Variable

Data description: The File name of the root object among the Files of the File Set.

Example

If a File Set contains sys:usr/bin and all its recursive subdirectories and the files within them, this data would be sys:usr/bin)

RESOURCE NAME SPACE

FID: #808038

Data length: Variable

Data description: The Name Space of the associated RESOURCE NAME Field (see NAME SPACE).

RESOURCE TYPE

FID: #808037

Data length: Variable

Data description: Per agreement between the originating and receiving systems. The data values to be used for this Field may be determined by a REGISTERED IDENTIFIER of the appropriate scope.

SECTOR SIZE

FID: #80800E

Data length: Variable

Data description: Sector size in bytes for the Volume.

SEPARATOR POSITIONS

FID: #28

Data length: Variable

Data description: A separator is a sequence of one or more non-NULL characters used to differentiate distinct elements of a Source entity name. Specification of which separator values are to be used is made by the NAME SPACE value relevant to this Field.

The Data of SEPARATOR POSITIONS shall be an array of 16-bit numbers containing the beginning position of each path separator in the associated PATH NAME Field. The first array element shall contain the position of the first separator, the second array element shall contain the position of the second separator, and so forth. Position counting shall be in terms of bytes, and shall begin with 0.

If the Source entity associated with this Field has the PARENT Field in its Characteristics Field Table set to one, it is considered a parent (for example, a Source directory). Unless specified otherwise by the associated NAME SPACE Field, the following requirements shall be met:

- If a Source entity is a parent, the separator array shall have, as its last element, the position after the last non-separator character of the name.
- If a Source entity is not a parent, the separator array shall have, as its last element, 16 bits set to ZERO.

In all cases, the number of array elements shall equal the number of elements in the associated NAME POSITIONS array, if recorded.

Example

The Data part of SEPARATOR POSITION for the non-parent source entity named "volume2::bin\foo" is: #07 #00 #0D #00 #00 #00.

SHAREABLE

FID: #18

Data length: Bit Data

Data description: This Field shall specify whether the Source entity is marked as shareable.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Entity not shareable |
| Bit b0 set to ONE: | Entity shareable |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

SOURCE ALIAS

FID: #808036

Data length: Variable

Data description: A string intended to specify an alternate name for the Source. This string shall be associated with the closest previously recorded SOURCE NAME TYPE of the Field Table.

SOURCE FILE HEADER

FID: #0E

Data length: Variable

Data description: Resynchronization data, CRC or empty.

SOURCE FILE TRAILER

FID: #0F

Data length: Variable

Data description: Resynchronization data, CRC or empty.

SOURCE NAME

FID: #02

Data length: Variable

Data description: A string intended to specify the Source name. The usage of this string is defined by FILE SET REGID.

SOURCE NAME TYPE

FID: #8009

Data length: Variable

Data description: Per agreement between the originating and receiving systems. The data values to be used for this Field may be determined by a REGISTERED IDENTIFIER of the appropriate scope.

SOURCE OPERATING SYSTEM

FID: #03

Data length: Variable

Data description: Per agreement between the originating and receiving systems. The data values to be used for this Field may be determined by a REGISTERED IDENTIFIER of the appropriate scope.

SOURCE OPERATING SYSTEM VERSION

FID: #04

Data length: Variable

Data description: An implementation use string intended to describe the Source operating system version.

SOURCE VOLUME HEADER

FID: #81EFFC

Data length: Variable

Data description: Resynchronization data, CRC or empty.

SOURCE VOLUME TRAILER

FID: #81EFFB

Data length: Variable

Data description: Resynchronization data, CRC or empty.

STREAM COMPRESS TYPE

FID: #8005

Data length: Variable

Data description: The type of compression used on the stream. Interpretation of this may be within the scope of a REGISTERED IDENTIFIER.

STREAM CRC

FID: #22

Data length: Variable

Data description: The CRC bits for the stream.

