American National Standard

Adopted for Use by the Federal Government

FIPS

FIPS PUB 123

See Notice on Inside Front Cover

for information systems –

specification for a data descriptive file for information interchange

american national standards institute, inc.
1430 broadway, new york, new york 10018

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 123, Specification for a Data Descriptive File for Information Interchange (DDF). For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Printed in the United States of America

A2M1286/9

ANSI/ISO 8211-1985

American National Standard
for Information Systems –

Specification for a Data Descriptive File for Information Interchange

Secretariat

Computer and Business Equipment Manufacturers Association

Approved February 28, 1986

American National Standards Institute, Inc

Foreword

(This Foreword is not part of American National Standard ANSI/ISO 8211-1985.)

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75% approval by the member bodies voting.

International Standard ISO 8211 was prepared by Technical Committee ISO/TC 97, Information Processing Systems.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

The text of ISO 8211-1985 has been adopted as the text of this American National Standard. It should be noted that certain conventions, spelling, and units in International Standards are different than those normally used in American National Standards, but are not expected to cause difficulty in understanding or use.

Suggestions for improvement of this standard will be welcome. They should be sent to the Computer and Business Equipment Manufacturers Association, 311 First Street, NW, Suite 500, Washington, DC 20001.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

Edward Lohse, Chair
Richard Gibson, Vice-Chair
Catherine A. Kachurik, Administrative Secretary

Organization Represented	*Name of Representative*
American Express	D. L. Seigal Lucille Durfee (Alt)
American Library Association	Paul Peters
American Nuclear Society	Geraldine C. Main D. R. Vondy (Alt)
AMP Incorporated	Patrick E. Lannan Edward Kelly (Alt)
Association for Computing Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals	Thomas M. Kurihara
AT&T Communications	Henry L. Marchese Richard Gibson (Alt)
AT&T Information Systems	Herbert V. Bertine Paul D. Bartoli (Alt)
Burroughs Corporation	Stanley Fenner
Control Data Corporation	Charles E. Cooper Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment	Thomas Easterday Donald Miller (Alt)
Data General Corporation	John Pilat Lyman Chapin (Alt)

Organization Represented	*Name of Representative*
Data Processing Management Association	Christian G. Meyer
Digital Equipment Computer Users Society	William Hancock
	Charles Mustain (Alt)
Digital Equipment Corporation	Gary S. Robinson
	Delbert L. Shoemaker (Alt)
Eastman Kodak	Charleton C. Bard
	Gary Haines (Alt)
General Electric Company	William R. Kruesi
	Richard W. Signor (Alt)
General Services Administration	William C. Rinehuls
	Larry L. Jackson (Alt)
GUIDE International	Frank Kirshenbaum
	Thomas F. O'Leary, Jr (Alt)
Harris Corporation	Walter G. Fredrickson
	Rajiv Sinha (Alt)
Hewlett-Packard	Donald C. Loughry
Honeywell Information Systems	Thomas J. McNamara
	David M. Taylor (Alt)
IBM Corporation	Mary Anne Gray
	Robert H. Follett (Alt)
IEEE Computer Society	Sava I. Sherr
	Thomas M. Kurihara (Alt)
	Thomas A. Varetoni (Alt)
Lawrence Berkeley Laboratory	David F. Stevens
	John S. Colonias (Alt)
Moore Business Forms	Delmer H. Oddy
National Bureau of Standards	Robert E. Rountree
	James Moulton (Alt)
National Communications System	Marshall L. Cain
	George W. White (Alt)
NCR Corporation	Thomas W. Kern
	A. Raymond Daniels (Alt)
Perkin-Elmer Corporation	Christopher Beling
	Russ Lombardo (Alt)
Prime Computer, Inc	Joseph Schmidt
	John McHugh (Alt)
Railine Corporation	R. A. Petrash
Recognition Technology Users Association	Herbert F. Schantz
	G. W. Wetzel (Alt)
SHARE, Inc	Thomas B. Steel
	Robert A. Rannie (Alt)
Sperry Corporation	Marvin W. Bass
	Jean G. Smith (Alt)
Texas Instruments, Inc	Presley Smith
	Richard F. Trow, Jr (Alt)
3M Company	J. Wade Van Valkenburg
	R. C. Smith (Alt)
Travelers Insurance Companies, Inc	Joseph T. Brophy
U.S. Department of Defense	Fred Virtue
	Belkis Leong-Hong (Alt)
VIM	Chris Tanner
	Madeline Sparks (Alt)
VISA International	Jean T. McKenna
	Susan Crawford (Alt)
Wang Laboratories, Inc	Marsha Hayek
	Joseph St. Amand (Alt)
Xerox Corporation	John L. Wheeler

Technical Committee X3L5 on Labels and File Structure, which reviewed this International Standard and submitted it to Committee X3 for consideration as an American National Standard, had the following members:

James V. Upperman, Chair
Robert L. Brandt, Vice-Chair
Herbert G. Madsen, International Representative

Sumner Blount
Alfred A. Brooks
G. William Cox
Richard Desjardins
Robert H. Follett
T. N. Hastings
D. Hekimi
W. Loehner
Howard Kaikow
Thomas W. Kern
Bruce Schafer
Cal C. Waller
Theodore C. Woo
R. Hedberg (Alt)
Jean G. Smith (Alt)

Contents

American National Standard for Information Systems –

Specification for a Data Descriptive File for Information Interchange

0 Introduction

This International Standard has been produced in response to an identified need for a mechanism to allow data structures to be easily moved from one computer system to another, independent of make. Data structures required to be interchanged can vary significantly in complexity and size, and a common method to accomplish these interchanges is desirable. It is also desirable that any medium such as a communication line, a magnetic tape, a disk pack, a flexible disk, etc., should be able to be used for the physical interchange and that, if possible, all information necessary to successfully recreate the structure in the target system should be contained within the information transported on the medium.

To meet these needs this International Standard specifies medium-independent and system-independent file and data record formats for the interchange of information between computer systems. This International Standard is intended for use with physical recorded media as well as with communications media. The contents in the user data structure can be of any internationally recognized character set and coding and are interchanged in a transparent fashion. The intermediate structure through which the information passes is designed for interchange purposes only and is not intended to be used for general processing.

The aim when developing this International Standard was to define an interchange format into which the sender's information is mapped and is conveyed to the receiver's system. Upon receipt of the information in the interchange format it is then mapped into the receiver's format without loss of structure and content. This International Standard specifies a method for describing a robust interchange structure which can accept most user data structures. The method enables the sender to preserve structure information and to convey it to the receiver with the data so that the receiver can remap the structure and data into the local system.

Most data structures in common use can be described and interchanged using this International Standard. The structures within the interchange file can be of the following forms : elementary data, vectors, arrays and hierarchies. User file structures such as sequential, hierarchical, relational and indices can be converted into the interchange structure. Network structures can be interchanged but additional pre-processing and post-processing is necessary to preserve logical linkages.

This International Standard is medium-independent and requires an environment in which International Standard labels and file structures can be written on or read from the standard media chosen. It is assumed that variable-length records can be processed by the supporting label and file processing system. It requires a computer process capability to map the user file or data base management system to the interchange file. This mapping function has to provide the necessary data and structure conversions. The parameters required to define the selection and conversion of these data items and structures into the formats specified by this International Standard are outside the scope of this International Standard. The interchange standard requires the use of the ISO 646 coded character set in control fields and permits the use of extended character sets in user data fields.

This International Standard provides for three interchange levels from which the users may choose based on the complexity of their data structures. The first interchange level supports multiple fields containing simple, unstructured character strings. The second level supports level one and processes multiple fields containing structured user data comprising a variety of data types. The third level supports level two and hierarchical data structures.

NOTE — Additional information concerning the application of this International Standard is given in annex A.

1 Scope and field of application

This International Standard specifies an interchange format to facilitate the transfer of files containing data records between computer systems. It is not designed as a record format for indigenous files of any specific system. It defines a generalized structure which can be used to transmit, between systems, records containing a wide variety of data types and structures. It provides the means for the description of the contents of data records but does not define their contents.

This International Standard specifies

a) media-independent file and data record descriptions for information interchange. It assumes the use of other International Standards for labelling and file structure such as ISO 1001, ISO 4341, ISO 7665;

b) the description of data elements, vectors, arrays and hierarchies containing character strings, bit strings and numeric forms. The numeric forms are specified by ISO 6093;

c) a data descriptive file comprising a data descriptive record and companion data records that enable information interchange to occur with minimal specific external description;

d) the data descriptive record that describes the characteristics of each data field within the companion data records;

e) three levels of interchange depending on the complexity of the allowed structure (see 5.2.1.2).

2 Conformance

Interchange files shall be in conformance with this International Standard when all of the data descriptive records and data records conform to the specifications of this International Standard. A statement of conformance shall specify the interchange level to which the contents of files conform.

This International Standard does not specify requirements for processing and implementation, therefore this processing cannot itself conform to this International Standard.

3 References

ISO 646, *Information processing — ISO 7-bit coded character set for information interchange.*

ISO 1001, *Information processing — Magnetic tape labelling and file structure for information interchange.*

ISO 2022, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques.*

ISO 4341, *Information processing — Magnetic tape cassette and cartridge labelling and file structure for information interchange.*

ISO 6093, *Information processing — Representation of numerical values in character strings for information interchange.*[1)]

ISO 7665, *Information processing — File structure and labelling of flexible disk cartridges for information interchange.*

The following document is also relevant to this International Standard :

ISO international register of character sets to be used with escape sequences.

4 Definitions

For the purposes of this International Standard the following definitions apply.

4.1 alphanumeric character : A character occurring in columns 2 to 7 inclusive [except position (7/15)] of the International Reference Version of ISO 646.

1) At present at the stage of draft.

NOTE — The characters specified in this International Standard are represented by their position (column/row) in the coded character set table of ISO 646, or by their acronyms, i.e., ESC, RS, US, and SP.

4.2 array descriptor : A sequence of numbers which specifies the number of dimensions and extent of an array.

4.3 base address of data : A data element the value of which is equal to the byte count up to the first data field following the field terminator of the directory, where the specified origin (0) is the first byte of the leader.

4.4 bit field : A data field comprising only binary digits and, when necessary, filled on the right with binary zeros to complete a byte. See also character mode bit string.

4.5 byte : A collection of n bits.

NOTE — This International Standard is media-independent and the number of bits will be media-dependent.

4.6 Cartesian label : An array of identifiers formed by the Cartesian product of the elements of two (or more) vector labels. The array elements have the same order as the elements of the direct product such that if $\vec{a}$ and $\vec{b}$ are the vector labels, $\vec{a} = [a(1), \ldots, a(n)]$ and $\vec{b} = [b(1), \ldots, b(m)]$, then the Cartesian label, $\vec{a}\cdot\vec{b} = [a(1)b(1),\ a(1)b(2),\ \ldots,\ a(1)b(m),\ \ldots,\ a(n)b(m)]$ where $a(i)b(j)$ is a concatenation of $a(i)$ and $b(j)$ which forms an identifier of the i,j element of a corresponding data array.

4.7 character mode bit string : A sequence of characters (0 or 1) that represents a string of binary digits. (See also bit field.)

4.8 compound data field : A field comprising one or more elementary data elements.

4.9 data descriptive file; DDF : A file containing a data descriptive record and its companion data records.

