INTERNATIONAL STANDARDIZED PROFILE

ISO/IEC ISP 10607-2

Second edition
1995-12-15

Information technology — International Standardized Profiles AFTnn — File Transfer, Access and Management —

Part 2:
Definition of document types, constraint sets and syntaxes

Technologies de l'information — Profils normalisés internationaux AFTnn — Transfert, accès et gestion de fichier —

Partie 2: Définition de types de documents, ensembles de contraintes et syntaxes

Reference number
ISO/IEC ISP 10607-2:1995(E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Contents

Page

Foreword iv

Introduction v

1 Scope 1

1.1 General 1

1.2 Position within the taxonomy 1

2 Normative references 1

2.1 Identical Recommendations | International Standards 1

2.2 Additional references 1

3 Definitions 2

4 Abbreviations 2

5 Conformance 2

6 Document type definitions 2

6.1 NBS-9 File directory file document type 3

6.2 INTAP-1 Record file document type 5

6.3 NBS-6 Sequential file document type 7

6.4 NBS-7 Random access file document type 10

6.5 NBS-8 Indexed sequential file document type 13

6.6 NBS-10 Random binary access file document type 16

6.7 NBS-11 Indexed file with unique keys document type 18

6.8 NBS-12 Simple text file document type 21

6.9 INTAP-2 Sequential file document type 24

6.10 INTAP-3 Relative file document type 27

6.11 INTAP-4 Indexed file document type 30

6.12 INTAP-5 Indexed file with unique keys document type 33

6.13 CGM-FTAM file document type 37

© ISO/IEC 1995
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.
ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7 Constraint sets 40

7.1 NBS ordered flat constraint set 40

7.2 NBS random access constraint set 41

7.3 INTAP sequential flat constraint set 42

7.4 Two levels depth hierarchical constraint set with unique names 43

8 Abstract syntaxes 45

8.1 Abstract syntax NBS-AS2 45

8.2 Abstract syntax INTAP-AS1 45

8.3 Abstract syntax NBS-AS1 46

8.4 NBS node name abstract syntax 48

8.5 NBS random binary access file abstract syntax 48

8.6 NBS simple text abstract syntax 49

8.7 Abstract syntax INTAP-AS2 49

8.8 Abstract syntax INTAP-AS3 51

9 Transfer syntaxes 52

9.1 Transfer syntax INTAP-TS1 52

Annexes

A Corrigenda 53

B Main differences between the 1st edition (1990/91/94) and the 2nd edition (1995) of this part of ISO/IEC ISP 10607 54

C FTAM-COBOL Language Binding for the INTAP-2, -3, -4, -5 Document Types (Tutorial) 55

C.1 Aim of COBOL document types 55

C.2 Support range of COBOL document types 55

C.3 Achievement method of COBOL document types 59

C.4 Mapping of COBOL data-type 61

C.5 Details of sequential file, relative file and indexed file 64

C.6 Mapping of FTAM service primitive and COBOL statement 65

C.7 Example of the COBOL mapping 68

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. In addition to developing International Standards, ISO/IEC JTC1 has created a Special Group on Functional Standardization for the elaboration of International Standardized Profiles.

An International Standardized Profile is an internationally agreed, harmonized document which identifies a standard or group of standards, together with options and parameters, necessary to accomplish a function or set of functions.

Draft International Standardized Profiles are circulated to national bodies for voting. Publication as an International Standardized Profile requires approval by at least 75 % of the national bodies casting a vote.

International Standardized Profile ISO/IEC ISP 10607-2 was prepared with the collaboration of

- Asia-Oceania Workshop (AOW);
- European Workshop for Open Systems (EWOS);
- Open Systems Environment Implementors' Workshop (OIW).

ISO/IEC ISP 10607 consists of the following parts, under the general title *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management :*

- *Part 1: Specification of ACSE, Presentation and Session protocols for the use by FTAM*
- *Part 2 : Definition of document types, constraint sets and syntaxes*
- *Part 3 : AFT11 - Simple File Transfer Service (unstructured)*
- *Part 4 : AFT12 - Positional File Transfer Service (flat)*
- *Part 5 : AFT22 - Positional File Access Service (flat)*
- *Part 6 : AFT3 - File Management Service*

This second edition cancels and replaces the first edition (ISO/IEC ISP 10607-2:1990), which has been revised. It also incorporates amendment 1:1991 and amendment 2:1994.

Annexes A, B and C of this part of ISO/IEC ISP 10607 are for information only.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Introduction

This part of ISO/IEC ISP 10607 is defined within the context of Functional Standardization, in accordance with the principles specified by ISO/IEC TR 10000, "Framework and Taxonomy of International Standardized Profiles". The context of Functional Standardization is one part of the overall field of Information Technology (IT) standardization activities, covering base standards, profiles, and registration mechanisms. A profile defines a combination of base standards that collectively perform a specific well-defined IT function. Profiles standardize the use of options and other variations in the base standards, and provide a basis for the development of uniform, internationally recognized system tests.

One of the most important roles for an ISP is to serve as the basis for the development (by organizations other than ISO and IEC) of internationally recognized tests and test centres. ISPs are produced not simply to legitimize a particular choice of base standards and options, but to promote real system interoperability. The development and widespread acceptance of tests based on this and other ISPs is crucial to the successful realization of this goal.

The text for this part of ISO/IEC ISP 10607 was developed in close co-operation among the FTAM Expert Groups of the three International OSI/OSE Workshops : OSE Implementors' Workshop (OIW), the European Workshop for Open Systems (EWOS) and the Asia-Oceania Workshop (AOW). This part of ISO/IEC ISP 10607 is harmonized among these three Workshops and it was finally ratified by the Workshops' plenary assemblies.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Information technology — International Standardized Profiles AFTnn — File Transfer, Access and Management —

Part 2:
Definition of document types, constraint sets and syntaxes

1 Scope

1.1 General

This part of ISO/IEC ISP 10607 contains the basic definitions of document types, constraint sets, abstract syntaxes, and transfer syntaxes as used and referenced in the FTAM application ISO/IEC ISP 10607-3 (AFT11), ISO/IEC ISP 10607-4 (AFT12), ISO/IEC ISP 10607-5 (AFT22) and ISO/IEC ISP 10607-6 (AFT3). Additional document types, constraint sets and syntaxes may be defined and added to this part of ISO/IEC ISP 10607 to be referenced by either the existing parts of ISO/IEC ISP 10607 or by further parts yet to be defined.

1.2 Position within the taxonomy

This part of ISO/IEC ISP 10607 is the second part, as common text, of a multi-part ISP identified in ISO/IEC TR 10000-2 as "AFT, File Transfer, Access and Management" (see also ISO/IEC TR 10000-1, 8.2 for the definition of multipart ISPs).

2 Normative references

The following documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC ISP 10607. At the time of publication, the editions indicated were valid. All documents are subject to revision, and parties to agreements based on this part of ISO/IEC ISP 10607 are warned against automatically applying any more recent editions of the documents listed below, since the nature of references made by ISPs to such documents is that they may be specific to a particular edition. Members of IEC and ISO maintain registers of currently valid International Standards and ISPs, and ITU-T maintains published editions of its current Recommendations.

Corrigenda to the base standards referenced : See annex A for a complete list of these documents which are identified in this part of ISO/IEC ISP 10607.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1995, *Information technology - ASN.1 encoding rules : Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

2.2 Additional references

- ISO 1989:1985, *Programming languages - COBOL.*
- ISO 6523:1984, *Data interchange - Structures for the identification of organizations.*
- ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*
- ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*
- ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3 : File Service Definition .*
- ISO 8571-4:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 4 : File Protocol Specification.*
- ISO 8571-4:1988/Amd.4:1992, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 4 : File Protocol Specification - Amendment 4 .*
- ISO/IEC 8571-5:1990, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 5 : Protocol Implementation Conformance Statement Proforma.*
- ISO/IEC TR 10000-1:1992[1)] , *Information technology - Framework and taxonomy of International Standardized Profiles - Part 1 : Framework.*
- ISO/IEC TR 10000-2:1994[1)], *Information technology - Framework and taxonomy of International Standardized Profiles - Part 2 : Principles and Taxonomy for OSI Profiles.*

---

1) Currently under revision.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

- ISO/IEC ISP 10607-1:1995, *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management - Part 1 : Specification of ACSE, Presentation and Session protocols for the use by FTAM.*
- ISO/IEC ISP 10607-3:1995, *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management - Part 3 : AFT11 - Simple File Transfer Service (unstructured).*
- ISO/IEC ISP 10607-4:1995, *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management - Part 4 : AFT12 - Positional File Transfer Service (flat).*
- ISO/IEC ISP 10607-5:1995, *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management - Part 5 : AFT22 - Positional File Access Service (flat).*
- ISO/IEC ISP 10607-6:1995, *Information technology - International Standardized Profiles AFTnn - File Transfer, Access and Management - Part 6 : AFT3 - File Management Service.*

3 Definitions

The terms used in this part of ISO/IEC ISP 10607 are defined in the referenced base standards.

4 Abbreviations

This clause lists only abbreviations as used in clauses 1 to 5.

AFT	Profile sub-class : File Transfer, Access and Management
ASN.1	Abstract Syntax Notation One
FTAM	File Transfer, Access and Management
ISP	International Standardized Profile
OSI	Open Systems Interconnection

5 Conformance

No conformance requirements are specified in this part of ISO/IEC ISP 10607.

NOTE - This part of ISO/IEC ISP 10607 is a register of document types, constraint sets, abstract syntaxes, and transfer syntaxes. Conformance requirements are specified in the parts of ISO/IEC ISP 10607 which reference these objects.