STREAM EXPANDED SIZE

FID: #8006

Data length: Variable

Data description: The logical size of the stream, for example the uncompressed size of a compressed stream, or the size represented by a sparse File.

STREAM FORMAT

FID: #2C

Data length: Variable

Data description: One of the five types of the physical representation used to record a Stream on the media. This is not necessarily the physical representation of the Source stream.

| Value | Type |
|-------|--------------|
| 0 | Clear data |
| 1 | Sparse |
| 2 | Compressed |
| 3 | Delta block |
| 4 | Delta extent |

No other values shall be recorded.

STREAM HEADER

FID: #1D

Data length: Variable

Data description: Resynchronization data, CRC or empty.

STREAM IS INVALID

FID: #21

Data length: Bit Data

Data description: This Field shall specify whether the originating system deemed that the Stream was invalid.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | Stream not invalid |
| Bit b0 set to ONE: | Stream invalid |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

STREAM SIZE

FID: #20

Data length: Variable

Data description: The size in bytes of the Stream as recorded on the media. This may be different from the logical, or expanded, size of the source stream, for example, as with a sparse format Stream.

STREAM TRAILER

FID: #1E

Data length: Variable

Data description: Resynchronization data, CRC or empty.

STREAM TYPE

FID: #2B

Data length: Variable

Data description: One of the following 7 types of Stream.

| Value | Type |
|--------|----------------------|
| 0 | Data |
| 1 | Resource |
| 2 | FTAM |
| 3 to 9 | Reserved, no meaning |
| 10 | Extended attributes |
| 11 | Alternate data |
| 12 | Security data |
| 13 | Link data |

No other values shall be recorded.

STREAM TYPE SEQUENCE

FID: #61

Data length: Fixed, 2 bytes

Data description: Ordinal sequence number (1, 2, 3, ..., n) of Stream for a given STREAM TYPE.

SYSTEM

FID: #19

Data length: Bit Data

Data description: This Field shall specify whether the Source indicated that this entity has the system characteristics.

| Setting | Meaning |
|---------------------|------------------------------------|
| Bit b0 set to ZERO: | File has no system characteristics |
| Bit b0 set to ONE: | File has system characteristics |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

TOTAL FILE SET SIZE

FID: #808022

Data length: Variable

Data description: The size of the File Set expressed in units of 1 024 bytes. This value is informative.

TOTAL STREAM SIZE

FID: #81F2FA

Data length: Fixed, 4 bytes

Data description: The number of 4 096-byte blocks occupied by all Streams.

TRANSACTION SET HEADER

FID: #81EFF3
Data length: Variable
Data description: Resynchronization data, CRC or empty.

TRANSACTION SET NAME

FID: #81EFEF
Datalength: Variable
Data description: Name string of a Transaction set File.

TRANSACTION SET TRAILER

FID: #81EFF2
Data length: Variable
Data description: Resynchronization data, CRC or empty.

TRANSACTION SET TYPE

FID: #81EFEE
Data length: Variable
Data description: Per agreement between the originating and receiving system.

TRANSACTIONAL

FID: #8135
Data length: Bit Data
Data description: This Field shall speciy whether the Source has indicated that this File shall be tracked for transaction processing.

| Setting | Meaning |
|---------------------|---|
| Bit b0 set to ZERO: | File not tracked for transaction processing |
| Bit b0 set to ONE: | File tracked for transaction processing |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

UNUSED IN THIS BUFFER

FID: #8000
Data length: Variable
Data description: The number of bytes not used at the end of the Buffer.

VOLUME HEADER

FID: #808000

Data length: Variable

Data description: Resynchronization data, CRC or empty.

VOLUME HEADER WRITE COUNT

FID: #80F104

Data length: Fixed, 2 bytes

Data description: The number of times a Volume Header has been recorded on this medium. This is an informative estimate only.

VOLUME INDEX

FID: #808011

Data length: Variable

Data description: Resynchronization data, CRC or empty.