4.10 data descriptive record; DDR : A record that logically precedes the data records and contains the control parameters and data definitions necessary to interpret companion data records. The data descriptive record is the first logical record of a file other than the file labels (if applicable).

4.11 data record; DR : A logical record containing user data.

4.12 delimited structure : A structure composed of a collection of data elements that are separated by delimiters.

4.13 delimiter : A single character that separates data elements and data fields. (See table 1 for the use of delimiters.)

4.14 directory : A table of identifiers and references to corresponding items of data.

4.15 directory entry : A fixed-length field within the directory that contains information about the tag, the length and the location of a specific field for a given record.

4.16 elementary : Having the property of being indivisible without loss of meaning.

4.17 entry map : A field in the leader that is used to indicate the structure of the entries in the directory.

4.18 escape character; ESC : A control character which is used to provide additional characters. It alters the meaning of a limited number of contiguously following bit combinations.

The use of this character is specified in ISO 2022.

4.19 field terminator; FT : A character used to terminate a field within a record, (1/14) in ISO 646.

4.20 file : A collection of related records treated as a unit.

4.21 file title : A string of characters that provides a displayable descriptive title for the interchange file.

This need not be the same as the file name.

4.22 hierarchy; hierarchical structure : A rooted, ordered tree structure comprising a superior root node with successive multiple ordered subtrees at increasingly inferior nodes, ultimately terminating in leaf nodes.

4.23 interchange level; level : The designation of a prescribed subset of the requirements of this International Standard.

4.24 interchange format : A format for the exchange, as opposed to the local processing, of records.

4.25 label : A character string used to identify or name a field or subfield and its contents.

4.26 leader : A fixed-length field that occurs at the beginning of each record and provides parameters for the processing of the record.

4.27 location : The byte count to the position of the first byte of a field.

Locations in the leader and directory are relative to the first (0) byte of the leader, and the location for data descriptive fields and user data fields are relative to the base address of data.

4.28 logical record; record : A collection of related data elements independent of their representation on a medium.

4.29 to map : To establish the correspondence between the elements of two structures.

4.30 null : Pertaining to the condition of non-occurrence of an entity, usually a data element, string or set.

4.31 preorder traversal sequence : A sequence of the nodes of a hierarchy produced by the following recursive algorithm :

a) enter the tree at the root node;

b) traverse the left-most subtree not previously traversed;

c) if b) is not possible, return to the node superior to the subtree and go to b).

4.32 record length : A data element the value of which is equal to the length in bytes of the record.

4.33 relative position; RP : The position of a byte expressed as a decimal integer relative to the beginning of a field.

The first relative position is numbered "0".

4.34 tag : An identifier in a directory entry used to specify the internal name of an associated field.

4.35 unit terminator; UT : A character used to delimit several types of subfields within variable-length fields in both DDR and DR, (1/15) in ISO 646.

4.36 variable-length field : A field the length of which varies from occurrence to occurrence.

4.37 vector label : A vector the elements of which are labels (i.e., "column" headings or "row" headings) used to identify each element in a vector of data elements.

5 Interchange file

5.1 General structure

This subclause specifies the general structure of the interchange file and subsequent subclauses provide the detailed specifications. Figure 1 shows a schematic representation of a file and the file labels.

ISO standard file labels	
Data Descriptive File :	Data Descriptive Record Data Records
ISO file termination indicator	

Figure 1 — File and file label schematic representation

This International Standard specifies multiple Data Descriptive Files (DDFs), each comprising logical records with the required ISO interchange file labels or headers for the particular medium. Each file shall consist of the following logical records :

a) the Data Descriptive Record (DDR), and

b) the Data Records (DRs).

The overall structure shall be as shown in figure 2, which is an expanded logical schematic representation of the DDR and DR, the leaders and directories of each record, typical records and a typical data field for each record. The DDR and DR records shall have the same leader, directory, field and record structure although their contents vary. A provision is made to omit the repetition of identical DR leaders and directories for the interchange for repetitive, fixed-format data.

NOTES

1 The logical juxtaposition of fields is shown and the meaning of some pointers and field lengths is indicated. For a physically sequential medium, figure 2 represents its physical order.

2 The special field tags specified in this International Standard are described in the following format : 0...n where "n" is a decimal number and "0..." implies sufficient zeros on the left to fill the tag field.

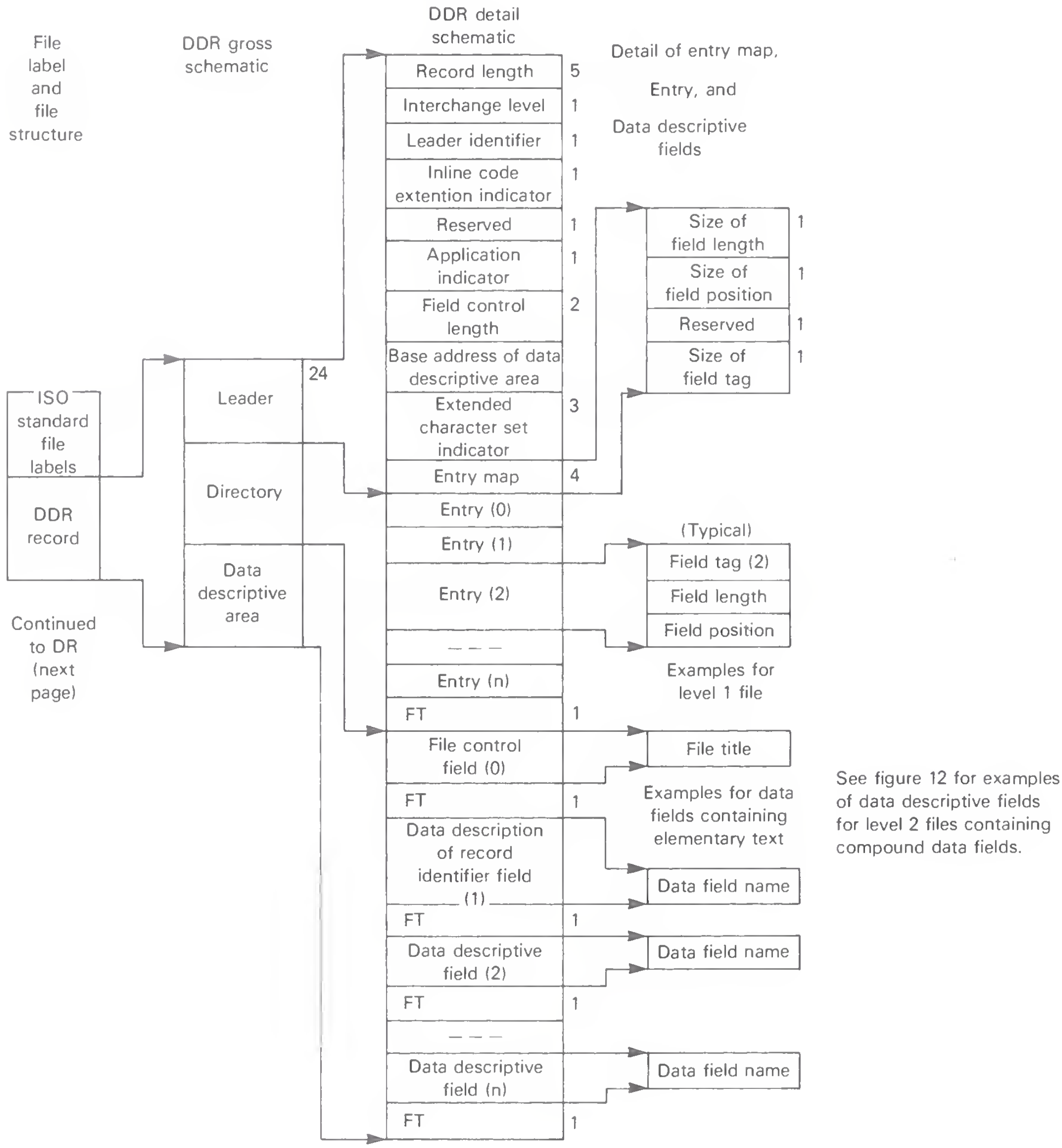
Each data descriptive field of the DDR contains a data description of the user data in a DR user data field having the same tag. The DDR (but not the DR) has a special tag 0...0 and a corresponding field which contains field controls, an optional file title and in the case of a hierarchy, structure information. The DRs have a special field to identify a record and the DDR contains a description of this field in the data descriptive field having the same tag (0...1). The contents of a DDR variable-length field vary depending upon the values of parameters in the DDR leader.

The DDR data descriptive fields shown in figure 2 are for elementary character data fields and those shown in figure 12 are for compound data fields with all optional subfields included.

NOTES

1 The contents of a DR user data field may be highly varied, depending upon its description in the DDR; no example is given in figures 2 and 12. See annex B for examples of data fields.

2 Throughout the remainder of this International Standard the length of fields, except for bit fields, is given in bytes whose length in bits may be media-dependent. The contents of fields are referred to as characters, and multiple byte character sets are permitted in user fields (see 7.1.4). Therefore in these cases the field length is not equal to the number of characters.



NOTES

1 Separator characters
FT = Field Terminator

2 Fixed field lengths are given in bytes to the right of each field.

3 Tags, their corresponding entries and fields are designated by (i).

Figure 2 — Expanded logical schematic representation of data descriptive file

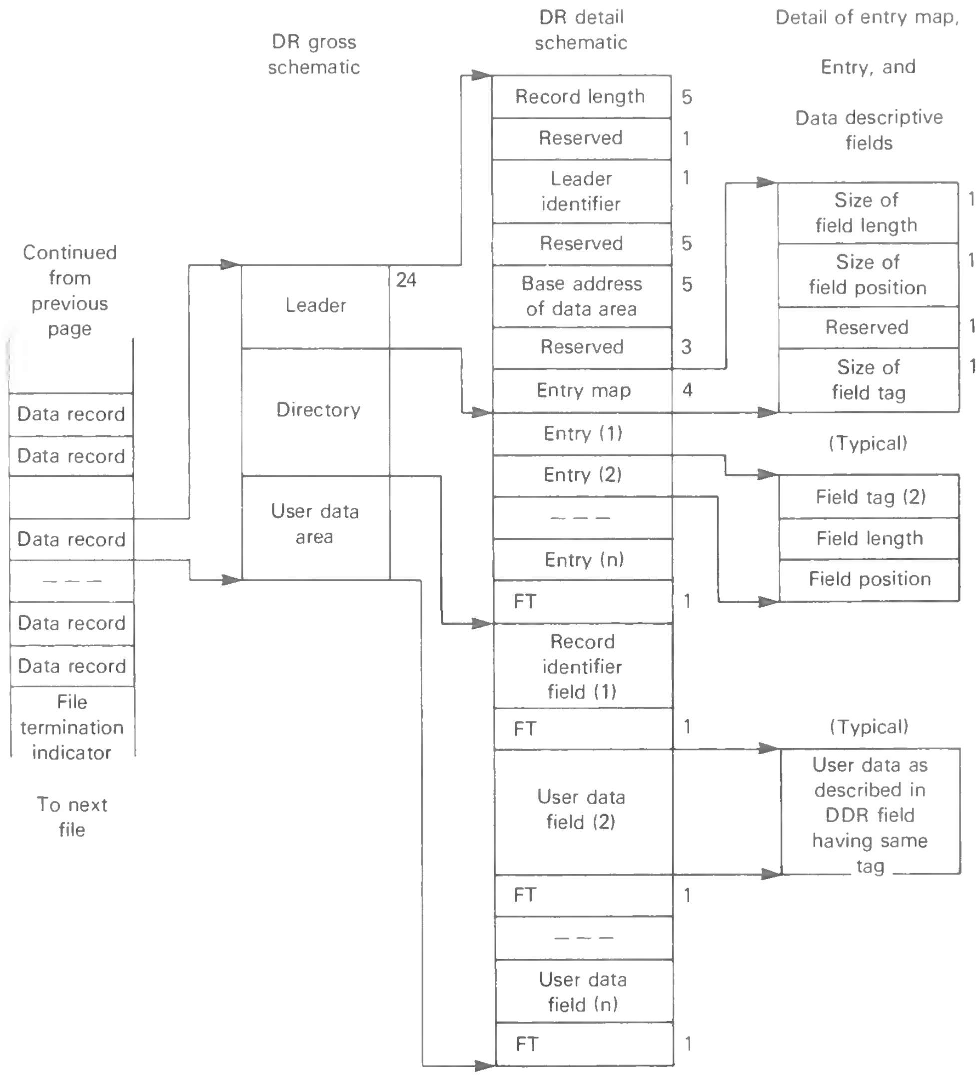


Figure 2 — Expanded logical schematic representation of data descriptive file (*concluded*)

5.2 Data Descriptive Record (DDR)

The DDR shall consist of the areas and terminators shown in figure 3 and shall be the first record of the file.