6 Document type definitions

The definition of "lexical order" which appears in some of the following document type definitions, is a local implementation issue.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.1 NBS-9 File directory file document type

6.1.1 Entry number : NBS-9

6.3.2 Information objects

Table 1 - Information objects in NBS-9

document type name	{ iso identified-organization oiw(14) ftamsig(5) document-type(5) file-directory(9) } "NBS-9 FTAM file-directory file"
abstract syntax names a) name for asname1	{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2) } "NBS file directory entry abstract syntax"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax PARAMETERS : := [0] IMPLICIT BIT STRING { -- Kernel group read-filename (0), read-permitted-actions (1), read-contents-type (2), -- Storage group read-storage-account (3), read-date-and-time-of-creation (4), read-date-and-time-of-last-modification (5), read-date-and-time-of-last-read-access (6), read-date-and-time-of-last-attribute-modification (7), read-identity-of-creator (8), read-identity-of-last-modifier (9), read-identity-of-last-reader (10), read-identity-of-last-attribute-modifier (11), read-file-availability (12), read-filesize (13), read-future-filesize (14), -- Security group read-access-control (15), read-legal-qualifications (16), -- Private group read-private-use (17) }	
file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) unstructured(1) } "FTAM unstructured constraint set"
file contents	Datatype1 ::= FileDirectoryEntry -- as defined by NBS-AS2 in 8.1

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.1.3 Scope and field of application

The document type defines the contents of a file for transfer (not for storage) using FTAM.

6.1.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.1.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.1.6 Abbreviations

FTAM	File Transfer, Access and Management
NBS	National Bureau of Standards, USA

6.1.7 Document semantics

The document consists of one file access data unit, which consists only of zero, one or more data elements of type "FileDirectoryEntry" (defined in NBS-AS2).

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the unstructured constraint set. These definitions appear in ISO 8571-2.

The parameter of the document type is used on F-OPEN request to specify the desired attributes of each of the files in the filestore, when reading the document.

6.1.8 Abstract syntactic structure

The abstract syntactic structure of the document is a series of file directory entries, each of which is defined by the "FileDirectoryEntry" definition in NBS-AS2.

Additional constraints are defined for this document type: file access actions are restricted to read. File-directory files may not be written or modified (except as a side effect of actions performed on individual files contained within a file directory).

6.1.9 Definition of transfer

6.1.9.1 Datatype definition

The file consists of zero or more values of Datatype1 defined in table 1.

6.1.9.2 Presentation data values

The document is transferred as a series of presentation data values. Each presentation data value shall consist of one value of the ASN.1 data type "Datatype1", carrying one of the file directory entries from the document.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 1.

6.1.9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of file directory entries within the data unit in the file.

6.1.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 1 for all presentation data values transferred. Implementations may optionally support other named transfer syntaxes.

6.1.11 ASE specific specifications for FTAM

Relaxation is allowed to any bitstring combination of the document type parameter.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.2 INTAP-1 Record file document type

6.2.1 Entry number : INTAP-1

6.2.2 Information objects

Table 2 - Information objects in INTAP-1

document type name	{ iso member-body 392 ftam(10) document-type(2) intap-record-file(1) } "INTAP record file"
abstract syntax names name for asname1	{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as1(1) } "INTAP abstract syntax AS1"
transfer syntax names	{ iso member-body 392 ftam(10) transfer-syntax(4) intap-ts1(1) } "INTAP transfer syntax TS1" { joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax PARAMETERS : := SEQUENCE { maximum-record-length [1] IMPLICIT INTEGER OPTIONAL, record-significance [2] IMPLICIT INTEGER {variable (0), fixed (1) } OPTIONAL }	
file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) unstructured(1) } "FTAM unstructured constraint set"
file contents Datatype1 ::= Record-Element -- as defined in 8.2	

6.2.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

6.2.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.2.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.2.5.1 record : an ordered series of one or more record-elements. Data units of this document type consist of one or more records (see figure 1).

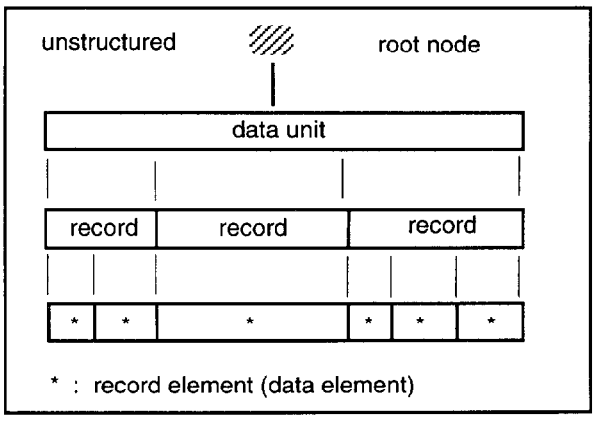


Figure 1 - Structure of INTAP-1 document type

6.2.6 Abbreviations

FADU	File Access Data Unit
FTAM	File Transfer, Access and Management
INTAP	Interoperability Technology Association for Information Processing, Japan

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.2.7 Document semantics

The document consists of one file access data unit, which consists only of zero, one or more records. The order of each of these records is significant.

The document structure takes the form allowed by the FTAM hierarchical file model as constrained by the unstructured constraint set (see table 2). These definitions appear in ISO 8571-2.

Each record consists of octets of any value from 0 to 255. The meaning attached to these values is not constrained by the document type.

There are no size or length limitations imposed by this definition, except those specified here. Each record is of a length determined by the number of octets given by the maximum-record-length parameter. If this parameter is not present, the default is that the length of records is unbounded. If the value of the record-significance parameter is variable, or if the parameter is not present, the length of each record is less than or equal to the length given in the maximum-record-length parameter. If the value is fixed, the length of each record is exactly equal to the length given.

6.2.8 Abstract syntactic structure

The abstract syntactic structure of the document is a series of octet strings (record elements). The grouping of record elements into records is indicated by the context-specific-number field of each record-element as specified in 8.2.

6.2.9 Definition of transfer

6.2.9.1 Datatype definition

The file consists of zero or more values of Datatype1 (see table 2 for the definition).

6.2.9.2 Presentation data values

The document is transferred as a series of presentation data values. Each presentation data value shall consist of one value of ASN.1 data type "Datatype1", carrying one of the record-elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1".

NOTE - Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

6.2.9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of record-elements within the data unit in the file.

6.2.10 Transfer syntax

For transferred presentation data values there are two types of transfer syntax names for INTAP-1 specified in table 2. They are INTAP transfer syntax TS1 and Basic Encoding of a single ASN.1 type. The former is used for compression of data, and the latter is used for basic encoding (non-compression) of data. Implementations supporting this document type shall support the syntax for the basic encoding and may optionally support the syntax for compression of data. Implementations may optionally support other named transfer syntaxes.

6.2.11 ASE specific specifications

6.2.11.1 String length relaxation

String length relaxation loses explicit information in the document type identification. A document of type INTAP-1 may be relaxed to another document of type INTAP-1 with a larger maximum-record-length parameter, or no maximum-record-length parameter.

6.2.11.2 The EXTEND operation

When the EXTEND operation is applied to a data unit of an INTAP-1 document, the transferred data shall be an INTAP-1 document with parameters equal to those of the original INTAP-1 document. The resulting document consists of the record-elements of the original INTAP-1 document, followed by the record-elements of the new INTAP-1 document. The boundary of the original and new record-elements is not visible in the new document.

6.2.11.3 The REPLACE operation

When the REPLACE operation is applied to the root FADU of an INTAP-1 document, the transferred material shall be any INTAP-1 document with the same parameter values.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.3 NBS-6 Sequential file document type

6.3.1 Entry number : NBS-6

6.3.2 Information objects

Table 3 - Information objects in NBS-6

document type name	{ iso identified-organization oiw(14) ftamsig(5) document-type(5) sequential(6) } "NBS-6 FTAM sequential file"
abstract syntax names	
a) name for asname1	{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) } "NBS abstract syntax AS1"
b) name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

parameter syntax

```
PARAMETERS      ::=  SEQUENCE OF CHOICE { parameter0 Parameter0,  parameter1 Parameter1,
                                          parameter2 Parameter2 }

Parameter0      ::=  [0] INTEGER {   univer-time   (23),
                                     gen-time      (24),
                                     boolean       (1),
                                     null          (5)    }

Parameter1      ::=  [1] SEQUENCE    {
                     universal-class-number-1   INTEGER    { int        (2),
                                                             bit        (3),
                                                             ia5        (22),
                                                             graphic    (25),
                                                             general    (27),
                                                             octet      (4)    },
                     string-length              INTEGER  }

Parameter2      ::=  [2] SEQUENCE    {
                     private-class-number       INTEGER    { float(0) },
                     length-1                   INTEGER,
                     length-2                   INTEGER }
```

file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) sequential-flat(2) } "FTAM sequential flat constraint set"
file contents	Datatype1 ::= PrimType --as defined by NBS abstract syntax AS1 in 8.3 Datatype2 ::= Node-Descriptor-Data-Element

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.3.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

NOTE - Storage refers to apparent storage within the virtual filestore.

6.3.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.3.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.3.6 Abbreviations

FTAM File Transfer, Access and Management

6.3.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely zero or one data unit which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the sequential flat constraint set (see table 3). These definitions appear in ISO 8571-2. As additional constraints FADU Identity will be limited to begin, end, first and next.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in 8.3. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

For Datatype1, the string-length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

6.3.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571-2, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

6.3.9 Definition of transfer

6.3.9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 3, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 3, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

6.3.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1"; or

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.3.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.3.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 3 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.3.11 ASE specific specifications

6.3.11.1 Structural simplification

This simplification loses information.