VOLUME INDEX LOCATION

FID: #808042

Data length: Variable

Data description: Sector Number of last Volume Index on Volume, intended for use on update in place media (i.e. magneto optical).

VOLUME INDEX REQUIRED

FID: #80802F

Data length: Bit Data

Data description: This Field in the Volume Header Field Table shall specify whether all originating systems that record to this Volume shall record a Volume Index after each File Set or set of Interleaved File Sets.

| Setting | Meaning |
|---------------------|----------------------------------|
| Bit b0 set to ZERO: | No Volume Index required |
| Bit b0 set to ONE: | Volume Index required |
| Bits b1 to b5 | shall be set to ZERO, no meaning |

VOLUME LABEL

FID: #808027

Data length: Variable

Data description: An implementation use string. There are no requirements for this string relative to Volume Labels for any other Volume in a Volume Set.

VOLUME SET ALIAS

FID: #808041
Data length: Variable
Data description: An alternate label string of the Volume Set.

VOLUME SET LABEL

FID: #808030
Data length: Variable
Data description: The label string of the Volume Set. The VOLUME SET LABEL, and its character specification, shall be the same for each Volume in a Volume Set.

VOLUME SET SEQUENCE

FID: #80F100
Data length: Fixed, 2 bytes
Data description: The ordinal sequence number of a Volume within the Volume Set (1, 2, 3, ..., n).

VOLUME SET TIME

FID: #80F400
Data length: Fixed, 16 bytes
Data description: The Timestamp when the Volume Set was created.

VOLUME SIZE

FID: #80F201
Data length: Fixed, 4 bytes
Data description: Size of the Volume in Sectors. This data is informative only.

VOLUME SUBINDEX

FID: #808031
Data length: Variable
Data description: Resynchronization data, CRC or empty.

VOLUME TIME

FID: #80F401

Data length: Fixed, 16 bytes

Data description: The Timestamp when this individual Volume was created.

VOLUME TRAILER

FID: #808003

Data length: Variable

Data description: Resynchronization Data, CRC, or empty.

Annex D
(informative)

Numerical list of the Field Identifiers specified by this ECMA Standard

Field Identifiers marked as “see annex E” are listed in annex E.

| | |
|-----|---------------------------------|
| #01 | OFFSET TO END |
| #02 | SOURCE NAME |
| #03 | SOURCE OPERATING SYSTEM |
| #04 | SOURCE OPERATING SYSTEM VERSION |
| #05 | BUFFER HEADER |
| #06 | BUFFER SIZE |
| #07 | BUFFER SEQUENCE |
| #08 | BUFFER ADDRESS |
| #09 | FILE HEADER |
| #0A | see annex E |
| #0B | FILE CHUNK SIZE |
| #0C | SOURCE DIRECTORY HEADER |
| #0D | SOURCE DIRECTORY TRAILER |
| #0E | SOURCE FILE HEADER |
| #0F | SOURCE FILE TRAILER |
| #10 | PATH |
| #11 | NAME SPACE |
| #12 | PATH NAME |
| #13 | CHARACTERISTICS |
| #14 | SOURCE DIRECTORY |
| #15 | HIDDEN |
| #16 | NEEDS ARCHIVE |
| #17 | READ ONLY |
| #18 | SHAREABLE |
| #19 | SYSTEM |
| #1A | see annex E |
| #1B | EA KEY |
| #1C | see annex E |
| #1D | STREAM HEADER |
| #1E | STREAM TRAILER |