Name of area	Length
Leader	24
Directory	$k \times p$
Field terminator	1
Data descriptive area	variable
Field terminator	1

Figure 3 — DDR schematic

Each of the logical records shall consist of

a) a leader of 24 characters;

b) a directory of length $k \times p$, terminated by a field terminator, (1/14), where k is the number of directory entries and p is the length of each entry (see 5.2.2); and

c) a set of k variable-length fields, each terminated by a field terminator, (1/14).

5.2.1 DDR leader

The DDR leader shall consist of the fields shown in figure 4 and is further specified in 5.2.1.1 to 5.2.1.9.

RP	Name of field	Length	Contents
0	Record length	5	digits
5	Interchange level	1	digit
6	Leader identifier	1	character
7	Inline code extension indicator	1	character
8	Reserved[1)]	1	SPACE character
9	Application indicator	1	character
10	Field control length	2	digits
12	Base address of data	5	digits
17	Code character set indicator	3	characters
20	Entry map	4	digits

1) Reserved for future standardization.

Figure 4 — DDR leader schematic

5.2.1.1 Record length field (DDR RP 0 to 4)

This field shall specify the total length of the DDR in bytes. The contents of this field shall be digits. A DDR length of 0 will signify a length in excess of 99 999.

5.2.1.2 Interchange level field (DDR RP 5)

This field shall specify the level of the interchange file. The content of this field shall be the digit 1, 2 or 3.

The value

1 shall mean that the file conforms to a level 1 file;

2 shall mean that the file conforms to a level 2 file;

3 shall mean that the file conforms to a level 3 file.

A level 1 file shall contain elementary character data fields (see 6.1) but not compound data fields nor hierarchical structures. A level 2 file shall contain compound data fields (see 6.2) but not hierarchical structures. A level 3 file shall contain compound data fields and a list of the tag pairs (see 5.2.3.1.3) describing the hierarchical structures.

5.2.1.3 Leader identifier field (DDR RP 6)

This field shall specify that this record is the DDR and shall contain the character "L".

5.2.1.4 Inline code extension indicator (DDR RP 7)

This field shall indicate if inline escape sequences are used in data fields to designate extended coded character sets as specified in ISO 2022.

The value

SP shall mean that no extensions are used;

E shall mean that extensions are used.

5.2.1.5 Reserved for future standardization (DDR RP 8)

This field is reserved for future standardization.

5.2.1.6 Application indicator field (DDR RP 9)

This field is reserved for future standardization and shall contain the character SPACE.

5.2.1.7 Field control length field (DDR RP 10 and 11)

This field shall specify the number of bytes of the data descriptive field devoted to data element type and structure codes, delimiters, and other positions reserved for future standardization (see 5.2.3.1.1).

The contents of this field shall be the digits 00, 03, 06 or 09 (see 5.3.3.2 and 7.1.3).

5.2.1.8 Base address of data descriptive area (DDR RP 12 to 16)

This field shall specify the position of the first data descriptive field of a DDR.

NOTE — The first data descriptive field will be the file control field or the record identifier field.

The contents of this field shall be digits and shall be equal to the combined length in bytes of the leader and directory including the field terminator at the end of the directory.

5.2.1.9 Extended character set indicator field (DDR RP 17 to 19)

This field shall specify the use of default coded character set extensions in the file.

The values in this field shall have the following meanings :

a) (2/0)(2/0)(2/0) : Only the International Reference Version of ISO 646 character set has been designated as the default for the file.

b) (2/0)(2/1)(2/0) : Extended character sets have been designated as the default for one or more data fields (see 7.1.1).

c) Truncated escape sequence : An extended character set has been designated as the default for the entire file (see 7.1.2).

5.2.1.10 Entry map field (DDR RP 20 to 23)

This field specifies the lengths of directory entry subfields and shall consist of the subfields shown in figure 5 (see 5.2.2).

Each subfield of this field shall contain a single digit.

RP	Name of field	Length	Contents
20	Size of field length	1	digit
21	Size of field position	1	digit
22	Reserved for future standardization	1	digit
23	Size of field tag	1	digit

Figure 5 — DDR entry map schematic

5.2.1.10.1 Size of field length field (DDR RP 20)

This field shall specify the size in bytes of the field length subfield of the directory entries and shall be a digit in the range of "1" to "9" inclusive.

5.2.1.10.2 Size of field position field (DDR RP 21)

This field shall specify the size in bytes of the field position subfield of the directory entries and shall be a digit in the range of "1" to "9" inclusive.

5.2.1.10.3 Reserved for future standardization (DDR RP 22)

This field is reserved for future standardization as an extended entry map and shall be the digit "0".

5.2.1.10.4 Size of field tag field (DDR RP 23)

This field shall specify the size in bytes of the field tag subfield of the directory entries and shall be a digit in the range of "1" to "7" inclusive.

NOTE — For the purposes of 5.2 the following nomenclature is used :

m is the size of the field length subfield;

n is the size of the field position subfield;

t is the size of the field tag subfield.

5.2.2 DDR directory

The DDR directory shall consist of repeated DDR directory entries, the subfield lengths of which shall be specified in the entry map. The DDR directory shall contain one DDR directory entry for each data descriptive field and shall end with a field terminator, (1/14). The DDR shall define all DR tags.

The DDR directory entry specifies the location and length of a corresponding data descriptive field and shall consist of the subfields shown in figure 6. Each entry shall contain a field tag, field length and field position, in that sequence and shall consist of $m + n + t$ bytes.

The DDR directory entries shall be in one to one correspondence with the data descriptive fields. For hierarchical data structures (see 5.2.3.1.3), the directory entries of the DDR shall be in the same order as the pre-order traversal sequence of the generic data tree.

RP	Name of field	Length	Contents
$p(i-1)$	Field tag	t	alphanumeric
$p(i-1)+t$	Field length	m	digits
$p(i-1)+t+m$	Field position	n	digits

where

$p = t + m + n$ and;

i = the index of the directory entry.

Figure 6 — DDR directory entry schematic

5.2.2.1 DDR field tag field

This field shall contain a field tag identifying a data descriptive field and shall consist of between one and seven alphanumeric characters. The same field tag shall occur only once within the DDR.

5.2.2.1.1 Field tag 0...0 shall identify the optional file control field and if present shall occur only in the DDR.

5.2.2.1.2 Field tag 0...1 shall occur once in each record and shall identify the record identifier field.

5.2.2.1.3 If used, field tag 0...2 shall identify an optional user field of the DDR which has no corresponding data field in the DR. An implementation shall pass this field to the user for processing.

NOTE — The contents of this field should be determined by the user and may be used to convey any augmented file description (for example file attributes relevant to the interchange) or ancillary file processing controls or application information.

5.2.2.1.4 Field tags 0...3 through 0...9 shall be reserved for future standardization.

5.2.2.1.5 When present, tags 0...0 to 0...9 inclusive shall occur first in the DDR directory and in ascending numeric order.

5.2.2.2 DDR field length field

This field specifies the length in bytes of the field to which it corresponds. This field shall contain a right-justified integer filled with leading zeros. The length of the field shall include the field terminator.

5.2.2.3 DDR field position field

This field specifies the relative position of the first byte in the field referenced by the entry. This field shall contain a right-justified integer filled with leading zeros. It shall be relative to the base address of the data descriptive area as specified in DDR RP 12 to 16. The first byte of the first field following the directory shall be numbered 0.

5.2.3 Data descriptive area

The data descriptive area shall contain in its data fields that information which defines and describes the corresponding (i.e., having the same tag) data fields of the DR and provides control parameters for automated processing. These fields shall have a specified format determined only by the DDR leader contents. All fields shall end with a field terminator, (1/14).

5.2.3.1 File control field (tag = 0...0)

The file control field shall contain the following subfields :

a) the field controls (if any);

b) an optional file title, and

c) a list of ordered tag pairs (only for hierarchical records) (see figure 7).

This field shall be terminated by the field terminator, (1/14).

Field controls	File title	U T	List of tag pairs	F T

Figure 7 — File control field schematic

5.2.3.1.1 Field control field

This field shall be present only in level 2 and level 3 files and its length is specified by the field control length (DDR RP 10 and 11) (see 5.2.1.7). These controls shall not be used in the title control field and shall have the value 0 or spaces (see 6.2.2).

5.2.3.1.2 File title field

This field shall specify an optional file title following the field controls. It shall contain an optional character string which shall be an external descriptive name of the interchange file.

5.2.3.1.3 List of tag pairs

The list of tag pairs shall describe the hierarchical structure. It shall follow the file title field and shall be preceded by a unit terminator, (1/15). The pairing of the tags shall be determined by the structural order from root node to leaf node in the generic data structure. These pairs may be placed in the list in any sequence and shall be contiguous. The root tag shall be the tag of the record identifier field, 0...1. Tags 0...2 to 0...9 inclusive shall not participate in the structure specification.

NOTE — The variable hierarchical data structures permitted in the DRs are described by supplying the pre-order traversal sequences of the data tree and a list of the tag pairs expressing the superior/subordinate logical association between the nodes of the data tree. The data structures of the DRs shall be derived from the generic data tree by the repetition or deletion of nodes and subtrees. The pre-order traversal sequence of the generic data tree is supplied by the order of the DDR directory entries (see 5.2.2). The pre-order traversal sequence of each instance of a derived data tree in a DR is supplied by the sequence of directory entries in the DR (see 5.3.2). See annex C for further explanation of ordered tag pairs.

5.2.3.2 Description of record identifier field (tag = 0...1)

The data descriptive area shall contain a data descriptive field describing the record identifier field of the DR. This data descriptive field shall comply with the requirements for descriptive fields which describe user data fields as given in 5.3.3 and clause 6.

5.2.3.3 Description of user data fields

The data descriptive area shall contain a data descriptive field for each of the user data fields. These data descriptive fields are specified in clause 6.

5.3 Data record (DR)

The data records shall consist of the areas and terminators shown in figure 8.