The document type NBS-6 may be simplified to the document type FTAM-3 (allowed only when reading the file). The octet representation of the transferred data is

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

6.3.11.2 Access context selection

A document of type NBS-6 may be accessed in any one of the access contexts defined in the sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.3.11.3 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any NBS-6 document with the same parameter values in access context FA.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.4 NBS-7 Random access file document type

6.4.1 Entry number : NBS-7

6.4.2 Information objects

Table 4 - Information objects in NBS-7

document type name	{ iso identified-organization oiw(14) ftamsig(5) document-type(5) random-file(7) } "NBS-7 FTAM random acess file"
abstract syntax names	
a) name for asname1	{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) } "NBS abstract syntax AS1"
b) name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

parameter syntax

```
PARAMETERS      ::=  SEQUENCE OF CHOICE { parameter0 Parameter0,  parameter1 Parameter1,
                                          parameter2 Parameter2 }

Parameter0      ::=  [0] INTEGER {      univer-time     (23),
                                        gen-time        (24),
                                        boolean         (1),
                                        null            (5)  }

Parameter1      ::=  [1] SEQUENCE       {
                     universal-class-number-1   INTEGER     { int           (2),
                                                              bit           (3),
                                                              ia5           (22),
                                                              graphic       (25),
                                                              general       (27),
                                                              octet         (4)  },
                     string-length              INTEGER  }

Parameter2      ::=  [2] SEQUENCE       {
                     private-class-number       INTEGER     { float (0) },
                     length-1                   INTEGER,
                     length-2                   INTEGER }
```

file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-ordered-flat(1) } "NBS ordered flat constraint set"