| | |
|-------|-------------------------------|
| #1F | STREAM NAME |
| #20 | STREAM SIZE |
| #21 | STREAM IS INVALID |
| #22 | STREAM CRC |
| #23 | see annex E |
| #24 | BLOCK SIZE |
| #25 | BLOCK MAP |
| #26 | see annex E |
| #27 | NAME POSITIONS |
| #28 | SEPARATOR POSITIONS |
| #29 | EXCLUSION OPTIONS |
| #2A | see annex E |
| #2B | STREAM TYPE |
| #2C | STREAM FORMAT |
| #2D | NEEDS ARCHIVE CHARACTERISTICS |
| #2E | see annex E |
| #40 | see annex E |
| #41 | see annex E |
| #42 | see annex E |
| #44 | ACCESS TIME |
| #45 | see annex E |
| #50 | PATH FULLY QUALIFIED |
| #51 | see annex E |
| #52 | see annex E |
| #54 | ARCHIVE TIME |
| #60 | BUFFER TYPE |
| #61 | STREAM TYPE SEQUENCE |
| #62 | see annex E |
| #64 | CREATION TIME |
| #70 | FILE TYPE |
| #72 | see annex E |
| #74 | MODIFIED TIME |
| #8000 | UNUSED IN THIS BUFFER |
| #8001 | FILE CONTINUATION HEADER |
| #8002 | AUTHENTICATION |
| #8003 | see annex E |
| #8004 | see annex E |
| #8005 | STREAM COMPRESS TYPE |

| | |
|-------|-------------------------|
| #8006 | STREAM EXPANDED SIZE |
| #8007 | see annex E |
| #8008 | BUFFER CRC |
| #8009 | SOURCE NAME TYPE |
| #800A | DELTA EXTENT OFFSET |
| #800B | DELTA EXTENT OLD SIZE |
| #8042 | see annex E |
| #8044 | DELTA BASE TIME |
| #8052 | see annex E |
| #8062 | FORMAT VERSION |
| #8072 | FILE SET ID |
| #8113 | see annex E |
| #8114 | see annex E |
| #8115 | DO NOT COMPRESS FILE |
| #8116 | COMPRESS FILE IMMEDIATE |
| #8117 | see annex E |
| #8118 | see annex E |
| #8119 | see annex E |
| #811A | see annex E |
| #8120 | see annex E |
| #8121 | see annex E |
| #8122 | see annex E |
| #8123 | see annex E |
| #8124 | see annex E |
| #8125 | see annex E |
| #8126 | see annex E |
| #8127 | see annex E |
| #8128 | see annex E |
| #8129 | see annex E |
| #812A | see annex E |
| #812B | see annex E |
| #812C | see annex E |
| #812D | see annex E |
| #812E | see annex E |
| #812F | see annex E |
| #8130 | see annex E |
| #8131 | see annex E |
| #8132 | see annex E |

| | |
|---------|-------------------------------------|
| #8133 | see annex E |
| #8134 | see annex E |
| #8135 | TRANSACTIONAL |
| #8136 | PURGE |
| #8137 | see annex E |
| #8138 | see annex E |
| #8139 | see annex E |
| #813A | INHIBITIONS |
| #813B | EXECUTE ONLY |
| #813C | see annex E |
| #813D | see annex E |
| #813E | see annex E |
| #813F | FILE INFORMATION |
| #8152 | see annex E |
| #808000 | VOLUME HEADER |
| #808001 | see annex E |
| #808002 | see annex E |
| #808003 | VOLUME TRAILER |
| #808004 | FILE SET HEADER |
| #808005 | FILE SET LABEL |
| #808006 | ORIGINATING SYSTEM SOFTWARE NAME |
| #808007 | ORIGINATING SYSTEM SOFTWARE TYPE |
| #808008 | ORIGINATING SYSTEM SOFTWARE VERSION |
| #808009 | FILE SET TRAILER |
| #80800A | see annex E |
| #80800B | see annex E |
| #80800C | see annex E |
| #80800D | see annex E |
| #80800E | SECTOR SIZE |
| #80800F | DATABASE LOCATION METHOD |
| #808010 | FILE SET INDEX |
| #808011 | VOLUME INDEX |
| #808012 | PARTITION NUMBER |
| #808013 | see annex E |
| #808014 | BUFFER OFFSET |
| #808015 | NUMBER OF FILE SETS |
| #808016 | NUMBER OF DATABASES |
| #808017 | PREVIOUS MEDIA INDEX |