Name of area	Length
Leader	24
Directory	$k' \times p'$
Field terminator	1
User data area	variable
Field terminator	1

where k' is the number of directory entries and p' is the number of bytes in a directory entry of the DR.

Figure 8 — DR schematic

For an interchange file consisting of variable length records followed by fixed-length records comprising only fixed-length data fields in which the DRs have identical leader and directory values, the leader and directory of the first DR having a leader and directory identical to the remaining DRs shall apply to all subsequent DRs. The leader and directory of the subsequent DRs may be omitted.

5.3.1 DR leader

Each DR leader shall consist of the fields shown in figure 9 and are further specified in 5.3.1.1 to 5.3.1.7.

RP	Name of field	Length	Contents
0	Record length	5	digits
5	Reserved for future standardization	1	SPACE character
6	Leader identifier	1	character
7	Reserved for future standardization	5	SPACE characters
12	Base address of data	5	digits
17	Reserved for future standardization	3	SPACE characters
20	Entry map	4	digits

Figure 9 — DR leader schematic

5.3.1.1 Record length field (DR RP 0 to 4)

This field shall specify the total length of the DR in bytes. The contents of this field shall be digits. A DR length of 0 will signify a length in excess of 99 999.

5.3.1.2 Reserved for future standardization (DR RP 5)

This field is reserved for future standardization.

5.3.1.3 DR leader identifier field (DR RP 6)

This field shall specify that the record is a DR and shall specify the repetitive characteristics of the subsequent DR leaders and directories. The values in this field shall have the following meanings :

D means that a leader and directory will be found in the subsequent DR;

R means that a leader and directory will not be found in the DRs subsequent to the current DR and that the leader and directory of the current DR shall be applied to each subsequent DR.

5.3.1.4 Reserved for future standardization (DR RP 7 to 11)

This field is reserved for future standardization.

5.3.1.5 Base address of user data area (DR RP 12 to 16)

This field shall specify the position of the first user data field of a DR record.

NOTE — The first user data field will be the record identifier field.

5.3.1.6 Reserved for future standardization (DR RP 17 to 19)

This field is reserved for future standardization.

5.3.1.7 Entry map field (DR RP 20 to 23)

This field specifies the lengths of directory entry subfields within each DR and shall consist of the subfields shown in figure 10.

Each subfield of this field shall contain a single digit.

RP	Name of field	Length	Contents
20	Size of field length	1	digit
21	Size of field position	1	digit
22	Reserved for future standardization	1	digit
23	Size of field tag	1	digit

Figure 10 — DR entry map schematic

5.3.1.7.1 Size of field length field (DR RP 20)

This field shall specify the size in bytes of the field length subfield of the directory entries and shall be in the range of "1" to "9" inclusive.

5.3.1.7.2 Size of field position field (DR RP 21)

This field shall specify the size in bytes of the field position subfield of the directory entries and shall be in the range of "1" to "9" inclusive.

5.3.1.7.3 Reserved for future standardization (DR RP 22)

This field is reserved for future standardization as an extended entry map. Its value shall be the digit "0".

5.3.1.7.4 Size of field tag field (DR RP 23)

This field shall specify the size in bytes of the field tag subfield of the directory entries and shall be in the range of "1" to "7" inclusive. The value in this field shall be equal to the value of the DDR size of field tag field.

NOTE — For the purposes of 5.3 the following nomenclature is used :

m' is the size of the field length subfield;

n' is the size of the field position subfield;

t is the size of the field tag subfield.

5.3.2 DR directory

The DR directory shall consist of repeated DR directory entries, the subfield lengths of which shall be specified in the entry map. The DR directory shall contain one DR directory entry for each user data field and shall end with a field terminator, (1/14). All DR tags shall be defined in the DDR.

The DR directory entry specifies the location and length of a corresponding user data field and shall consist of the subfields shown in figure 11. Each entry shall contain a field tag, field length and field position, in that sequence and shall consist of $m' + n' + t$ bytes.

The DR directory entries shall be in one to one correspondence with the user data fields. Except for tag 0...1, a tag may be repeated in a DR directory. For non-hierarchical structures, repeated tags shall be logically contiguous in a DR directory and the DR tags shall occur in the same order as the tags of the DDR. The tags corresponding to missing user data fields may be omitted from the directory unless required to describe structure. For hierarchical data structures (see 5.2.3.1.3), the directory entries of the DR shall be in the same order as the preorder traversal sequence of the corresponding data fields of the derived data tree of that DR.

RP	Name of field	Length	Contents
$p(i-1)$	Field tag	t	alphanumeric
$p(i-1)+t$	Field length	m'	digits
$p(i-1)+t+m'$	Field position	n'	digits

where

$p = t + m' + n'$ and;

i = index of the directory entry.

Figure 11 — DR directory entry schematic

5.3.2.1 DR field tag field

This field shall contain a field tag identifying a data field and shall consist of between one and seven alphanumeric characters.

Field tag 0...1 shall occur once in each DR, shall identify the record identifier field and shall occur first in the DR directory.

5.3.2.2 DR field length field

This field specifies the length in bytes of the field to which it corresponds. This field shall contain a right-justified integer filled with leading zeros. The length of the field shall include the field terminator.

5.3.2.3 DR field position field

This field specifies the relative position of the first byte in the field referenced by the entry. This field shall contain a right-justified integer filled with leading zeros. It shall be relative to the base address of the data descriptive area as specified in DR RP 12 to 16. The first byte of the first field following the directory shall be numbered 0.

5.3.3 User data area

The data fields of the user data area shall contain the user information to be interchanged. Each data field shall be an instance of the user data structure and data types defined by the DDR data descriptive field with the corresponding field tag. The subfields shall contain the data elements corresponding to the labels contained in the corresponding DDR subfields. Each field shall end with the field terminator, (1/14). In a delimited structure, where the delimiter character of subfields is (1/15), missing data elements shall be represented by successive delimiters. A terminal string of adjacent subfield delimiters may be replaced by a field terminator, (1/14). Elementary character data fields whose termination is determined by a field length, format width, user determined delimiter or field terminator may contain a unit separator character which shall be treated by an implementation as a user data character.

The description of the user data is further defined in clause 6.

5.3.3.1 Record identifier field (tag = 0...1)

Each DR shall contain exactly one record identifier field. The content of the DR record identifier field shall conform to the corresponding DDR data descriptive field and it shall be unique within the file. Each identifier field shall be left-justified and filled on the right by spaces (2/0) if the field is alphanumeric, and right-justified and filled on the left by zeros (3/0) if numeric.

NOTE — See clause A.3 to annex A for further discussion of unique identifiers.

5.3.3.2 User data fields

5.3.3.2.1 Elementary character data fields

The user data fields of the DR shall contain a single string of characters terminated by the appropriate field terminator (see table 1).

Interchange files composed solely of user data fields of this structure and type shall have the following control characters in their DDR leader :

Leader field	RP	Characters
Application indicator	9	SP
Field control length	10 and 11	00

The remaining DDR leader fields are specified in 5.2.1.

5.3.3.2.2 Compound data fields

The data in these fields shall be characters, delimiters and bit strings which conform to the specifications given in the corresponding data descriptive field (see 6.2).

Interchange files composed solely of fields of these structures shall have the following control characters in their DDR leader :

Leader field	RP	Characters
Application indicator	9	SP
Field control length	10 and 11	06

The remaining DDR leader fields are specified in 5.2.1.

6 Description of user data types and structures

6.1 Description of elementary character data fields

This subclause specifies the method for the description of user data comprising elementary character strings.

Data descriptive field : The data descriptive field for elementary data fields shall contain only an elementary character string constituting the name of the user data field (see figure 2).

6.2 Description of compound data fields

This subclause specifies the method for the description of compound structures and data types which are more complex than those covered by 6.1.

The data structures specified in this subclause shall be elementary, vector and array structures containing character strings, implicit point, explicit point, explicit point scaled, character mode bit string, bit field, and mixed data types. The data descriptive field shall contain control information, delimiters, field name, subfield labels for vectors and arrays, and data field format information as specified in 6.2.1 to 6.2.4 and 7.1.

6.2.1 Data descriptive fields

The data descriptive fields for compound data fields shall comprise the field control field, the data field name, data element labels and data formats. These are shown schematically in figure 12. The field controls, names, labels, and formats of the permitted data structures and types are given in tables 1 and 2 and further specified in this clause. Table 1 specifies the use of delimiters. Table 2 specifies the format of the data descriptive field. The allowable data types described by the data descriptive field are

Type	Contents
character	character strings
implicit point	implicit point numeric representations
explicit point	unscaled explicit point numeric representations
explicit point scaled	scaled explicit point numeric representations
character mode bit string	digits 0 and 1
bit field	binary digits
mixed types	one or more of the above data types

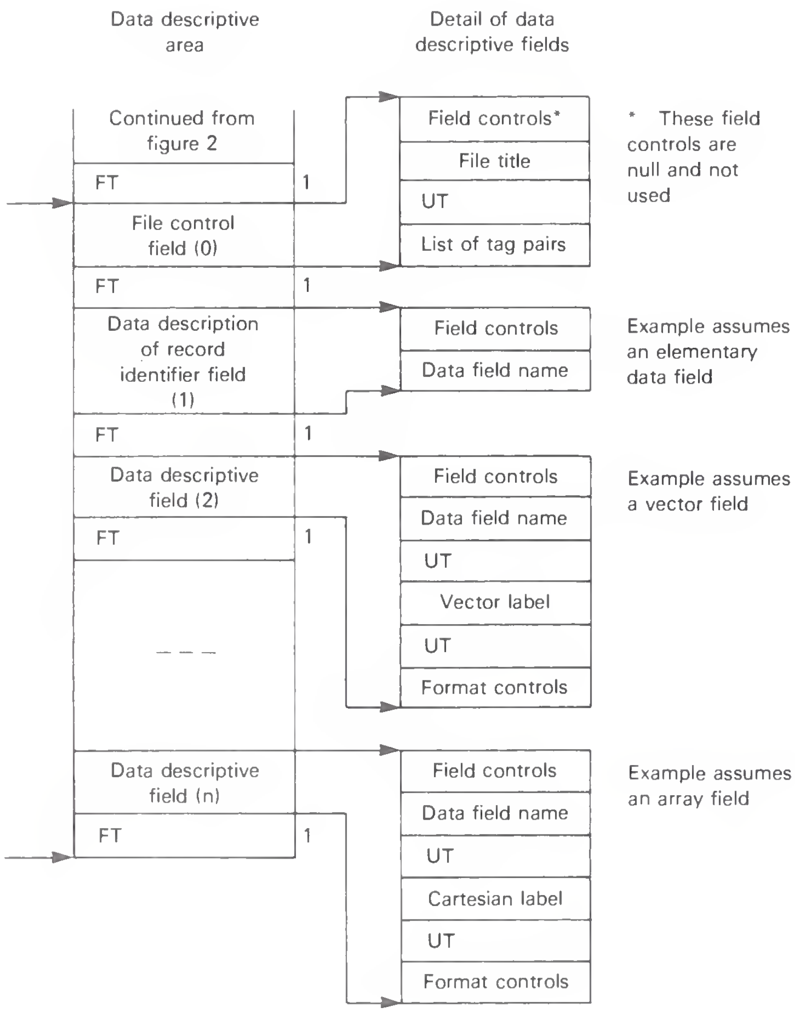


Figure 12 — Level 2 data descriptive area schematic representation

Table 1 — Delimiters and their uses

Information separator	Print symbol	Record	Use
(RS) 1/14 [1]	;	DDR, DR	Field terminator (FT)
(US) 1/15 [1]	&		Unit terminator (UT)
		DR	a) to delimit subfields in a data field not specified by a format
		DDR	b) to pre- and post-delimit an optional field name and a vector label
		DDR	c) to pre-delimit the hierarchy controls in field 0...0
		DDR	d) to pre-delimit format controls
(!) 2/1	!	DDR	to delimit data element labels within a vector label
(*) 2/10	*	DDR	to delimit the vector labels in a Cartesian label

1) The graphics for these ISO 646 information separator characters are not printable on some printing devices. The symbols given in the next column are substituted for the standard graphic representations of the separators throughout the manuscript for readability.