file contents

```
Datatype1   ::= PrimType    --as defined by NBS abstract syntax AS1 in 8.3
Datatype2   ::= CHOICE {    node-descriptor-data-element  Node-Descriptor-Data-Element,
                            enter-subtree-data-element  Enter-Subtree-Data-Element,
                            exit-subtree-data-element  Exit-Subtree-Data-Element }
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.4.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

NOTE - Storage refers to apparent storage within the virtual filestore.

6.4.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.4.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.4.6 Abbreviations

FTAM File Transfer, Access and Management

6.4.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely zero or one data unit which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the NBS-ordered-flat constraint set (see table 4). These definitions appear in 7.1.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in 8.3. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

For Datatype1, the string-length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

6.4.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571-2, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

6.4.9 Definition of transfer

6.4.9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 4, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 4, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

6.4.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1"; or

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.4.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.4.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 4 for all presentation data values transferred.

Implementation may optionally support other named transfer syntaxes.

6.4.11 ASE specific specifications

6.4.11.1 Structural simplification

This simplification loses information.

The document type NBS-7 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

type parameter in F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type NBS-7 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the contents type parameter on the F-OPEN request.

6.4.11.2 Access context selection

A document of type NBS-7 may be accessed in any one of the access contexts defined in the NBS ordered flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.4.11.3 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any NBS-7 document with the same parameter values in access context FA.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.5 NBS-8 Indexed sequential file document type

6.5.1 Entry number : NBS-8

6.5.2 Information objects

Table 5 - Information objects in NBS-8

document type name			{ iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file(8) } "NBS-8 FTAM indexed file"
abstract syntax names			
a) name for asname1			{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) } "NBS abstract syntax AS1"
b) name for asname2			{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names			{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax			
PARAMETERS	: :=	SEQUENCE { datatypes DataTypes, keytype KeyType, keyposition KeyPosition }	
DataTypes	: :=	SEQUENCE OF CHOICE { parameter0 Parameter0, parameter1 Parameter1, parameter2 Parameter2 }	
KeyType	::=	CHOICE { parameter0 Parameter0, parameter1 Parameter1, parameter2 Parameter2 } -- Parameter0, Parameter1, Parameter2 as defined for the document types NBS-6 -- and NBS-7	
KeyPosition	: :=	INTEGER	
file model			{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set			{ iso standard 8571 constraint-set(4) ordered-flat(3) } "FTAM ordered flat constraint set"
file contents	Datatype1	::= PrimType	--as defined by NBS abstract syntax AS1 in 8.3
	Datatype2	::= CHOICE {	node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element }
	Datatype3	::= PrimType	--as defined by NBS abstract syntax AS1 in 8.3

6.5.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access using FTAM.

NOTE - Storage refers to apparent storage within the virtual filestore.

6.5.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.5.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.5.6 Abbreviations

FTAM File Transfer, Access and Management

6.5.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely one data unit which consists of zero, one or more data elements. The order of each of these elements is significant.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set (see table 5). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set:

- The FADU Identities first, last, and node number are not required for conformant implementations;
- The identities next and previous are allowed for all FADUs.

Each data element is a data type from the set of primitive data types defined in 8.3. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the DataTypes parameter.

For Datatype1 and Datatype3, the string-length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, in Datatype1 and Datatype3, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

Each data unit in the file has a key associated with it, which is the user-coded form of Node-Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in 8.3.

The type and length of the key are defined by the KeyType parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in table 6.

The position of the key in the data unit is specified by the KeyPosition parameter:

- KeyPosition = 0 implies the key is not part of the data;
- KeyPosition > 0 specifies the actual data element in the data unit.

Table 6 - Datatypes for keys

Key Type	Minimum Range (octets)	Order
ASN.1 INTEGER	(1-2)	increasing numeric value
ASN.1 IA5String	(1-16)	lexical order
ASN.1 GraphicString	(1-16)	lexical order
ASN.1 GeneralString	(1-16)	lexical order
ASN.1 OCTET STRING	(1-16)	increasing value
ASN.1 GeneralizedTime		increasing time value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing numeric value

6.5.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571-2, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

6.5.9 Definition of transfer

6.5.9.1 Datatype definitions

The file consists of data values which are of

a) Datatype1 defined in table 5, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 5, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3, defined in table 5, which specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where user-coded is defined as an EXTERNAL.

6.5.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1"; or

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2"; or

c) a value of "Datatype3" carrying a key. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.5.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.5.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 5 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.5.11 ASE specific specifications

6.5.11.1 Structural simplification

This simplification loses information.

The document type NBS-8 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents type parameter in F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type NBS-8 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the contents type parameter on the F-OPEN request. The traversal order of the FADUs must be maintained.

NOTE - The traversal order is as reading the file as NBS-8 in key order.

6.5.11.2 Access context selection

A document of type NBS-8 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.5.11.3 The INSERT operation

When the INSERT operation is applied, the transferred material shall be the series of FADUs which would be generated by reading any NBS-8 document with the same parameter values in access context FA.

The insertion of a new FADU after an already existing FADU will be indicated via a diagnostic on F-TRANSFER-END.

6.5.11.4 The EXTEND operation

This operation is excluded for use with this document type.

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.6 NBS-10 Random binary access file document type

6.6.1 Entry number : NBS-10

6.6.2 Information objects

Table 7 - Information objects in NBS-10

document type name		{ iso identified-organization oiw(14) ftamsig(5) document-type(5) random-binary(10) } "NBS-10 FTAM random binary access file"
abstract syntax names		
a) name for asname1		{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-random-binary(4) } "NBS random binary access file abstract syntax"
b) name for asname2		{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
c) name for asname3		{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-node-name(3) } "NBS random access node name abstract syntax"
transfer syntax names		{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
file model		{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set		{ iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-random-access(2) } "NBS random access constraint set"
file contents	Datatype1	::= OCTET STRING
	Datatype2	::= Node-Name -- The type to be used for Node-Name is defined in ISO8571-FADU -- The only choice for Node-Name is user-coded.
	Datatype3	::= NBS-Node-Name -- as defined by NBS random access node name -- abstract syntax in 8.4

6.6.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

6.6.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.6.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.6.6 Abbreviations

FTAM File Transfer, Access and Management

6.6.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely one data unit which consists of precisely one data element. The data element is made up of one octet. The order of each of these elements is significant. The semantics of the data element is not specified by this document type.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the NBS random access constraint set. The definition for FTAM hierarchical file model appears in ISO 8571-2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

There are no size or length limitations imposed by this definition.

6.6.8 Abstract syntactic structure

The abstract syntactic structure of the document is a series of octets.

6.6.9 Definition of transfer

6.6.9.1 Datatype definition

A presentation data value used for transfer is an ASN.1 OCTET STRING.

Datatype2 is used to specify the FADU Identity of name-list in the FTAM PDUs specifying FADU Identity, where name-list is defined as a SEQUENCE OF EXTERNAL. The EXTERNAL is defined as Node-Name in the FTAM FADU abstract syntax. The use of Datatype2 is defined in 7.2 (NBS random access constraint set).

Datatype3 specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where user-coded is defined as an EXTERNAL. That EXTERNAL is defined by Datatype3. The use of Datatype3 is defined in 7.2 (NBS random access constraint set).

6.6.9.2 Presentation data values

The document is transferred as a series of presentation data values. Each presentation data value shall consist of the data from one or more FADUs concatenated together. The result is one value of the ASN.1 data type OCTET STRING. The fadu-count field supplied in the Node-Name specifies the number of FADUs to transfer during a Read operation. The requested FADUs may be transferred as one or more presentation data values.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 7.

NOTE - Specific carrier standards may impose additional constraints on the presentation context to be used, when the above permits a choice.

Boundaries between P-DATA primitives and between presentation data values are chosen locally by the sending entity at the time of transmission. The boundaries are not preserved when the file is stored and they carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

6.6.9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of data units in the file.

6.6.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 7 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.6.11 ASE specific specifications

6.6.11.1 Simplification

The document type NBS-10 may be simplified to the document type FTAM-3. The resultant document contains the same sequence of data values as would result from accessing the file as an NBS-10 file.

6.6.11.2 The READ operation

A READ operation may be applied to a range of FADUs via the FADU Identity of NodeSeq. The starting-fadu part of the node name specifies the node number of the first FADU; the fadu-count specifies the number of consecutive FADUs to be transferred.

A READ operation applied to a range of FADUs that spans beyond the end of file is valid. All available data in the range is transferred. An informative diagnostic (5005) is returned on the F-DATA-END request indicating that the end of file was reached and a portion of the request was satisfied.

6.6.11.3 The REPLACE operation

When the REPLACE operation is applied to the root FADU of an NBS-10 document, the transferred data shall be any NBS-10 document.

The REPLACE operation applied to a FADU Identity of node number is used to replace a series of FADUs, starting at the specified position in the file, by the new FADUs being transferred. The number of replaced FADUs is determined by the number of transferred FADUs.

If the replacement spans beyond the end of the existing file, then the additional FADUs are inserted at the end of the file.

6.6.11.4 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred data shall be the series of FADUs which would be generated by reading any NBS-10 document in access context UA.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.7 NBS-11 Indexed file with unique keys document type

6.7.1 Entry number : NBS-11

6.7.2 Information objects

Table 8 - Information objects in NBS-11

document type name	{ iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file-with-unique-keys(11) } "NBS-11 FTAM indexed file with unique keys"
abstract syntax names	
a) name for asname1	{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) } "NBS abstract syntax AS1"
b) name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax	
PARAMETERS ::=	SEQUENCE { datatypes DataTypes, keytype KeyType, keyposition KeyPosition }
DataTypes ::=	SEQUENCE OF CHOICE { parameter0 Parameter0, parameter1 Parameter1, parameter2 Parameter2 }
KeyType ::=	CHOICE { parameter0 Parameter0, parameter1 Parameter1, parameter2 Parameter2 } -- Parameter0, Parameter1, Parameter2 as defined for the document types NBS-6 -- and NBS-7
KeyPosition ::=	INTEGER
file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) ordered-flat-unique-names(4) } "FTAM ordered flat constraint set with unique names"
file contents	Datatype1 ::= PrimType --as defined by NBS abstract syntax AS1 in 8.3 Datatype2 ::= CHOICE { node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element } Datatype3 ::= PrimType --as defined by NBS abstract syntax AS1 in 8.3

6.7.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access using FTAM.

NOTE - Storage refers to apparent storage within the virtual filestore.

6.7.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

6.7.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.7.6 Abbreviations

FTAM File Transfer, Access and Management

6.7.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely one data unit which consists of zero, one or more data elements. The order of each of these elements is significant.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see table 8). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set with unique names:

- o The FADU Identity node number is not required for conformant implementations;
- o The identities next and previous are allowed for all FADUs.

Each data element is a data type from the set of primitive data types defined in 8.3. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the DataTypes parameter.

For Datatype1 and Datatype3, the string-length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, in Datatype1 and Datatype3, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

Each data unit in the file has a key associated with it, which is the user-coded form of Node-Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in 8.3.

The type and length of the key are defined by the KeyType parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in table 9.

The position of the key in the data unit is specified by the KeyPosition parameter:

- o KeyPosition = 0 implies the key is not part of the data;
- o KeyPosition > 0 specifies the actual data element in the data unit.

Table 9 - Datatypes for keys

Key Type	Minimum Range (octets)	Order
ASN.1 INTEGER	(1-2)	increasing numeric value
ASN.1 IA5String	(1-16)	lexical order
ASN.1 GraphicString	(1-16)	lexical order
ASN.1 GeneralString	(1-16)	lexical order
ASN.1 OCTET STRING	(1-16)	increasing value
ASN.1 GeneralizedTime		increasing time value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing numeric value

6.7.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571-2, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

6.7.9 Definition of transfer

6.7.9.1 Datatype definitions

The file consists of data values which are of

a) Datatype1 defined in table 8, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 8, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3, defined in table 8, which specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where user-coded is defined as an EXTERNAL.

6.7.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1"; or

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2"; or

c) a value of "Datatype3" carrying a key. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.7.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.7.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 8 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.7.11 ASE specific specifications

6.7.11.1 Structural simplification

This simplification loses information.

The document type NBS-11 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents type parameter in F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type NBS-11 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the contents type parameter on the F-OPEN request. The traversal order of the FADUs must be maintained.

NOTE - The traversal order is as reading the file as NBS-11 in key order.

A document of type NBS-11 may be accessed as a document of type NBS-8 (allowed only when reading the file) by specifying document type NBS-8 in the contents type parameter in the F-OPEN request.

6.7.11.2 Access context selection

A document of type NBS-11 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.7.11.3 The INSERT operation

When the INSERT operation is applied, the transferred material shall be the series of FADUs which would be generated by reading any NBS-11 document with the same parameter values in access context FA.

A transferred FADU whose name duplicates that of an already existing FADU will cause the INSERT operation to fail. The failure shall be signalled by using an F-CANCEL request with a corresponding diagnostic.

6.7.11.4 The EXTEND operation

This operation is excluded for use with this document type.

6.7.11.5 The REPLACE operation

When the replace operation is applied with FADU Identity begin, a transferred FADU whose name duplicates that of a previously tranferred FADU will cause the REPLACE operation to fail. The failure shall be signalled by issuing an F-CANCEL request with a corresponding diagnostic.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.8 NBS-12 Simple text file document type

6.8.1 Entry number : NBS-12

6.8.2 Information objects

Table 10 - Information objects in NBS-12

document type name	{ iso identified-organization oiw(14) ftamsig(5) document-type(5) simple-text-file(12) } "NBS-12 FTAM simple text file"
abstract syntax names	
a) name for asname1	{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-simple-text(5) } "NBS simple text abstract syntax"
b) name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

parameter syntax
PARAMETERS ::= SEQUENCE {
universal-class-number [0] IMPLICIT INTEGER,
maximum-string-length [1] IMPLICIT INTEGER,
string-significance [2] IMPLICIT INTEGER { variable (0), fixed (1) },