| | |
|---------|--|
| #808018 | DATABASE NAME |
| #808019 | BLANK SPACE |
| #80801A | see annex E |
| #80801B | see annex E |
| #80801C | see annex E |
| #80801D | see annex E |
| #80801E | see annex E |
| #80801F | see annex E |
| #808020 | FILE MARK USAGE |
| #808021 | NUMBER OF FILES |
| #808022 | TOTAL FILE SET SIZE |
| #808023 | RESOURCE NAME |
| #808024 | COMPRESSION TYPE (reserved for future use) |
| #808025 | ENCRYPTION TYPE (reserved for future use) |
| #808026 | see annex E |
| #808027 | VOLUME LABEL |
| #808028 | FILE MARK INTERVAL |
| #808029 | NEXT OBJECT LOCATION |
| #80802A | PREV OBJECT LOCATION |
| #80802B | FILE SET COMMENT |
| #80802C | see annex E |
| #80802D | FILE SET INDEX PRESENT |
| #80802F | VOLUME INDEX REQUIRED |
| #808030 | VOLUME SET LABEL |
| #808031 | VOLUME SUBINDEX |
| #808032 | DEVICE INFO |
| #808033 | FILE SET SUBINDEX |
| #808034 | FILE SET INDEX FIELDS |
| #808035 | FILE SET CONTINUATION HEADER |
| #808036 | SOURCE ALIAS |
| #808037 | RESOURCE TYPE |
| #808038 | RESOURCE NAME SPACE |
| #808039 | FILE SET ABORTED |
| #80803A | FSH VOLUME SET SEQUENCE |
| #80803B | FSH PARTITION NUMBER |
| #80803C | FILE SET HEADER LOCATION |
| #80803D | FST VOLUME SET SEQUENCE |
| #80803E | FST PARTITION NUMBER |

| | |
|---------|---------------------------|
| #80803F | FILE SET TRAILER LOCATION |
| #808040 | CHAR SPEC |
| #808041 | VOLUME SET ALIAS |
| #808042 | VOLUME INDEX LOCATION |
| #80F000 | see annex E |
| #80F001 | see annex E |
| #80F002 | see annex E |
| #80F003 | FILE IS INVALID |
| #80F100 | VOLUME SET SEQUENCE |
| #80F101 | see annex E |
| #80F102 | see annex E |
| #80F104 | MEDIA USAGE COUNT |
| #80F200 | see annex E |
| #80F201 | VOLUME SIZE |
| #80F202 | see annex E |
| #80F203 | POSIX FILE MODE |
| #80F204 | POSIX GROUP OWNER ID |
| #80F205 | see annex E |
| #80F206 | see annex E |
| #80F207 | see annex E |
| #80F208 | see annex E |
| #80F209 | POSIX OWNER ID |
| #80F20A | see annex E |
| #80F20B | see annex E |
| #80F20C | see annex E |
| #80F20D | POSIX NUMBER OF LINKS |
| #80F20E | POSIX RDEVICE |
| #80F20F | POSIX FSID |
| #80F210 | POSIX FILEID |
| #80F300 | see annex E |
| #80F400 | VOLUME SET TIME |
| #80F401 | VOLUME TIME |
| #80F402 | CLOSE TIME |
| #80F403 | FILE SET TIME |
| #80F404 | EXPIRATION TIME |
| #81EFE6 | CANT COMPRESS DATA |
| #81EFE7 | REMOTE DATA ACCESS |
| #81EFE8 | REMOTE DATA INHIBIT |