Table 2 — Compound data field descriptors

	Field controls		
Data structure/Data type	Relative position 0 1 23	45	Name/label/format controls[1]
Elementary/			
Character	0 0 00		
Implicit point	0 1 00		
Explicit point	0 2 00		['Name'];
Explicit point, scaled	0 3 00		
Character mode bit string	0 4 00		
Bit field	0 5 00		['Name']&'fctrl';
Vector/			
Character	1 0 00		
Implicit point	1 1 00	[;&]	
Explicit point	1 2 00		['Name']&['Vector label']&['fctrl'];
Explicit point, scaled	1 3 00		
Character mode bit string	1 4 00		
Bit field	1 5 00		['Name']&['Vector label']&'fctrl';
Mixed types	1 6 00		
Array/			
Character	2 0 00		
Implicit point	2 1 00		
Explicit point	2 2 00		['Name']&['Cartesian label']&['fctrl'];
Explicit point, scaled	2 3 00		['Array descriptor']
Character mode bit string	2 4 00		
Bit field	2 5 00		['Name']&['Cartesian label']&'fctrl';
Mixed types	2 6 00		['Array descriptor']

1) The description may be terminated after any subfield by (1/14). The user graphics shown are those used in this International Standard to represent the non-printable information separators. The user-defined printable graphic symbols chosen by the user to represent the non-printable information separators may be any set of displayable characters not in conflict with the data.

The syntax of table 2 is defined as follows :

a) [ ] specifies optional presence of the field;

b) 'Name' specifies a character string which shall identify the entire field;

c) 'Vector label' specifies a set of elementary labels which shall correspond to the set of subfields in the data records and shall take the form, label1!label2!... ;

d) 'Cartesian label' specifies a set of labels which shall comprise vector labels forming a Cartesian product having an order corresponding to the set of subfields in the data records and shall take the form :

label1!label2!...*labela!labelb!...*... ;

e) 'fctrl' specifies a format control which shall consist of a string of characters defining the format of the data field.

To prevent ambiguity, an incomplete data description shall require the presence of adjacent delimiters to indicate the absence of a name, label or format.

6.2.2 Field controls

The field controls, in positions 0 to 5, shall consist of four numeric characters followed by two user-defined printable graphics symbols to represent the two system delimiters, 1/14, 1/15 in that order. Relative positions 0 and 1 shall be interpreted as a structure-type code and relative positions 2 and 3 shall be reserved for future standardization. The value 00 in RP 2 and 3 shall mean that no auxiliary control is specified.

NOTE — The two printable graphics may be chosen by the user from characters which do not occur in the data fields. They shall not be substituted for the ISO information separators (see table 1) in the data fields but are optionally provided to facilitate displaying by the recipient and the sender. The space character (2/0) should be used as a default where no optional character is specified.

6.2.3 Names, labels and format controls

The use of names, labels and format controls are specified below :

6.2.3.1 Name

A name is an optional title for the data field and its contents. National variant characters or characters from the default set as specified in 7.1 are permitted in names.

6.2.3.2 Label

A label is an optional title for elementary data fields. Where the elementary data fields form a vector, the label shall be in the form of a corresponding vector label. Where the elementary data fields form an array, the label shall be in the form of a corresponding Cartesian label which, when expanded by the defining conventions, forms an array of labels corresponding to the data array. The vector labels of a Cartesian label provide row and column headings for the appropriate cross-section of the array.

The first vector label of a Cartesian label may be null, permitting the description of a two- (or higher) dimensional array without "row" names. A null first vector label shall be indicated by the adjacent delimiters "(1/15) (2/10)". The use of this construction requires a format which shall describe the array in the form of a row. The use of null individual labels in label vectors is permitted provided all delimiters are present (see 6.2.4).

National variant characters or characters from the default set as specified in 7.1 are permitted in labels. The symbols "!" and "*" are specific graphic characters used to delimit vector labels; therefore, "!" or "*" cannot appear in the name of a data element in a vector label subfield.

6.2.3.3 Format controls

The format controls specify the character-by-character or bit-by-bit structure of the data field. The format controls are mandatory for bit field data type and mixed data types and optional for all other data types, for which the lack of format controls indicates delimited data. The format controls are required to specify the sequence and type of the subfields in a mixed data field or to specify the field width in nondelimited fields or to specify the user delimiters.

The format controls shall be delimited by the unit terminator, (1/15) and, in the case of the last subfield, by the field terminator, (1/14) and shall take the form

({Y|mY|k(mY,...),...},...)

where

Y implies Z|Z(*)|Z(n);

Z =
- A signifies character data;
- I signifies implicit-point representation;
- R signifies explicit-point unscaled representation;
- S signifies explicit-point scaled representation;
- C signifies character mode bit field;
- B signifies bit field data;
- X signifies unused character positions. (The content of the unused position is not specified and shall be ignored in interchange.);

{ } implies the enclosed expression shall be treated as an entity for purposes of repetition and nesting;

| implies an alternative choice;

(*) and (n) are field width specifications;

* is an arbitrary user delimiter;

n is a positive integer specifying field width (see 7.1.4);

m,k are positive integers signifying the number of times the following data type or group of data types, respectively, shall be repeated;

... implies repetition of the preceding expression.

The use of the format controls is governed by the following rules :

a) The order of fields and their types specified with format controls shall correspond to the data field when the format controls shall be traversed from left to right expanding the nested terms from the left. If the data field is not exhausted, the format shall repeat from the left-hand parenthesis corresponding to the next to the last right-hand parenthesis not including those parentheses used to delimit field width and using the associated repetition factor if any. If there is no such right-hand parenthesis, format control shall revert to the first left parenthesis of the format specification.

b) Z - implies delimiting of the DR fields by the unit terminator, (1/15), and in the case of the last subfield, by the field terminator, (1/14).

c) (*) - implies the presence of the character, *, as a terminating user delimiter for the corresponding data subfield where * is an arbitrary character. The delimiter of the last subfield of a data field shall be replaced by a field terminator, (1/14). The user delimiter may be a national variant character or a character from the default set as specified in 7.1.

d) Data fields for I-type, R-type, and S-type will specify a number in a form defined by the appropriate numeric representation standards. An R-type field may contain a fully defined S-type numeric form.

e) Data fields containing character mode bit string data (C-type) will specify bit strings as a sequence of the characters "0" or "1" corresponding to the digits in the bit string represented.

f) Fixed-length bit fields (B-type with field width specification) shall be specified by a format and shall not have subfield delimiters. The width of a fixed-length bit field shall be specified in bits. Vectors and arrays containing fixed-length bit data shall have each subfield adjacent to the preceding subfield. The last byte of a fixed-length bit field or subfield or a series of adjacent fixed-length bit fields or subfields shall be filled on the right with binary zeros. The field shall be terminated by the appropriate field terminator. The first of a series of fixed-length bit fields shall start on a byte boundary and a series of fixed-length bit fields shall not have an implicit repetition of the format from the left parenthesis.

g) Variable-length bit fields (B-type without field width specification) shall be specified by the following format :

byte 0	bytes 1 to n	bytes n + 1 to m	
field length count	bit field length	bit field data comprising binary digits	zerofill to complete byte

The field length count shall be a single digit specifying the number of decimal digits in the bit field length. The bit field length shall be a sequence of decimal digits specifying the length of the bit field in bits. A variable-length bit field shall start on a byte boundary.

NOTE — Multiple variable-length bit fields (vectors and arrays) may be specified by the use of an appropriate format statement and variable-length bit fields may be used in mixed data fields.

6.2.4 The number of elements in vectors and arrays

The extent (number of elements) of a vector shall be specified by its label, the application of format controls or the use of delimiters. The dimension and extent of an array shall be specified by the Cartesian label. A Cartesian label shall not consist of a single vector label composed of a single element containing solely digits and commas.

NOTE — If an array has a fixed dimension and extent in all data records and a Cartesian label is not required, the label may be replaced by an array descriptor comprising the dimension of the array followed by the extent of each vector, all separated by commas.

If the absence of the Cartesian label or array descriptor, the DR data field shall be preceded by a positive integer giving the dimension of the array and a series of positive integers giving the extent of each dimension. The elements of this array description shall be delimited by (1/15). The order of an array shall be in the form of a row.

7 Coded character set extensions

7.1 Use of coded character sets

This International Standard requires the use of the International Reference Version of ISO 646 in a 7- or 8-bit environment for all leaders, directory entries and data field controls, including system delimiters and format controls. National variant characters are not permitted in control fields except in names, labels and user delimiters. Reference is made in this International Standard to ISO 2022 for the use of extended codes in data fields, user delimiters, names and labels. The use of coded character set extensions as specified in ISO 2022 as default character sets in data fields, names, labels, and for user delimiters shall be limited to sets having three or four character escape sequences associated with the announcing sequences as specified in ISO 2022.

NOTE — Default character set extensions may be invoked for a file or separately for each field as described in this clause. Within data fields, national variant sets and any C0, C1, G0, G1, G2 and G3 character sets may be used in the manner described by ISO 2022 without restriction to length of escape sequences.

When an extended G1 set has been specified as a default character set in a 7-bit environment, the name, label and user delimiter fields shall begin with the G0 set unless an SO control is present to invoke the G1 set. The scope of the invocation shall terminate with the end of the field.

7.2 Invocation of default code sets for fields

A default character set having an *n*-character escape sequence for each field shall be invoked by the placing of a (2/0)(2/1)(2/0) in DDR RP 17 to 19 and increasing the field control length by three. The three additional control characters immediately following the DDR field controls shall contain the last $(n-1)$ characters of the escape sequence used to specify the extended character set, where "n" is less than or equal to "4", for the corresponding data field. The characters from the escape sequence shall be left justified and the field shall be filled on the right with (2/0) if necessary. When there is no character set extension specified for a field, these three characters shall be spaces.

7.3 Invocation of default code set for file

A default character set shall be invoked for a file by placing the last $(n-1)$ characters of its designating n character escape sequence, where "n" is less than or equal to "4", in DDR RP 17 to 19. The characters from the escape sequence shall be left justified and the field shall be filled on the right with (2/0) if necessary.

7.4 Scope of extended code sets used in data fields of any file

The characters of extended code sets may be used in all data fields of all files. Each field of each record shall begin with a character from the extended character set designated by the field controls or by the three characters in DDR RP 17 to 19. If other escape sequences are used within the data field, their scope shall terminate at the end of each field.

7.5 Use of multiple byte character sets

NOTE — Except for B-type fields, multiple byte character sets may be used in data fields, in field names, in vector and Cartesian labels, and in user delimiter fields.