character-set [3] IMPLICIT OCTET STRING OPTIONAL
}	

file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) sequential-flat(2) } "FTAM sequential flat constraint set"
file contents	Datatype1 ::= NBS-Text -- as defined by NBS simple text abstract syntax in 8.6 Datatype2 ::= Node-Descriptor-Data-Element

6.8.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

NOTE - Storage refers to apparent storage within the virtual filestore.

6.8.4 References

ISO/IEC 6429:1992, *Information technology - Control functions for coded character sets.*

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

ISO/IEC 8824-1:1995, *Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

ISO/IEC 8825-1:1995, *Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

6.8.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1. In

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

addition, it makes use of the terms character string, graphics character and format effector as defined in document type registration entry FTAM-2 in ISO 8571-2.

6.8.6 Abbreviations

FTAM File Transfer, Access and Management

6.8.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU consists of precisely one data unit which consists of precisely one character string. The order of each of these elements is significant. The semantics of the character strings is not specified by this document type.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM sequential flat constraint set. These definitions appear in ISO 8571-2. As additional constraints, FADU Identity will be limited to the following values:

a) begin and end when using the transfer or transfer and management service classes;

b) begin, end, first and next when using the access service class.

Each character string consists of characters from the character set defined by the ASN.1 (ISO/IEC 8824-1) character set type whose universal class number is given by the universal-class-number parameter and by the escape sequences contained in the optional character-set parameter. If the character set type allows explicit escape sequences, the character-set parameter, if present, contains escape sequences which designate and invoke specific character sets. If the character-set parameter is not present, character sets are assumed to be designated and invoked as specified in table 2 in ISO/IEC 8825-1. Character strings shall not contain escape sequences.

There are no size or length limitations imposed by this definition, except those specified here. Each character string is of a length determined by the number of characters given by the maximum-string-length parameter.

NOTE - The length restriction refers to the number of characters from the applicable character set, not to the number of octets in the encoding, nor to the line length in any rendition of the document, where these are different.

The exact significance of the character strings is determined by the string-significance parameter. If its value is variable, the length of the character strings is less than or equal to the length given. If the value is fixed, the length of each character string is exactly equal to the length given.

If the document is interpreted on a character imaging device (outside the scope of ISO 8571), the interpretation depends on the character set in use:

a) If the character set contains format effectors, they shall be interpreted as defined in ISO/IEC 6429; end of string and end of file access data unit are given no formatting significance, and do not contribute to the document semantics;

b) If the character set does not contain format effectors, the end of each character string is interpreted as implying carriage return and line feed actions in any rendition. The end of file access data unit is given no formatting significance beyond that attached to the end of string in it.

6.8.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU and ISO8571-CONTENTS in ISO 8571-2, in which each of the file contents data elements has the abstract syntactic structure of NBS-Text.

6.8.9 Definition of transfer

6.8.9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 10, the ASN.1 datatype declared as "NBS-Text" in the NBS simple text abstract syntax definition. The choice in "NBS-Text" is determined by the universal-class-number parameter; or

b) Datatype2 defined in table 10, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

6.8.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the character strings of the document. Each character shall be transmitted using one of the character sets identified by the universal-class-number parameter. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 10, or

b) one value of the ASN.1 datatype "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2" declared in table 10.

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between P-DATA primitives are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

6.8.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of character strings within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.8.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 10 for all presentation data values transferred.

6.8.11 ASE specific specifications

6.8.11.1 Simplification and relaxation

6.8.11.1.1 Simplification to FTAM-1

This simplification loses information.

The document type NBS-12 may be accessed as a document type FTAM-1. The resultant document contains the same sequence of data values as would result from accessing the structured text file in access context UA. That is, only the presentation data values in the abstract syntax "asname1" are present. If the character-set parameter was present before the simplification, its contents will be added to the beginning of each string.

NOTE - The boundary between file access data units remains a boundary between strings, but any special significance given to it is lost.

6.8.11.1.2 Relaxation to FTAM-2

The document type NBS-12 may be relaxed to the document type FTAM-2. If the character-set parameter was present before the relaxation, its contents will be added to the beginning of each string.

6.8.11.1.3 Character set relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to a different document of type NBS-12 with

- a different universal-class-number parameter value;
- a different character-set parameter value;
- different values for both of these parameters;
- a different universal-class-number parameter value and no character-set parameter value; or
- no character-set parameter value;

if the resultant document type permits all characters from the original document type. If this relaxation involves including format effectors and none were present before the relaxation, the characters carriage return and line feed shall be added to the end of each string.

NOTE - If the characters carriage return and line feed are not part of the format effectors, the formatting action may be represented by newline or some other implementation specific choice if there is no representation of newline defined.

6.8.11.1.4 String length relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to another document of type NBS-12 with a larger maximum-string-length parameter.

6.8.11.2 Access context selection

A document of type NBS-12 may be accessed in any one of the access contexts defined in the FTAM sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.8.11.3 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any NBS-12 document with the same parameter values in access context FA.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.9 INTAP-2 Sequential file document type

6.9.1 Entry number : INTAP-2

6.9.2 Information objects

Table 11 - Information objects in INTAP-2

document type name	{ iso member-body 392 ftam(10) document-type(2) sequential-file(2) } "INTAP-2 sequential file"
abstract syntax names name for asname1 name for asname2	 { iso member-body 392 ftam(10) abstract-syntax(3) intap-as2(2) } "INTAP abstract syntax AS2" { iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

parameter syntax
PARAMETERS : := SEQUENCE {
record-significance [0] IMPLICIT INTEGER { variable (0), fixed (1) } OPTIONAL,
parameters SEQUENCE OF CHOICE { parameter1 Parameter1, parameter2 Parameter2 } }
Parameter1 ::= [1] IMPLICIT SEQUENCE {
universal-class-number-1 INTEGER { int (2),
ia5 (22), graphic (25),
general (27), octet (4) },
maximum-string-length INTEGER }
Parameter2 ::= [2] IMPLICIT SEQUENCE {
private-class-number INTEGER {
decimal-type-unsign (0), decimal-type-sign (1),
numeric-type-unsign (2), numeric-type-trailing (3),
numeric-type-trailing-with-separate-character (4),
numeric-type-leading (5),
numeric-type-leading-with-separate-character (6) },
length-1 INTEGER,
maximum-string-length INTEGER }	

file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso member-body 392 ftam(10) constraint-set(5) sequential-flat(1) } "INTAP sequential flat constraint set"
file contents	Datatype1 ::= PrimType2 --as defined by INTAP-AS2 in 8.7 Datatype2 ::= CHOICE { node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element }

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.9.3 Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM. This document type defines the COBOL-like access structure of a sequential file.

NOTE - Storage refers to the apparent storage within the virtual filestore.

6.9.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3: File Service Definition.*

6.9.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.9.6 Abbreviations

FTAM File Transfer, Access and Management

6.9.7 Document semantics

The document consists of zero, one or more file access data units and each FADU consists of zero or one data unit, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the INTAP sequential flat constraint set (see table 11). These definitions appear in ISO 8571-2. As additional constraints FADU Identity will be limited to begin, end, first, current and next.

One record corresponds to one data unit. The exact significance of the record is determined by the record-significance parameter. If its value is variable, the length of the record is less than or equal to the sum of each maximum-string-length given by parameter1 and parameter2. If the value is fixed, or the parameter is not present, the length of each record is exactly equal to the sum of each maximum-string-length given by parameter1 and parameter2.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in 8.7. Each data unit contains the same data element types in the same order as all other data units. These types are determined by parameter1 and parameter2.

The maximum-string-length field of Parameter1 specifies the length of the value in octets for the INTEGER and OCTET STRING types. For character-type data elements, the maximum-string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For decimal-type or numeric-type numbers, length-1 specifies the precision. The precision of an exact numeric literal is the number of digits that it contains.

The maximum-string-length field of parameter2 specifies the length of the value in octets for DecimalType and NumericType.

6.9.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of INTAP-AS2 as defined by the parameters.

6.9.9 Definition of transfer

6.9.9.1 Datatype definition

The file consists of data values which are of either

a) Datatype1 defined in table 11, where the PrimType2 in the datatype is given by the INTAP-AS2 definition; or

b) Datatype2 defined in table 11, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

6.9.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1"; or

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.9.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.9.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 11 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.9.11 ASE specific specifications for FTAM

6.9.11.1 Structural simplification

This simplification loses information.

A document type INTAP-2 may be simplified to the document type FTAM-3 (allowed only when reading the file). The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

6.9.11.2 Access context selection

A document of type INTAP-2 may be accessed in any one of the access contexts defined in the INTAP sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.9.11.3 The READ operation

When the READ operation is applied at last of file with an FADU Identity next, the responder transfers the error identifier which contains "5005" by F-DATA-END.

6.9.11.4 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any INTAP-2 document with the same parameter values in access context FA.

6.9.11.5 The REPLACE operation

The REPLACE operation applied to an FADU Identity of current is used to replace an FADU with the new FADU being transferred. The length of the new FADU shall be the same as the existing FADU.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.10 INTAP-3 Relative file document type

6.10.1 Entry number : INTAP-3

6.10.2 Information objects

Table 12 - Information objects in INTAP-3

document type name		{ iso member-body 392 ftam(10) document-type(2) relative-file(3) } "INTAP-3 relative file"
abstract syntax names		
name for asname1		{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as2(2) } "INTAP abstract syntax AS2"
name for asname2		{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
name for asname3		{ iso member-body 392 ftam(10) abstract-syntax(3) intap-node-name(3) } "INTAP node name abstract syntax"
transfer syntax names		{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax		
PARAMETERS ::=	SEQUENCE { datatypes DataTypes, numberlength NumberLength }	
DataTypes ::=	SEQUENCE { record-significance [0] IMPLICIT INTEGER { variable (0), fixed (1) } OPTIONAL, parameters SEQUENCE OF CHOICE { parameter1 Parameter1, parameter2 Parameter2 } } -- Parameter1 and Parameter2 as defined for the document type INTAP-2	
NumberLength ::=	INTEGER	
file model		{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set		{ iso standard 8571 constraint-set(4) ordered-flat-unique-names(4) } "FTAM ordered flat constraint set with unique names"
file contents	Datatype1 ::= PrimType2	--as defined by INTAP-AS2 in 8.7
	Datatype2 ::= CHOICE {	node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element }
	Datatype3 ::= INTAP-Node-Name	--as defined by INTAP-AS3 in 8.8

6.10.3 Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM. This document type defines the COBOL-like access structure of a relative file.

NOTE - Storage refers to the apparent storage within the virtual filestore.

6.10.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3: File Service Definition.*

6.10.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.10.6 Abbreviations

FTAM File Transfer, Access and Management

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.10.7 Document semantics

The document consists of zero, one or more file access data units and each FADU consists of zero or one data unit, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see table 12). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the FTAM ordered flat constraint set with unique names:

- The FADU Identity traversal is not required for conformant implementations;
- An INTAP node name is used to get a record with an access condition. An INTAP node name which consists of an access key and an access condition is carried by an FADU Identity of a primitive parameter.

One record corresponds to one data unit. The exact significance of the record is determined by the record-significance parameter. If its value is variable, the length of the record is less than or equal to the sum of each maximum-string-length given by parameter1 and parameter2. If the value is fixed, or the parameter is not present, the length of each record is exactly equal to the sum of each maximum-string-length given by parameter1 and parameter2.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in 8.7. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the DataTypes parameter.

The maximum-string-length field of Parameter1 specifies the length of the value in octets for the INTEGER and OCTET STRING types. For character-type data elements, the maximum-string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For decimal-type or numeric-type numbers, length-1 specifies the precision. The precision of an exact numeric literal is the number of digits that it contains.

The maximum-string-length field of parameter2 specifies the length of the value in octets for DecimalType and NumericType.

Each data unit in the file has a relative record number associated with it. The relative record number of each data unit is the INTEGER type defined in 8.7. The INTEGER does not contain zero. The maximum length of the relative record number is defined by the NumberLength parameter. The position of the relative record number is in the node name. The access condition in the INTAP-Node-name is specified as only equal-to on the relative file.

6.10.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of INTAP-AS2 as defined by the parameters.

6.10.9 Definition of transfer

6.10.9.1 Datatype definition

The file consists of data values which are of

a) Datatype1 defined in table 12, where the PrimType2 in the datatype is given by the INTAP-AS2 definition;

b) Datatype2 defined in table 12, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3 specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where "user-coded" is defined as an EXTERNAL. That EXTERNAL is defined by the INTAP-AS3 definition.

6.10.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1";

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2"; or

c) a value of "Datatype3". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname3".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.10.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file mode in ISO 8571-2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.10.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 12 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.10.11 ASE specific specifications for FTAM

6.10.11.1 Structural simplification

This simplification loses information.

A document type INTAP-3 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents-type parameter on the F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type INTAP-3 can be accessed as a document of type INTAP-2 (allowed only when reading the file) by specifying document type INTAP-2 with appropriate data type parameters in the contents-type parameter on the F-OPEN request.

6.10.11.2 Access context selection

A document of type INTAP-3 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.10.11.3 The READ operation

When the READ operation is applied at last of file with an FADU Identity next, the responder transfers the error identifier which contains "5005" by F-DATA-END.

6.10.11.4 The INSERT operation

When the INSERT operation is applied at the begin of file, the transferred material shall be the series of FADUs which would be generated by reading any INTAP-3 document with the same parameter values in access context FA.

The relative record number must be at least one greater than the number of the last existing node.

An error will occur if the relative record number is not greater than that of the last existing node.

6.10.11.5 The EXTEND operation

This operation is excluded for use with this document type.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.11 INTAP-4 Indexed file document type

6.11.1 Entry number : INTAP-4

6.11.2 Information objects

Table 13 - Information objects in INTAP-4

document type name	{ iso member-body 392 ftam(10) document-type(2) indexed-file(4) } "INTAP-4 indexed file"
abstract syntax names	
name for asname1	{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as2(2) } "INTAP abstract syntax AS2"
name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
name for asname3	{ iso member-body 392 ftam(10) abstract-syntax(3) intap-node-name(3) } "INTAP node name abstract syntax"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

parameter syntax