| | |
|---------|-------------------------|
| #81EFE9 | see annex E |
| #81EFEA | see annex E |
| #81EFEB | see annex E |
| #81EFEC | see annex E |
| #81EFED | see annex E |
| #81EFEE | TRANSACTION SET TYPE |
| #81EFEF | TRANSACTION SET NAME |
| #81EFF0 | see annex E |
| #81EFF1 | see annex E |
| #81EFF2 | TRANSACTION SET TRAILER |
| #81EFF3 | TRANSACTION SET HEADER |
| #81EFF4 | see annex E |
| #81EFF5 | see annex E |
| #81EFF6 | see annex E |
| #81EFF7 | see annex E |
| #81EFF8 | INDEXED |
| #81EFF9 | see annex E |
| #81EFFA | see annex E |
| #81EFFB | SOURCE VOLUME TRAILER |
| #81EFFC | SOURCE VOLUME HEADER |
| #81EFFD | see annex E |
| #81EFFE | see annex E |
| #81EFFF | HEADER DEBUG STRING |
| #81F0FC | see annex E |
| #81F0FD | PARENT |
| #81F0FE | DELETED FLAG |
| #81F0FF | MODIFIED FLAG |
| #81F2F6 | see annex E |
| #81F2F7 | see annex E |
| #81F2F8 | see annex E |
| #81F2F9 | see annex E |
| #81F2FA | TOTAL STREAM SIZE |
| #81F2FC | CREATOR NAME SPACE |
| #81F2FD | see annex E |
| #81F2FE | ATTRIBUTES |
| #81F2FF | see annex E |

Annex E
(informative)

Numerical list of Field Identifiers which are deprecated (D) or not specified by this ECMA Standard

| | |
|-------|---|
| #0A | DATA RECORD SEQUENCE (D) |
| #1A | EXTENDED ATTRIBUTES (D) |
| #1C | EA VALUE (D) |
| #2A | CODE PAGE (D) |
| #23 | SPARSE STREAM HEADER (D) |
| #26 | MAC CHARACTERISTICS (D) |
| #2E | SECTOR ADDRESS (D) |
| #40 | LAST BUFFER IN FILE SET (D) |
| #41 | ACCESS DATE (D) |
| #42 | ARCHIVE TIME (D) |
| #45 | MAC FINDER INFO |
| #51 | ACCESS TIME (D) |
| #52 | ARCHIVE DATE AND TIME (D) |
| #62 | CREATION DATE AND TIME (D) |
| #72 | MODIFIED DATE AND TIME (D) |
| #8003 | MAC SYSTEM SCRIPT |
| #8004 | MAC BLESSED FOLDER |
| #8007 | STREAM MIGRATION KEY |
| #8042 | FILE SET TIME OLD (D) |
| #8052 | FORMAT NAME |
| #8113 | SOURCE DIRECTORY OBJECT DATA |
| #8114 | SOURCE DIRECTORY OBJECT NAME |
| #8117 | ARCHIVER NAME |
| #8118 | TRUSTEE NAME |
| #8119 | OWNER NAME |
| #811A | SOURCE DIRECTORY OBJECT TRAILER |
| #8120 | MODIFIER NAME |
| #8121 | SUPPORT MODULE ROOT BIT |
| #8122 | TRANSACTION SET DATA STREAM TRAILER (D) |
| #8123 | TRANSACTION SET DATA STREAM HEADER (D) |
| #8124 | TRANSACTION SET TRUSTEE HEADER (D) |