This use is subject to the following conditions :

a) The field names and labels are written in the designated default set. Other escape sequences may occur in the names and labels, and the scope of the invocation shall terminate at the end of the subfield excluding the terminator.

b) The delimiters (2/1) and (2/10) of the vector and Cartesian labels shall be written in the multiple byte set.

c) If multiple byte delimiters are used as the user delimiter, they shall be invoked and their scope shall terminate within the parentheses of the applicable format.

7.6 Field widths with extended character sets

When the use of extended character sets requires the presence of escape sequences, shiftin, shiftout or other control characters, the subfield width shall be indicated by delimiters (see 6.2.1 and 6.2.3.3) and fixed width formats shall not be used.

Annex A

Guidelines for implementation

(This annex does not form part of this International Standard.)

A.1 Mapping structures into the interchange structure

The use of this International Standard implies a mapping of the user's internal or processing data structure into the interchange data structure. The interchange structure may be classified as a set of ordered rooted tree (hierarchical) structures. In the case of a single, non-repetitive structure the interchange structure is reduced to a single logical record. The problem then is to map the commonly used processing structures on to a collection of tree structures.

This mapping, in most cases, can be accomplished directly without loss of structure information. In other cases, pointer linkages in the original structure may require changes and replacement by value linkages through which the original structure could be reconstructed. Alternatively, the original structure can be reduced to an equivalent simpler form which maps into the interchange structure. The required machine readable data descriptive record can be produced automatically from the sender's data dictionary, logical schema or other automated data description. This annex suggests what could be done but not how, as this is strongly dependent on the original system implementation techniques.

Suggestions are made for the mapping of the following data structures into the interchange structure :

a) Sequential structures

b) Relational structures

c) Hierarchical structures

d) Network structures

e) Indexes

A.1.1 Sequential structures

Sequential structures include repetitive elementary fields. The fields of the storage structure can be mapped into either a collection of elementary fields or structured fields whichever is more appropriate to the interchange needs. Highly repetitive, fixed-format data should be placed in the form of rows into an array to prevent excessive storage overhead. Advantage may be taken of the fact that formats repeat from the left opening parenthesis until the data field is exhausted. In a similar fashion, the Cartesian label can provide a repetitive description of the data in the form of rows. This restructuring function can easily be made part of the input routines. It should be noted that a level one system may be able to receive and display higher level structures at the field level even though it cannot automatically process those fields. Multiple data files such as a main file and authority files can be mapped into multiple data descriptive files. File and field names are contained in the appropriate DDR fields.

A.1.2 Relational structures

The relational structure (or flat table) can be mapped into the interchange structure at one n-tuple to a logical record using multiple fields or the vector structure in a single field. Several relations can be mapped into several data descriptive files as required. The relation names and the domain names would be transmitted in the appropriate DDR fields.

Alternatively, one can regenerate and transmit a sequential hierarchical form which represents the original data base in its normal forms.

A.1.3 Hierarchical structures

Conceptually a single hierarchical structure can be mapped into the interchange structure using a single logical record and the appropriate hierarchical description. In actual practice for structures of large extent this leads to processing problems both for the sender and receiver. A more practical solution is to denote each node or suitable collection of nodes forming trees of reasonable size as an entity to be mapped into a single logical record. The linkages which are destroyed by this process shall be replaced by a data field containing the value link which can subsequently be used to reconstruct the linkage. This data field contains a list of the unique identifiers of the logical records previously linked with each node. Either predecessor or successor nodes may be listed. Multiple instances of such data fields can be used to preserve multiple classes of linkages based on different data base attributes.

The methods specified in this International Standard may be used to transmit a collection of trees in a single record. The list of tag pairs of the DDR, the pre-order traversal sequence and the processing algorithm of annex D are sufficient to create the corresponding binary tree.

A.1.4 Network structures

A network database may be resolved into its relational counterparts and transmitted as a set of relations. Each record type including linkage records will form a relation to be described and transmitted as a data descriptive file.

Storage structures, classed as directed graphs, are sufficiently complex to prevent their representation in hierarchical form unless one resorts to the breaking of linkages and providing lists of logical record (node) identifiers. The principles are identical to those used for hierarchical structures but the processing algorithms for reconstruction must detect closed paths and act accordingly.

A.1.5 Indexes

Indexes, including chains or threads, form structured information about a database which is implementation-dependent and not directly transmittable. However, a logical equivalent exists wherein system-oriented pointers are replaced by logical record identifiers. An index of a database attribute can be mapped to a data descriptive file where each logical record is an instance of a value of the attribute and an associated list of unique identifiers of logical records (or nodes) having this value of the attribute. In this manner indexes may be described and interchanged.

A.2 Guidelines for special implementation

In order to preserve maximum interchangeability, special implementations of this International Standard should preserve the specifications for the leader, directory and other control features and should limit the special implementation to changes in data field formats. Special implementations are anticipated to vary in coded character sets and application-specific field contents and structures. Guidelines are presented here for both of these variations.

A.2.1 Character sets

It is recommended that ISO 2022 should be used as a standard for the extension of ISO 646. The use of ISO 2022 will permit common extended character set processing algorithms.

A.2.2 Application-specific systems

The potential provision of the control features for application specific implementations is primarily intended to accommodate existing standard applications and new implementations should use the provisions of this International Standard to ensure a wide base of interchangeability.

Application-specific systems may provide a complete external description for each field or may use some of the data types of this International Standard. In either case, the special practice should be signalled by the use of a private control character in DDR RP 9 chosen from columns 4 to 7 of ISO 646. This practice will ensure that future extensions of this International Standard will not invalidate private agreements.

NOTE — If users require a new data structure that might be widely applied, they should apply to ISO/TC 97 for an extension of this International Standard.

A.2.3 Restrictions to user environments

This International Standard has been constructed to permit its application to sophisticated interchanges and thus may permit more complex constructions than some user groups may find necessary. In this case and when the user can be reasonably assured of the scope of his interchange group, restrictions can be placed on the use of this International Standard in a manner such that upward compatibility is maintained. Thus, processing by a complete implementation is assured and processing within the restricted user group may be facilitated.

The user shall be aware that in doing this he may reduce his ability to receive interchange files from senders who have not adopted any restrictions. In this case, when the number of interchanges is small, the user of limited software will be able to receive a file and process it into an output file composed of records containing data fields. These records may be further processed by application specific software into a useful form.

If users have to apply restrictions which introduce upward incompatibility, they should make use of the application indicator (DDR RP 9) to indicate a privately agreed-upon system. If the users wish to transmit auxiliary information associated with a restricted system, they should do so in the field associated with DDR tag 0...2. This field can have a similar purpose to a DDR record and contain information including restriction identification.

Two general areas in which restrictions might be useful are those based on computer size, and those based on computer languages. The following are guidelines for the establishment of restrictions that may be useful in those areas and how this International Standard can be used to facilitate the restricted interchange.

A.2.3.1 Small computer restrictions

This International Standard has been constructed in such a way that by parallel processing of the logical subfields (i.e., directories, data descriptive fields, and data fields), traversing each serially, the memory requirement for the storage of processing information may be reduced. The maximum block and record sizes are obtainable from the appropriate fields of the file labels. Also, the length of each logical record is available as a control word, and field sizes are given in the directory of each record.

In the case where the above information does not suffice, then by private agreement the user group can

a) restrict the data fields to fully formatted fields and convey the maximum field and subfield sizes simply by not using the repetition feature of an entire format (i.e., from left-most parenthesis). During the format scan while processing the DDR, the maximum subfield and field widths as well as the maximum dimensions of each vector and array can be determined;

b) for variable-length fields terminated by delimiters, provide in the user reserved field 0...2, a set of maximum field widths expressed in the form of a format which when parsed will yield maximum sizes and dimensions. It is suggested that this information be constructed as a subdirectory with associated fields so that it can be processed by similar algorithms as for the DDR;

c) substitute the maximum dimensions for vectors and arrays for the name and Cartesian label, as appropriate.

It is suggested that the 0...1 field in the sub-DDR be used to provide a positive identification of the restricted user group. Examination of this field should precede further processing.

A.2.3.2 Language restrictions

For those users who wish to facilitate an implementation interface to a specific higher level language, the following private agreement restrictions may be useful :

a) Data structures and types : Restrict the data structures and types possible in this International Standard to the subsets which are permitted by the higher level language under consideration (for example restrict a FORTRAN interface to elementary data, vectors, arrays and fixed length subfields).

b) Field names and tags : Restrict to those character strings which can be used to form variable names in the language under consideration. The concatenation of these strings could be used to generate hierarchical names if the high level language could accept and use them.

A.3 Unique identifiers

Data records often do not require a unique identifier since this purpose is served by the value of specific fields or combination of fields such as the principal key of the relational data model. Nevertheless, a logical record may represent all of the information which is associated with an individual whose identity in the data base is unique. A unique identifying field is often of value particularly if subsequent updating is to occur. In the cases where an arbitrary identifier is to be supplied, sequential integers often suffice. In other cases, a data field naturally serves this purpose and can be used. Where possible, an elementary field is desirable but structured fields such as a two element vector can also be used. This does not preclude the inclusion of information in other data fields which can be used to establish a relationship with other individuals within this or other populations.

Sorting is desirable if references to displayable representation are to be used to locate and correct errors, to understand the interchange file, or to provide for a merging of files as received. Where variable-length fields are used, provision for filling may be necessary, if the file is to be sorted into order.

Annex B

Data field examples

(This annex does not form part of this International Standard.)

B.1 General

This annex contains several examples of data descriptive fields and the corresponding data fields. Since the permitted combination of structure, type, names, labels and format controls is very large, only a sample is given. Throughout the examples, the following printable characters have been substituted for delimiters :

; — Field terminator (1/14)

& — Subfield delimiter (1/15)

B.1.1 Elementary character data elements (DDR RP 9 = (2/0); RP 10 and 11 = "00")

Example	Field contents	
	DDR	DR
1	AUTHOR;	Doe,Jane;
2	AGE;	24;
3	HEIGHT;	5.5;
4	WEIGHT;	2.45E2;
5	BITSTR;	010101;

B.1.2 Compound data elements (DDR RP 9 = (2/0); RP 10 and 11 = "06")

B.1.2.1 Elementary fields

Example	Data type	Field contents	
		DDR	DR
1	Character	0000;&NAME;	DOE,JANE;
2	Implicit-point	0100;&AGE;	18;
3	Explicit-point	0200;&GPA;	3.46;
4	Explicit-point-scaled	0300;&DIST;	+0.5E+02;
5	Character mode bit string	0400;&BITSTR;	010101;
6	Bit field (fixed)	0500;&BITFLDF&B(6);	(01010100);
7	Bit field (variable)	0500;&BITFLDV&B;	16(01010100);

where the digits in parentheses are binary digits.