```
PARAMETERS  ::=  SEQUENCE  { datatypes DataTypes, keytype KeyType, keyposition KeyPosition }
DataTypes   ::=  SEQUENCE  {
                 record-significance [0] IMPLICIT INTEGER { variable (0), fixed (1) } OPTIONAL,
                 parameters SEQUENCE OF CHOICE { parameter1 Parameter1, parameter2 Parameter2 } }
KeyType     ::=  CHOICE { parameter1 Parameter1, parameter2 Parameter2 }
                 -- Parameter1, Parameter2 as defined for the document type INTAP-2
KeyPosition ::=  CHOICE { fieldposition FieldPosition, offsetposition OffsetPosition }
FieldPosition ::= INTEGER
OffsetPosition ::= SEQUENCE { offset INTEGER, length INTEGER }
```

file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso standard 8571 constraint-set(4) ordered-flat(3) } "FTAM ordered flat constraint set"

file contents

```
Datatype1  ::= PrimType2                    --as defined by INTAP-AS2 in 8.7
Datatype2  ::= CHOICE {  node-descriptor-data-element Node-Descriptor-Data-Element,
                         enter-subtree-data-element Enter-Subtree-Data-Element,
                         exit-subtree-data-element Exit-Subtree-Data-Element }
Datatype3  ::= INTAP-Node-Name              --as defined by INTAP-AS3 in 8.8
```

6.11.3 Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM. This document type defines the COBOL-like access structure of an indexed file which is accessed using secondary keys.

NOTE - Storage refers to the apparent storage within the virtual filestore.

6.11.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3: File Service Definition.*

6.11.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.11.6 Abbreviations

FTAM File Transfer, Access and Management

6.11.7 Document semantics

The document consists of zero, one or more file access data units and each FADU consists of zero or one data unit, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set (see table 13). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the FTAM ordered flat constraint set:

- The FADU Identities end, first, last, previous, and traversal are not required for conformant implementations;
- An INTAP node name is used to get a record with an access condition. An INTAP node name which consists of an access key and an access condition is carried by an FADU Identity of a primitive parameter.
- The allowable file actions are the LOCATE and the READ operation.

One record corresponds to one data unit. The exact significance of the record is determined by the record-significance parameter. If its value is variable, the length of the record is less than or equal to the sum of each maximum-string-length given by parameter1 and parameter2. If the value is fixed, or the parameter is not present, the length of each record is exactly equal to the sum of each maximum-string-length given by the parameter1 and parameter2.

Each data element is a data type from the set of primitive data types defined in 8.7. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the DataTypes parameter.

The maximum-string-length field of parameter1 specifies the length of the value in octets for the INTEGER and OCTET STRING types. For character-type data elements, the maximum-string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For decimal-type or numeric-type numbers, length-1 specifies the precision. The precision of an exact numeric literal is the number of digits that it contains.

The maximum-string-length field of parameter2 specifies the length of the value in octets for DecimalType and NumericType.

Each data unit in the file has a key associated with it, which is the user-coded form of Node Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in 8.7.

The type and length of the key are defined by the KeyType parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in table 14.

The position of the key in the data unit is specified by the KeyPosition parameter:

- FieldPosition > 0 specifies the actual data element in the data unit;
- FieldPosition = 0 implies out of the scope;
- OffsetPosition implies that the data unit contains octet string only, the key position is specified by the offset from the top of the data unit and the length.

Table 14 - Datatypes for keys

Key Type	Minimum Range	Order
ASN.1 INTEGER	(1-2) octets	increasing numeric value
ASN.1 IA5String	(1-16) characters	lexical order
ASN.1 GraphicString	(1-16) characters	lexical order
ASN.1 GeneralString	(1-16) characters	lexical order
ASN.1 OCTET STRING	(1-16) characters	increasing value
INTAP-AS2 DecimalType	(1-18) digits	increasing numeric value
INTAP-AS2 NumericType	(1-18) digits	increasing numeric value

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.11.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of INTAP-AS2 as defined by the parameters.

6.11.9 Definition of transfer

6.11.9.1 Datatype definition

The file consists of data values which are of

a) Datatype1 defined in table 13, where the PrimType2 in the datatype is given by the INTAP-AS2 definition;

b) Datatype2 defined in table 13, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3 specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where "user-coded" is defined as an EXTERNAL. That EXTERNAL is defined by the INTAP-AS3 definition.

6.11.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1";

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2"; or

c) a value of "Datatype3". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname3".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.11.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.11.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 13 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.11.11 ASE specific specifications for FTAM

6.11.11.1 Structural simplification

This simplification loses information.

A document type INTAP-4 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents-type parameter on the F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type INTAP-4 can be accessed as a document of type INTAP-2 (allowed only when reading the file) by specifying document type INTAP-2 with appropriate data type parameters in the contents-type parameter on the F-OPEN request. The traversal order of the FADUs must be maintained.

NOTE - The traversal order is as reading the file as INTAP-4 in key order.

6.11.11.2 Access context selection

A document of type INTAP-4 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.11.11.3 The READ operation

When the READ operation is applied at last of file with an FADU Identity next, the responder transfers the error identifier which contains "5005" by F-DATA-END.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.12 INTAP-5 Indexed file with unique keys document type

6.12.1 Entry number : INTAP-5

6.12.2 Information objects

Table 15 - Information objects in INTAP-5

document type name			{ iso member-body 392 ftam(10) document-type(2) indexed-file-with-unique-keys(5) } "INTAP-5 indexed file with unique keys"
abstract syntax names			
name for asname1			{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as2(2) } "INTAP abstract syntax AS2"
name for asname2			{ iso standard 8571 abstract-syntax(2) ftam-fadu(2) } "FTAM FADU"
name for asname3			{ iso member-body 392 ftam(10) abstract-syntax(3) intap-node-name(3) } "INTAP node name abstract syntax"
transfer syntax names			{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax			
PARAMETERS	::=	SEQUENCE { datatypes DataTypes, keytype KeyType, keyposition KeyPosition }	
DataTypes	::=	SEQUENCE { record-significance [0] IMPLICIT INTEGER { variable (0), fixed (1) } OPTIONAL, parameters SEQUENCE OF CHOICE { parameter1 Parameter1, parameter2 Parameter2 } }	
KeyType	::=	CHOICE { parameter1 Parameter1, parameter2 Parameter2 } -- Parameter1, Parameter2 as defined for the document type INTAP-2	
KeyPosition	::=	CHOICE { fieldposition FieldPosition, offsetposition OffsetPosition }	
FieldPosition	::=	INTEGER	
OffsetPosition	::=	SEQUENCE { offset INTEGER, length INTEGER }	
file model			{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set			{ iso standard 8571 constraint-set(4) ordered-flat-unique-names(4) } "FTAM ordered flat constraint set with unique names"
file contents	Datatype1	::= PrimType2	--as defined by INTAP-AS2 in 8.7
	Datatype2	::= CHOICE {	node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element }
	Datatype3	::= INTAP-Node-Name	--as defined by INTAP-AS3 in 8.8

6.12.3 Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM. This document type defines the COBOL-like access structure of an indexed file with unique keys.

NOTE - Storage refers to the apparent storage within the virtual filestore.

6.12.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3: File Service Definition.*

6.12.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6.12.6 Abbreviations

FTAM File Transfer, Access and Management

6.12.7 Document semantics

The document consists of zero, one or more file access data units and each FADU consists of zero or one data unit, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see table 15). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the FTAM ordered flat constraint set with unique names:

- o The FADU Identities end, previous and traversal are not required for conformant implementations;
- o An INTAP node name is used to get a record with an access condition. An INTAP node name which consists of an access key and an access condition is carried by an FADU Identity of a primitive parameter.

One record corresponds to one data unit. The exact significance of the record is determined by the record-significance parameter. If its value is variable, the length of the record is less than or equal to the sum of each maximum-string-length given by the parameter1 and parameter2. If the value is fixed, or parameter is not present, the length of each record is exactly equal to the sum of each maximum-string-length given by the parameter1 and parameter2.

Each data element is a data type from the set of primitive data types defined in 8.7. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the DataTypes parameter.

The maximum-string-length field of parameter1 specifies the length of the value in octets for the INTEGER and OCTET STRING types. For character-type data elements, the maximum-string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For decimal-type or numeric-type numbers, length-1 specifies the precision. The precision of an exact numeric literal is the number of digits that it contains.

The maximum-string-length field of parameter2 specifies the length of the value in octets for the DecimalType and NumericType.

Each data unit in the file has a key associated with it, which is the user-coded form of Node Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in 8.7.

The type and length of the key are defined by the KeyType parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in table 16.

The position of the key in the data unit is specified by the KeyPosition parameter:

- o FieldPosition > 0 specifies the actual data element in the data unit;
- o FieldPosition = 0 implies out of the scope;
- o OffsetPosition implies that the data unit contains octet string only, the key position is specified by the offset from the top of the data unit and the length.

Table 16 - Datatypes for keys

Key Type	Minimum Range	Order
ASN.1 INTEGER	(1-2) octets	increasing numeric value
ASN.1 IA5String	(1-16) characters	lexical order
ASN.1 GraphicString	(1-16) characters	lexical order
ASN.1 GeneralString	(1-16) characters	lexical order
ASN.1 OCTET STRING	(1-16) characters	increasing value
INTAP-AS2 DecimalType	(1-18) digits	increasing numeric value
INTAP-AS2 NumericType	(1-18) digits	increasing numeric value

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.12.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of INTAP-AS2 as defined by the parameters.

6.12.9 Definition of transfer

6.12.9.1 Datatype definition

The file consists of data values which are

a) Datatype1 defined in table 15, where the PrimType2 in the datatype is given by the INTAP-AS2 definition;

b) Datatype2 defined in table 15, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3 specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where "user-coded" is defined as an EXTERNAL. That EXTERNAL is defined by the INTAP-AS3 definition.

6.12.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1";

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2"; or