| | |
|---------|--|
| #8125 | TRANSACTION SET EXTENDED ATTRIBUTES (D) |
| #8126 | TRANSACTION SET MAC CHARACTERISTICS (D) |
| #8127 | TRANSACTION SET CHARACTERISTICS (D) |
| #8128 | TRANSACTION SET PATH NAME (D) |
| #8129 | TRANSACTION SET NAME SPACE (D) |
| #812A | TRANSACTION SET SEPARATOR POSITIONS (D) |
| #812B | TRANSACTION SET NAME POSITIONS (D) |
| #812C | MAC DIR RIGHTS MASK |
| #812D | MAC PRO DOS INFO |
| #812E | TRANSACTION SET REQUESTED NAME SPACE (D) |
| #812F | TRUSTEE |
| #8130 | TRUSTEE ID |
| #8131 | TRUSTEE HEADER |
| #8132 | EA ACCESS |
| #8133 | DIR SPACE RESTRICTIONS |
| #8134 | FTAM CHARACTERISTICS |
| #8137 | OWNER ID |
| #8138 | MODIFIER ID |
| #8139 | LOW MED HIGH SEARCH |
| #813C | MAC INHIBITIONS |
| #813D | ARCHIVER ID |
| #813E | FILE NAME (D) |
| #8152 | INHERITED RIGHTS MASK |
| #808001 | MEDIA OWNER (D) |
| #808002 | CRC TYPE (D) |
| #80800A | POSIX CHARACTERISTICS (D) |
| #80800B | HARD LINK PATHNAME |
| #80800C | HARD LINK USER ID |
| #80800D | HARD LINK DELIMITER |
| #808013 | MEDIA SECTOR ADDRESS (D) |
| #80801A | SMM HEADER (D) |
| #80801B | SMM TYPE (D) |
| #80801C | SMM PREV FILE MARK (D) |
| #80801D | SMM NEXT FILE MARK (D) |
| #80801E | SMM PREV SET MARK (D) |
| #80801F | SMM NEXT SET MARK (D) |
| #808026 | PRIOR MEDIA TYPE (D) |
| #80802C | FILE SET INDEX ADDRESS (D) |

| | |
|---------|---|
| #80F000 | POSIX LINKED FLAG |
| #80F001 | POSIX HARD LINK PATHS |
| #80F002 | POSIX ACS FLAGS |
| #80F101 | MEDIA TYPE (D) |
| #80F102 | SPANNED SECTOR COUNT (D) |
| #80F200 | MEDIA OPEN TIME (D) |
| #80F202 | CLOSE TIME OLD (D) |
| #80F205 | POSIX RDEVICE OLD (D) |
| #80F206 | POSIX NUMBER OF LINKS OLD (D) |
| #80F207 | HARD LINK GROUP ID |
| #80F208 | HARD LINK FILE ACCESS MODE |
| #80F20A | MY FLAGS |
| #80F20B | MEDIA OPEN TIME CAL (D) |
| #80F20C | MS OPEN TIME CAL (D) |
| #80F300 | REVISION LEVEL (D) |
| #81EFE5 | DONT SUB ALLOCATE BIT |
| #81EFE6 | SOURCE VOLUME NAME |
| #81EFE9 | SOURCE VOLUME RESTRICTIONS NAME |
| #81EFEA | 286 BINDERY TRAILER |
| #81EFEB | 286 BINDERY HEADER |
| #81EFEC | TRANSACTION SET VOLUME RESTRICTIONS (D) |
| #81EFED | TRANSACTION SET POSIX CHARACTERISTICS (D) |
| #81EFF0 | TRANSACTION SET FILE TRAILER (D) |
| #81EFF1 | TRANSACTION SET FILE HEADER (D) |
| #81EFF4 | SOURCE VOLUME RESTRICTIONS LIMIT |
| #81EFF5 | SOURCE VOLUME RESTRICTIONS ID |
| #81EFF7 | SOURCE VOLUME RESTRICTIONS |
| #81EFF9 | BINDERY TRAILER) |
| #81EFFA | BINDERY HEADER |
| #81EFFD | OTHER INFO |
| #81EFFE | MEDIA SET LABEL (D) |
| #81F0FC | POSIX FIRST CREATOR FLAG |
| #81F2F6 | ARCHIVE TIME (D) |
| #81F2F7 | MODIFIED TIME (D) |
| #81F2F8 | CREATE TIME (D) |
| #81F2F9 | ACCESSED TIME (D) |
| #81F2FD | CREATOR ID |
| #81F2FF | MEDIA SET OPEN TIME (D) |

This Standard ECMA-208 is available free of charge from:

**ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland**

**Fax: +41 22 849.60.01
Internet: helpdesk@ecma.ch**

This Standard can also be downloaded as file E208-DOC.EXE or E208-PSC.EXE from ECMANEWS