B.1.2.2 Vector fields

B.1.2.2.1 Delimited character vector with name and subfield labels

DDR 1000;&MAILING ADDRESS&NAME!STREET!CITY!STATE!ZIP;
DR DOE,JOHN&1010 MAPLE ST.&OSHGOSH&OHIO&12345;

B.1.2.2.2 Formatted implicit point vector with name and subfield labels

DDR 1100;&POPULATION&CY60!CY65!CY70!CY75&(4I(6));
DR 765432987345903231897654;

B.1.2.2.3 Delimited explicit point vector with name and no labels

DDR 1200;&GRAINS;
DR 3.46&2.47&11.94;

B.1.2.2.4 Formatted mixed vectors with name and no labels (this also can be interpreted as an array)

DDR 1600;&LIVESTOCK&&(A(,),I(5),R(5));
DR PIGS,0274437.46STEERS,1776447.84;

B.1.2.3 Array fields

B.1.2.3.1 Delimited character array with name and Cartesian label

DDR 2000;&PROPERTIES&GOLD!SODIUM!COPPER*DENSITY!COLOUR!ACTIVITY;
DR HIGH&YELLOW&INERT&LOW&GREY&HIGH&MEDIUM&REDDISH&LOW;

Equivalent table

	PROPERTIES		
	DENSITY	COLOUR	ACTIVITY
GOLD	HIGH	YELLOW	INERT
SODIUM	LOW	GREY	HIGH
COPPER	MEDIUM	REDDISH	LOW

B.1.2.3.2 Formatted implicit point array with no name and no label

DDR 2100;&&&(9I(2));
DR 2&3&3&124788334672441921;

B.1.2.3.3 Delimited explicit point array with name

DDR 2200;&UNIT MATRIX;
DR 2&3&3&1.&0.&0.&0.&1.&0.&0.&0.&1.;

B.1.2.3.4 Formatted mixed array with name and column vector label containing a null row vector label

DDR 2600;&TABLE II&*METAL!DENSITY!COLOUR!ACTIVITY&(A(:),R(4),A(,)R(4));
DR GOLD:14.8YELLOW,-1.3SODIUM:0.63GREY,4.81COPPER:10.6REDDISH,.043;

B.2 Use of extended coded character sets

B.2.1 Default coded character sets for fields

The use of a default G1 coded character set for fields is specified by establishing values in the DDR RP 17 to 19 and in the extended field controls for each field as shown below :

a) DDR RP 17 to 19 contains (2/0) (2/1) (2/0) signifying that individual fields may contain extended coded character sets;

b) DDR RP 10 and 11 contains "09" signifying that there are 9 bytes containing field controls the last three of which are escape sequence information;

c) the DDR field controls which precede the field name contain the data structure/type, printable delimiters and escape sequence information.

Several examples of these extended field controls are shown below for several registered character sets :

Default G1 characters	Field controls - RP 0	1	2	3	4	5	6	7	8
a) JIS Katakana	0	0	0	0	;	&	(2/9)	(4/9)	(2/0)
b) Italian Graphics	0	0	0	0	;	&	(2/9)	(5/9)	(2/0)
c) Swedish Graphics for names	1	0	0	0	;	&	(2/9)	(4/8)	(2/0)
d) No extended set used	0	0	0	0	;	&	(2/0)	(2/0)	(2/0)

The data fields in cases a) and b) would contain unstructured text formed from ISO 646 as the G0 set and the indicated G1 set. Case c) would contain sequences of character strings separated by delimiters. Case d) shows the controls for fields where no G1 set is specified.

NOTE — In cases a) to d), RP 8 contains a (2/0) due to the use of 3-character escape sequences in the examples. Other bit patterns will be present if 4-character escape sequences are used.

B.2.2 Default character sets for the file

The use of a default G1 character set for an entire file is specified by placing the final characters of the escape sequence in DDR RP 17 to 19.

To invoke the German Graphics characters as the G1 default for the entire file DDR RP 17 to 19 would contain (2/9) (4/11) (2/0). One or more data fields would contain, in the 8-bit environment, high order codes and, in the 7-bit environment, SO and SI controls modifying the meaning of the subsequent bit patterns.

B.2.3 Inline code extensions

Inline escape sequences may be used freely in data fields. The recipient is forewarned of such use by the contents of DDR RP 7. The character "E" implies that inline coded character set extensions will be found and the character "SP" implies that inline escape sequences will not be found.

Thus for a file containing inline escape sequences

a) DDR RP 7 contains "E" and,

b) one or more data fields will contain an escape sequence, ESC (I) (F), where (I) is one or more intermediate characters and (F) is the terminal character of the escape sequence.

These escape sequences are not limited to those appropriate to a G1 set but may be any one of all sequences permitted under ISO 2022 in both 7- and 8-bit environments.

Annex C

DDF model and logical record structure

(This annex does not form part of this International Standard.)

C.1 DDF model

In order to compare the DDF model with other file models, it is desirable to describe the data structure of the DDF in graph theoretic terminology i.e. as a set of rooted ordered trees. Each tree corresponds to a logical record. The root node of each tree is the record identifier field that shall contain a unique record identifier. The data fields are associated with the subordinate nodes of the tree, these being intermediate and leaf nodes.

The DR tree structures shall be an instance of a tree derived from the generic tree described in the DDR by a binary relation and the pre-order transversal sequence of the generic tree. A specific data tree is described by the binary relation in the DDR and the pre-order transversal required of the specific tree in the DR. The DR tree structure may be incomplete in that nodes that are possible in the DDR structure not containing data nor required to retain structure may be absent. Multiple instances of a DDR subtree are permitted in a DR tree and the order of occurrence in the pre-order transversal sequence provides identification.

The contents of the nodes is data which may have a further regular structure [i.e., elementary, n-tuple (vector) or array]. The DR data fields contain data items. The data types are character, three numeric forms, character mode bit string and bit field. The DR data fields are described by formats in the DDR and the data fields can contain mixed data types.

The tree structure is not mandatory. Without it, the file is a collection of uniquely identified logical records, each containing a set of named (tagged) data fields. The fields in a DR may be absent and more than one instance of a field may occur but association cannot be inferred from the order of the fields. This form can be considered to be a single level rooted unordered tree, where each field is implicitly connected to the record identifier node.

One property of the DDF model is that, due to the requirement of a unique record identifier as the root node of each tree, an entire set of trees can be expressed as a single ordered tree rooted to the DDR.

C.2 Hierarchical logical records

This clause describes the means by which the inherently linear structure of the tagged fields can be reconstituted into a hierarchical structure. The hierarchy or ordered rooted tree structure used has data fields as its nodes and the directed logical associations between data field contents as the links in the tree structure. A rooted tree has a unique node, the root, and each node may have zero or more ordered subtrees. A specific instance of the tree structure for a record may have several subtrees formed from multiple occurrences of the data fields having the same tag. However, such a tree shall be derived from (i.e., have the same logical field associations as) a generic tree for the data record which contains each field once and contains all of the possible links between fields.

A rooted tree has directed links and shall have only one root node which has no links entering it. Each remaining node may be entered by only one link. The nodes which have no departing links are called leaf nodes. Once the root node is specified, the direction of all links is specified from the root node to the leaf node. The hierarchical level is determined by numbering the root node (1) and by incrementally numbering each successive node on all paths to the leaf nodes. Trees in this International Standard are drawn hanging from the root with the subtrees of each node ordered from left to right. Examples of ordered rooted trees are shown in figure 13. The tree described is not a binary tree which preserves left- and right-handedness in the absence of one binary branch.

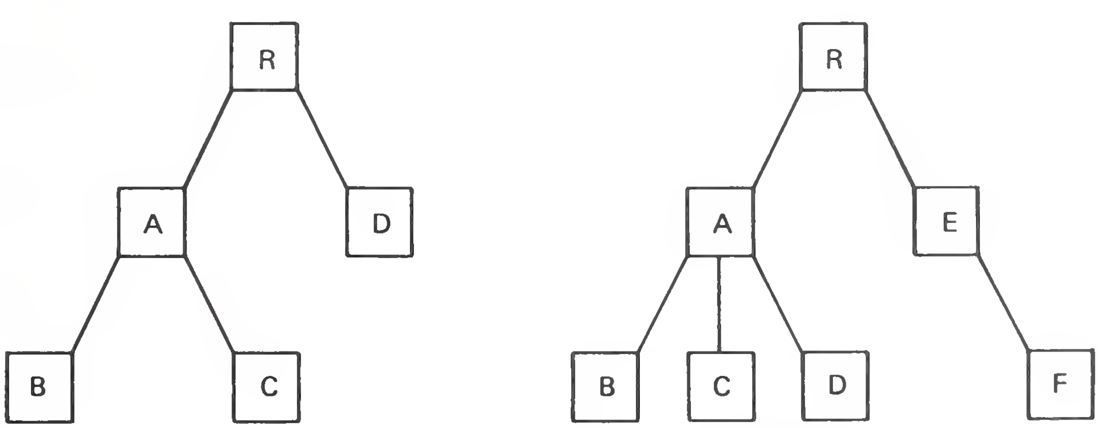


Figure 13 — Examples of ordered rooted trees

The method used here to represent rooted tree structures is closely related to the user view of the data and not the most usual computer representation of tree structures which is the corresponding binary tree (see figures 14 and 16).

The tags serve as node (i.e., data field) identifiers. A tree structure shall be designated by two pieces of information

a) the ordered pairs of node identifiers (i.e., tags) of the generic structure in the top down order of parent-offspring and;

b) the ordered list of node identifiers (i.e., tags) in the pre-order traversal sequence.

The generic structure of a logical record might be the tree shown in figure 14.

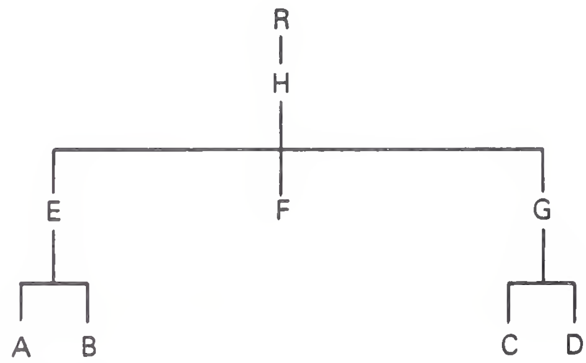


Figure 14 — Generic structure of a logical record

It has the set of ordered pairs of tags RH, HE, EA, EB, HF, HG, GC, GD, where R is the unique record identifier tag, 0...1. The pre-order traversal sequence of the tags of this structure is - RHEABFGCD and each occurrence of a node is unique.

A specific instance of the previous generic structure having multiple occurrences of data fields having the same tag might be

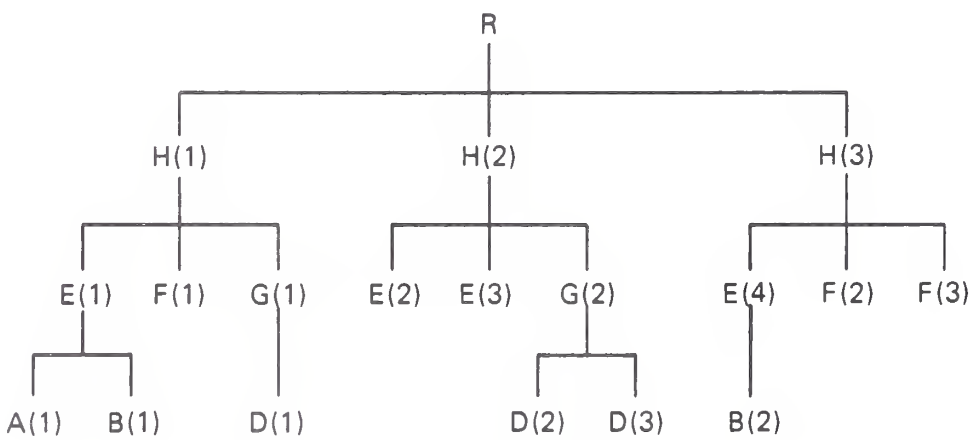


Figure 15 — Instance of a user data tree

where R is the unique record identifier tag, and ordering subscripts for repetitive tags have been added for clarity. Its pre-order traversal sequence is

RH(1)E(1)A(1)B(1)F(1)G(1)D(1)H(2)E(2)E(3)G(2)D(2)D(3)H(3)E(4)B(2)F(2)F(3)

which may be written without subscripts with no loss of meaning. Note that only the root node, R, is unique and that several instances of the field H, E, A, etc., may occur, in which case they are designated by repetition of the same tag.