c) a value of "Datatype3". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname3".

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

6.12.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a data unit, and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.12.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 15 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.12.11 ASE specific specifications for FTAM

6.12.11.1 Structural simplification

This simplification loses information.

The document type INTAP-5 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the contents-type parameter in F-OPEN request, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local real filestore of the responder.

A document of type INTAP-5 can be accessed as a document of type INTAP-2 (allowed only when reading the file) by specifying document type INTAP-2 with appropriate data type parameters in the contents-type parameter on the F-OPEN request. The traversal order of the FADUs must be maintained.

NOTE - The traversal order is as reading the file as INTAP-5 in key order.

A document of type INTAP-5 may be accessed as a document of type INTAP-4 (allowed only when reading the file) by specifying document type INTAP-4 in the contents-type parameter in the F-OPEN request.

6.12.11.2 Access context selection

A document of type INTAP-5 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

6.12.11.3 The READ operation

When the READ operation is applied at last of file with an FADU Identity next, the responder transfers the error identifier which contains "5005" by F-DATA-END.

6.12.11.4 The INSERT operation

When the INSERT operation is applied the transferred material shall be the series of FADU which would be generated by reading any INTAP-5 document with the same parameter values in access context FA.

A transferred FADU whose name duplicates that of an already existing FADU will cause the INSERT operation to fail. The failure shall be signalled by issuing an F-TRANSFER-END response or an F-CANCEL request with a corresponding diagnostic.

6.12.11.5 The EXTEND operation

This operation is excluded for use with this document type.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

© ISO/IEC

6.12.11.6 The REPLACE operation

When the REPLACE operation is applied with FADU Identity begin, a transferred FADU whose name duplicates that of a previously transferred FADU will cause the REPLACE operation to fail. The failure shall be signalled by issuing an F-CANCEL request with a corresponding diagnostic.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.13 CGM-FTAM file document type

6.13.1 Entry number : CGM-FTAM

6.13.2 Information objects

Table 17 - Information objects in CGM-FTAM

document type name	{ iso(1) identified-organization(3) ewos(16) eg(2) ft(2) document-type(5) cgm(1) } "CGM-FTAM file"
abstract syntax names	
name for asname1	{ iso standard 8571 abstract-syntax(2) unstructured-binary(4) } "FTAM unstructured binary abstract syntax"
name for asname2	{ iso standard 8571 abstract-syntax(2) ftam-fadu (2) } "FTAM FADU"
transfer syntax names	{ joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"
parameter syntax PARAMETERS ::= SEQUENCE { maximum-string-length [0] IMPLICIT INTEGER OPTIONAL, string-significance [1] IMPLICIT INTEGER {variable(0), fixed(1), not-significant(2)} DEFAULT not-significant, cgm-encoding [2] IMPLICIT INTEGER {binary(0), character(1)} DEFAULT binary }	
file model	{ iso standard 8571 file-model(3) hierarchical(1) } "FTAM hierarchical file model"
constraint set	{ iso identified-organization ewos(16) eg(2) ft(2) constraint-set(4) two-levels-depth-hierarchical-unique-names(8) } "FTAM two levels depth hierarchical constraint set with unique names"
file contents	Datatype1 ::= OCTET STRING Datatype2 ::= CHOICE { node-descriptor-data-element Node-Descriptor-Data-Element, enter-subtree-data-element Enter-Subtree-Data-Element, exit-subtree-data-element Exit-Subtree-Data-Element }

6.13.3 Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

6.13.4 References

ISO 8571-1:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1 : General introduction.*

ISO 8571-2:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 2 : Virtual Filestore Definition.*

ISO/IEC 8632-1:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 1 : Functional specification.*

ISO/IEC 8632-2:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 2 : Character encoding.*

ISO/IEC 8632-3:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 3 : Binary encoding.*

6.13.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1. Moreover the terms metafile, BEGIN METAFILE, END METAFILE, metafile descriptor, picture, BEGIN PICTURE, END PICTURE, binary encoding and character encoding are defined in ISO 8632-1, ISO 8632-2, and ISO 8632-3.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.13.6 Abbreviations

FTAM File Transfer, Access and Management

CGM Computer Graphics Metafile

6.13.7 Document semantics

The document consists of zero, one or more file access data units, each of which consists of zero, one or more binary strings. The order of each of these elements is significant. The semantics of the binary strings is not specified by this document type.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the two levels depth hierarchical constraint set with unique names (see table 17).

An example of CGM mapping onto FTAM structure using CGM-FTAM file document type is shown in figure 2.

The contents of the document is a CGM file either with binary encoding or character encoding, depending on the value of the cgm-encoding parameter.

The node at level zero corresponds to the CGM file. It has no name and no data unit associated.

The nodes at the first level correspond to metafiles.

Each node at the first level shall have no data unit associated. The name of a node at the first level is the GraphicString corresponding to the identifier contained in the CGM BEGIN METAFILE delimiter element.

NOTE - The indentifier, in a binary encoded CGM, is a string encoded as a count (unsigned integer) followed by characters. Only characters following the count are assumed to form the corresponding node name. In a character encoded CGM, the identifier is a string encoded as sequences of bytes, starting with START OF STRING (SOS) and terminated by STRING TERMINATOR (ST). Only the characters between SOS and ST are assumed to form the corresponding node name.

The nodes at level 2 shall have a name and a data unit associated as follows:

meta1.bm (i.e. the name of the father node with ".bm" appended) shall be the name associated to the node corresponding to the BEGIN METAFILE element; the data unit associated shall be the OctetString corresponding to the BEGIN METAFILE element.

meta1.mfd (i.e. the name of the father node with ".mfd" appended) shall be the name associated to the node corresponding to the CGM metafile descriptor; the data unit associated shall be the OctetString corresponding to the metafile descriptor, i.e. all the CGM elements between the BEGIN METAFILE element and the first occurring delimiter element, being either BEGIN PICTURE or END METAFILE. These two elements are not part of the contents of the data unit.

pict1, pict2, ..., pictn (i.e. the GraphicStrings corresponding to the identifier (see note above) contained in the CGM BEGIN PICTURE element) shall be the names associated to the nodes corresponding to the CGM pictures; the data units associated shall be the OctetString corresponding to the pictures, i.e. all the CGM elements between the BEGIN PICTURE and END PICTURE elements. These two elements are part of the contents of the data units.

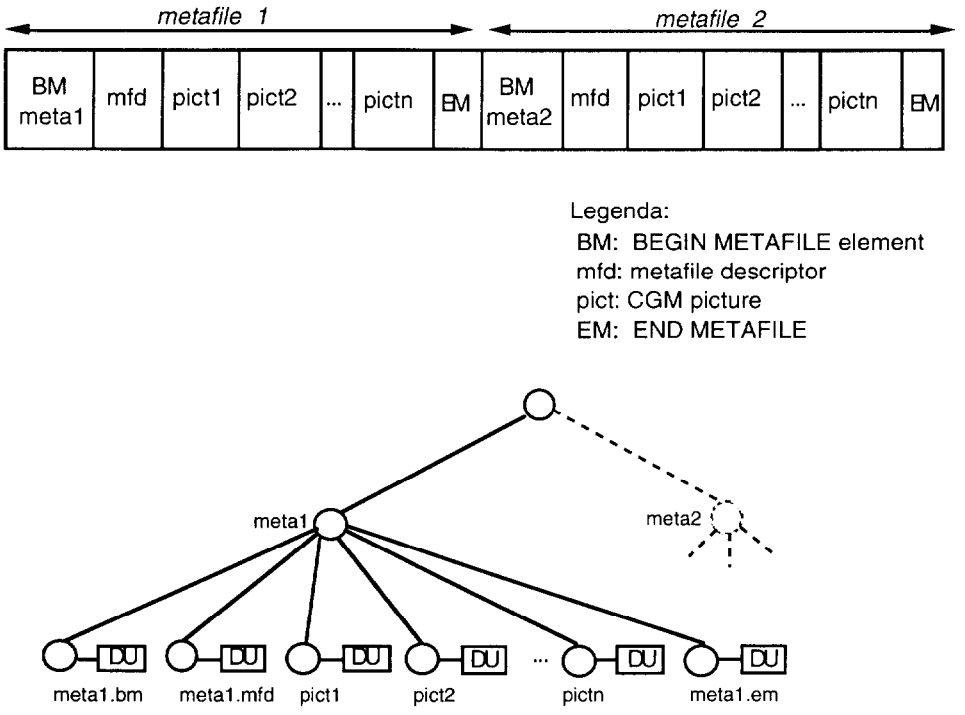


Figure 2 - Structure of an CGM-FTAM file (example)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

meta1.em (i.e. the name of the father name of the metafile with ".em" appended) shall be the name associated to the node corresponding to the CGM END METAFILE element; the data unit associated shall be the OctetString corresponding to the END METAFILE element.

There are no size or length limitations imposed by this definition, except those specified here. Each binary string is of a length determined by the number of octets given by the maximum-string-length parameter. If the maximum-string-length parameter is not present, the default is that the binary strings are unbounded.

The exact significance of the binary strings is determined by the string-significance parameter. If its value is variable, the length of the binary strings is less than or equal to the length given. If the value is fixed, the length of each binary string is exactly equal to the length given. If the value is not-significant, the boundaries of the binary strings are not necessarily preserved when the file is stored and do not contribute to the document semantics. The default value is not-significant.

The type of CGM encoding is determined by the cgm-encoding parameter. If its value is binary, the CGM binary encoding is applied. If the value is character, the CGM character encoding is applied. The default value is binary.

6.13.8 Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 modules ISO8571-FADU and ISO8571-CONTENTS in ISO 8571, in which each of the data elements has the abstract syntactic structure of an ISO FTAM unstructured binary document defined in document type registration entry FTAM-3 in ISO 8571.

6.13.9 Definition of transfer

6.13.9.1 Datatype definition

The presentation data values used for transfer are of either

a) Datatype1 defined in table 17;

b) Datatype2 defined in table 17, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

6.13.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" declared in table 17; or

b) one value of the ASN.1 datatype "Datatype2". All values are transmitted in the same (but any) presentation context extablished to support the abstract syntax name "asname2" declared in table 17.

NOTES

1 Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2 Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between P-DATA primitives are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

6.13.9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of type a) and b) is the same as the sequence of data elements within a data unit (one data element per data unit) and data units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

6.13.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 17 for all presentation data values transferred.

Implementations may optionally support other named transfer syntaxes.

6.13.11 ASE specific specifications for FTAM

6.13.11.1 Structural simplification

This simplification loses information.

The document type CGM-FTAM may be simplified to the document type FTAM-3 (allowed only when reading the file). The resultant document contains the same sequence of data values as would result from accessing the structured binary file in access context UA.

6.13.11.1.1. String length relaxation

This operation loses explicit information in the document type identification.

A document of type CGM-FTAM may be relaxed to another document of type CGM-FTAM with a larger maximum-string-length parameter, or no maximum-string-length parameter (unbounded strings).

6.13.11.2 Access context selection

A document type of type CGM-FTAM may be accessed in any one of the access contexts defined in the two levels depth hierarchical constraint set with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7 Constraint sets

7.1 NBS ordered flat constraint set

7.1.1 Field of application

The NBS ordered flat constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made on an FADU basis by position in the sequence.

7.1.2 Basic constraints

The basic constraints of the NBS ordered flat constraint set are given in table 18.

Table 18 - Basic constraints in the NBS ordered flat constraint set

Constraint set descriptor	"NBS ordered flat constraint set"
Constraint set identifier	{ iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-ordered-flat(1) }
Node names	None
File access actions	Locate, Read, Insert, Erase, Replace
Qualified actions	None
Available access contexts	HA, FA, UA
Creation state	Root node without an associated data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected; previous gives last node in traversal sequence, current and next give an error
Read whole file	Read in access context FA or UA with FADU Identity of begin