The pre-order traversal sequence is typically combined with link information to represent the structure of a tree. In most cases the corresponding binary tree is used. This binary tree has the same pre-order traversal sequence as the ordered tree and the link to the first subtree of a node forms a left-hand link in the binary tree and all sibling nodes are linked in order by right-hand links between siblings. These new links are easily derived from the list of tag pairs and the pre-order traversal sequence. The user can then construct whatever linking convention his system requires.

An algorithm which produces the left- and right-hand links of the corresponding binary tree is given below using the generic tree of this annex as an example :

ALGORITHM L : Given that P(q) is a binary relation on a set of nodes, q, expressing the directed inter-node associations permitted and T(q) is the pre-order traversal sequence of a tree (or forest) :

Construct L and R, vectors of indices in T, expressing the left- and right-hand linkages of the corresponding binary tree.

NOTE — Read -E as "not an element of" and "| |" as "concatenated to".

L0 [Initialize] m ← number of pairs in P

n ← number of nodes in T

S ← 0, working stack containing the indices in T of the nodes of a path from the root to a node, T(S(j)).

L ← 0

R ← 0

L1 [Set stack to root node] j ← 1, S(1) ← 1

L2 [Set node index to root node] i ← 1

L3 [Advance to next node] i ← i + 1

L4 [Test for extent of T] if i > n end

L5 [Test paired stack node and i-th node] if T(S(j))| |T(i) -E P go to L8

L6 [Store left link for stack node] L(S(j)) ← i

L7 [Add i-th node to stack] j ← j + 1, S(j) ← i, go to L3

L8 [Test paired penultimate stack node and i-th node] if T(S(j – 1))| |T(i) -E P go to L11

L9 [Store right link for stack node] R(S(j)) ← i

L10 [Reset stack node] S(j) ← i, go to L3

L11 [Regress to lower node in path] j ← j – 1

L12 [Test for disjoint node (new tree)] if j = 0 go to L9

L13 [Retest new node pair] go to L8

Example :

P: RH, HE, EA, EB, HF, GC, GD, HG

j:	1	2	3	4	5	6	7	8	9
T:	R	H	E	A	B	F	G	C	D
L:	2	3	4	0	0	0	8	0	0
R:	0	0	6	5	0	7	0	9	0

The algorithm is valid for rooted-oriented trees with or without replicate tags for nodes. It will process an ordered set of trees. The "true" branch at L12 may be set to an error if it is desired to limit processing to trees.

The corresponding binary tree for the ordered tree shown in figure 15 is shown in figure 16 :

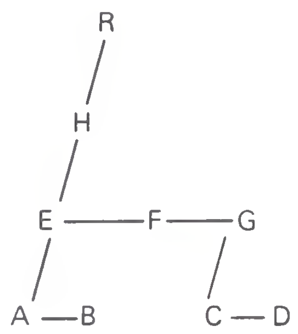


Figure 16 — Binary tree corresponding to the rooted tree shown in figure 15

X3.115-1984 Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)
X3.116-1986 Recorded Magnetic Tape Cartridge, 4-Track, Serial 0.250 Inch (6.30 mm) 6400 bpi (252 bpmm), Inverted Modified Frequency Modulation Encoded
X3.117-1984 Printable/Image Areas for Text and Facsimile Communication Equipment
X3.118-1984 Financial Services — Personal Identification Number — PIN Pad
X3.119-1984 Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter
X3.120-1984 Contact Start/Stop Storage Disk
X3.121-1984 Two-Sided, Unformatted, 8-Inch (200-mm), 48-tpi, Double-Density, Flexible Disk Cartridge for 13 262 ftpr Two-Headed Application
X3.124-1985 Graphical Kernel System (GKS) Functional Description
X3.124.1-1985 Graphical Kernel System (GKS) FORTRAN Binding
X3.125-1985 Two-Sided, Double-Density, Unformatted 5.25-inch (130-mm), 48-tpi (1,9-tpmm), Flexible Disk Cartridge for 7958 bpr Use
X3.126-1986 One- or Two-Sided Double-Density Unformatted 5.25-inch (130-mm), 96 Tracks per Inch, Flexible Disk Cartridge
X3.128-1986 Contact Start-Stop Storage Disk — 83 000 Flux Transitions per Track, 130-mm (5.118-in) Outer Diameter and 40-mm (1.575-in) Inner Diameter.
X3.129-1986 Intelligent Peripheral Interface, Physical Level
X3.130-1986 Intelligent Peripheral Interface, Logical Device Specific Command Sets for Magnetic Disk Drive
X3.131-1986 Small Computer Systems Interface
X3.136-1986 Serial Recorded Magnetic Tape Cartridge for Information Interchange, Four and Nine Track
X3.140-1986 Open Systems Interconnection — Connection Oriented Transport Layer Protocol Specification
X11.1-1977 Programming Language MUMPS
IEEE 416-1978 Abbreviated Test Language for All Systems (ATLAS)
IEEE 716-1982 Standard C/ATLAS Language
IEEE 717-1982 Standard C/ATLAS Syntax
IEEE 770X3.97-1983 Programming Language PASCAL
IEEE 771-1980 Guide to the Use of ATLAS
ISO 8211-1986 Specifications for a Data Descriptive File for Information Interchange
MIL-STD-1815A-1983 Reference Manual for the Ada Programming Language

---

X3/TRI-82 Dictionary for Information Processing Systems (Technical Report)

American National Standards for Information Processing

X3.1-1976 Synchronous Signaling Rates for Data Transmission
X3.2-1970 Print Specifications for Magnetic Ink Character Recognition
X3.4-1986 Coded Character Sets — 7-Bit ASCII
X3.5-1970 Flowchart Symbols and Their Usage
X3.6-1965 Perforated Tape Code
X3.9-1978 Programming Language FORTRAN
X3.11-1969 General Purpose Paper Cards
X3.14-1983 Recorded Magnetic Tape (200 CPI, NRZI)
X3.15-1976 Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission
X3.16-1976 Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.17-1981 Character Set for Optical Character Recognition (OCR-A)
X3.18-1974 One-Inch Perforated Paper Tape
X3.19-1974 Eleven-Sixteenths-Inch Perforated Paper Tape
X3.20-1967 Take-Up Reels for One-Inch Perforated Tape
X3.21-1967 Rectangular Holes in Twelve-Row Punched Cards
X3.22-1983 Recorded Magnetic Tape (800 CPI, NRZI)
X3.23-1985 Programming Language COBOL
X3.25-1976 Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.26-1980 Hollerith Punched Card Code
X3.27-1978 Magnetic Tape Labels and File Structure
X3.28-1976 Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links
X3.29-1971 Specifications for Properties of Unpunched Oiled Paper Perforator Tape
X3.30-1971 Representation for Calendar Date and Ordinal Date
X3.31-1973 Structure for the Identification of the Counties of the United States
X3.32-1973 Graphic Representation of the Control Characters of American National Standard Code for Information Interchange
X3.34-1972 Interchange Rolls of Perforated Tape
X3.36-1975 Synchronous High-Speed Data Signaling Rates between Data Terminal Equipment and Data Communication Equipment
X3.37-1980 Programming Language APT
X3.38-1972 Identification of States of the United States (Including the District of Columbia)
X3.39-1986 Recorded Magnetic Tape (1600 CPI, PE)
X3.40-1983 Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)
X3.41-1974 Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange
X3.42-1975 Representation of Numeric Values in Character Strings
X3.43-1986 Representations of Local Time of Day
X3.44-1974 Determination of the Performance of Data Communication Systems
X3.45-1982 Character Set for Handprinting
X3.46-1974 Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)
X3.47-1977 Structure for the Identification of Named Populated Places and Related Entities of the States of the United States for Information Interchange
X3.48-1986 Magnetic Tape Cassettes (3.81-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)
X3.49-1975 Character Set for Optical Character Recognition (OCR-B)
X3.50-1986 Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets
X3.51-1986 Representations of Universal Time, Local Time Differentials, and United States Time Zone References
X3.52-1976 Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)
X3.53-1976 Programming Language PL/I
X3.54-1986 Recorded Magnetic Tape (6250 CPI, Group Coded Recording)
X3.55-1982 Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase encoded
X3.56-1986 Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.57-1977 Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control
X3.58-1977 Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)
X3.59-1981 Magnetic Tape Cassettes, Dual Track Complementary Return-to-Bias (CRB) Four-States Recording on 3.81-mm (0.150-Inch) Tape
X3.60-1978 Programming Language Minimal BASIC
X3.61-1986 Representation of Geographic Point Locations
X3.62-1979 Paper Used in Optical Character Recognition (OCR) Systems
X3.63-1981 Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)
X3.64-1979 Additional Controls for Use with American National Standard Code for Information Interchange
X3.66-1979 Advanced Data Communication Control Procedures (ADCCP)
X3.72-1981 Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.73-1980 Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)
X3.74-1981 Programming Language PL/I, General-Purpose Subset
X3.76-1981 Unformatted Single-Disk Cartridge (Top Loading, 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)
X3.77-1980 Representation of Pocket Select Characters
X3.78-1981 Representation of Vertical Carriage Positioning Characters in Information Interchange
X3.79-1981 Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures
X3.80-1981 Interfaces between Flexible Disk Cartridge Drives and Their Host Controllers
X3.82-1980 One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)
X3.83-1980 ANSI Sponsorship Procedures for ISO Registration According to ISO 2375
X3.84-1981 Unformatted Twelve-Disk Pack (200 Megabytes) (General, Physical, and Magnetic Requirements)
X3.85-1981 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge
X3.86-1980 Optical Character Recognition (OCR) Inks
X3.88-1981 Computer Program Abstracts
X3.89-1981 Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)
X3.91M-1982 Storage Module Interfaces
X3.92-1981 Data Encryption Algorithm
X3.93M-1981 OCR Character Positioning
X3.94-1985 Programming Language PANCM
X3.95-1982 Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters
X3.96-1983 Continuous Business Forms (Single-Part)
X3.98-1983 Text Information Interchange in Page Image Format (PIF)
X3.99-1983 Print Quality Guideline for Optical Character Recognition (OCR)
X3.100-1983 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network
X3.101-1984 Interfaces Between Rigid Disk Drive(s) and Host(s)
X3.102-1983 Data Communication Systems and Services — User-Oriented Performance Parameters
X3.103-1983 Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in)
X3.104-1983 Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded
X3.105-1983 Data Link Encryption
X3.106-1983 Modes of Operation for the Data Encryption Algorithm
X3.110-1983 Videotex/Teletext Presentation Level Protocol Syntax
X3.111-1986 Optical Character Recognition (OCR) Matrix Character Sets for OCR-M
X3.112-1984 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk
X3.114-1984 Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983

(continued on reverse)