Write whole file (append)	Transfer the series of leaf FADUs which would be generated by reading the whole file in access context FA; perform the transfer with an FADU Identity of end and a file access action of insert.
Write whole file (replace)	Transfer the series of leaf FADUs which would be generated by reading the whole file in access context HA; perform the transfer with FADU Identity of begin and file action of replace.

7.1.3 Structural constraints

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and may have an associated data unit; all arcs from the root node shall be of length one.

7.1.4 Action constraints

Insert: The insert action is allowed only at the end of file. If the FADU Identity is end, the new node is inserted following all existing nodes in the file. If the FADU Identity is node number, the number must be at least one greater than the node number of the last existing node. Any nodes between the last existing node and the new node are empty, i.e., nodes without data. If the FADU identity is a node number not greater than that of the last existing node, an error will occur.

The location following insert is end.

Erase: The erase action is only allowed at the root node to empty the file, with FADU Identity of begin. The result is a solitary root node without an associated data unit.

NOTE - It is the intention when using this constraint set to allow for emptying an FADU, i.e., leaving an FADU with a DU of data length 0 (or without a DU); afterwards data may be reinserted into this hole. In order to empty an FADU, the replace operation may be used with new data of length zero (or with an FADU whose data-exists bit is set to FALSE and no DU). Refilling the hole is accomplished by a replace operation with the new DU (or with the new FADU, whose data-exists bit is set to TRUE and the new DU).

7.1.5 Identity constraints

The FADU Identity associated with the file action shall be one of the identities begin, end, first, last, current, next, previous or a node number greater than or equal to one. The actions with which these identities can be used are given in table 19.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 19 - Identity constraints in the NBS ordered flat constraint set

Action	Begin	End	First	Last	Current	Next	Previous	Node number
Locate	valid	valid	valid	valid	valid	valid	valid	valid
Read	whole		leaf	leaf	leaf	leaf	leaf	leaf
Insert		leaf						leaf
Erase	whole							
Replace	whole		leaf	leaf	leaf	leaf	leaf	leaf

7.2 NBS random access constraint set

7.2.1 Field of application

The NBS random access constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made randomly by NodeSeq. Structuring of the file into individual FADUs is determined by the Node-Name.

7.2.2 Basic constraints

The basic constraints of the NBS random access constraint set are given in table 20.

Table 20 - Basic constraints in the NBS random access constraint set

Constraint set descriptor	"NBS random access constraint set"
Constraint set identifier	{ iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-random-access(2) }
Node names	All names shall be of the same type; the type of the names and an ordering of the names shall be defined when reference is made to the constraint set.
File access actions	Locate, Read, Insert, Erase, Replace
Qualified actions	None
Available access contexts	UA
Creation state	Root node without an associated data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected
Read whole file	Read in access context UA with FADU Identity of begin
Write whole file	Transfer a series of leaf FADUs which would be generated by reading the whole file in access context UA; perform the transfer with an FADU Identity of end and a file access action of insert, or with an FADU Identity of begin and an action of replace.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.2.3 Structural constraints

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and shall have an associated data unit; all arcs from the root node shall be of length one.

7.2.4 Action constraints

Insert: The insert action is allowed only at the end of file, with FADU Identity of end; the new node is inserted following all existing nodes in the file. The location following the insert is end.

Erase: The erase action is allowed at the root node to empty the file, with FADU Identity of begin. The result is a solitary root node without an associated data unit. Erase with the FADU Identity of node number means truncation of the file.

Replace whole file: The FADU Identity is begin and the complete series of new FADU contents is sent.

Replace new leaves: The FADU Identity is node number and the number of FADUs being replaced is given by the number of FADUs sent.

7.2.5 Identity constraints

The FADU Identity associated with the file action shall be one of the identities begin, end, node number or NodeSeq. The actions with which these identities can be used are given in table 21.

Table 21 - Identity constraints in the NBS random access constraint set

Action	Begin	End	NodeSeq	Node number
Locate				leaf
Read	whole		leaf	
Insert		leaf		
Erase	whole			leaf
Replace	whole			leaf

NOTE - NodeSeq = A sequence of node names with a single member

7.3 INTAP sequential flat constraint set

7.3.1 Field of application

The INTAP sequential flat constraint set applies to files which are structured into a sequence of individual FADUs and to which COBOL-like access may be made on an FADU basis by position in the sequence.

7.3.2 Basic constraints

The basic constraints of the INTAP sequential flat constraint set are given in table 22.

7.3.3 Structural constraints

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and shall have an associated data unit; all arcs from the root node shall be of length one.

7.3.4 Action constraints

Insert: The insert action is allowed only at the end of file, with an FADU Identity of end; the new node is inserted following all existing nodes in the file.

The location following insert is end.

Replace leaf node: The replace leaf node action is only allowed with an FADU Identity of current, and the length of a transferred data unit must be equal to the length of a data unit which already exists.

Erase: The erase action is only allowed at the root node to empty the file, with FADU Identity of begin. The result is a solitary root node without an associated data unit.

7.3.5 Identity constraints

The FADU Identity associated with the file action shall be one of the identities begin, end, first, last, current, next, previous or a traversal number greater than or equal to one. The actions with which these identities can be used are given in table 23.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 22 - Basic constraints in the INTAP sequential flat constraint set

Constraint set descriptor	"INTAP sequential flat constraint set"
Constraint set identifier	{ iso member-body 392 ftam(10) constraint-set(5) sequential-flat(1) }
Node names	None
File access actions	Locate, Read, Insert, Replace, Erase
Qualified actions	None
Available access contexts	HA, FA, UA
Creation state	Root node without an associated data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected; previous gives last node in traversal sequence, current and next give an error
Read whole file	Read in access context FA or UA with FADU Identity of begin
Write whole file (append)	Transfer the series of leaf FADUs which would be generated by reading the whole file in access context FA; perform the transfer with an FADU Identity of end and a file access action of insert.
Write whole file (replace)	Transfer the FADU represented by the series of data elements which would be generated by reading the file in access context HA; perform the transfer with an FADU Identity of begin and file action of replace.

Table 23 - Identity constraints in the INTAP sequential flat constraint set

Action	Begin	End	First	Last	Current	Next	Previous	Traversal
Locate	valid	valid	valid	valid	valid	valid	valid	valid
Read	whole		leaf	leaf	leaf	leaf	leaf	leaf
Insert		leaf						
Erase	whole							
Replace	whole				leaf			

7.4 Two levels depth hierarchical constraint set with unique names

7.4.1 Field of application

The Two levels depth hierarchical constraint set with unique names applies to files which are hierarchically structured. The file structure consists of a hierarchical file with at least one level and at most two levels in depth. The node names of the children of any given parent are unique.

7.4.2 Basic constraints

The basic constraints in the Two levels depth hierarchical constraint set with unique names are given in table 24.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 24 - Basic constraints in the two levels depth hierarchical constraint set with unique names

Constraint set descriptor	"FTAM two levels depth hierarchical constraint set with unique names"
Constraint set identifier	{ iso(1) identified-organization(3) ewos(16) eg(2) ft(2) constraint-set (4) two-levels-depth-hierarchical-unique-names (8) }
Node names	All names shall be of the same type; the type of the names shall be defined when reference is made to the constraint set; the names of the children of any particular parent shall be unique
File access actions	Locate, Read, Insert, Replace, Extend, Erase
Qualified actions	Insert: normal action "insert as a sister"; qualified action "insert as a child"
Available access contexts	HA, HN, FA, FL, FS, UA, US
Creation state	Root node without an associated data unit
Location after open	Beginning of file
Beginning of file	No node selected; next gives the first node in the traversal sequence (the root node); current and next give an error
End of file	No node selected; previous gives the last node in the traversal sequence; current and next give an error
Read whole file	Read in access context HA with FADU identity consisting of an empty sequence of node names
Write whole file	Transfer the FADU as the sequence of data-elements which would be generated by reading the file in access context HA: perform the transfer with a FADU identity consisting of node names and a file access action of replace

7.4.3 Structural constraints

The root node shall not have an associated data unit and shall not have a node name. All nodes at level one as well as all nodes at level two shall have an associated data unit and shall have a node name. All children of the nodes at level one, if any, shall be leaf nodes. All arcs shall be of length one.

7.4.4 Action constraints

Insert: there are two forms of the insert action. The name of the inserted node must be distinct from that of any node already present as a sister of the inserted node.

Normal action - Insert as a sister: the created FADU is placed in the preorder traversal sequence after the current location so that the arc linking it to its parent is of length one. The inserted FADU and the current location share the same parent node.

Insert as a sister cannot be performed if the current location is the root of the file.

The current location after the action is the root node of the newly inserted FADU.

Qualified action - Insert as a child: on beginning a series of qualified actions, a copy of the current location is recorded and a new location established which is in the traversal sequence between the previous location and the next node in the traversal sequence (that is, before the first child of the previously located node, if any).

Qualified actions are then performed with the same semantics as the normal insert action insert as sister, but using this temporary location.

Insert as a child cannot be performed if the current location is a node at level two of the hierarchy.

At the end of a series of qualified actions, the current location is restored from the copy of the location taken at the beginning of the series of qualified actions.

Other actions: other actions are exactly as specified in clause 11 of ISO 8571-2.

7.4.5 Identity constraints.

The FADU identity associated with the file action shall be one of the identifiers begin, end, current, next, previous, a node name, or a sequence of node names. The actions with which these identifiers can be used are given in table 25.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 25 - Identity constraints in the NBS random access constraint set

Action	Begin	End	Current	Next	Previous	NodeName	NodeSeq
Locate	valid	valid	valid	valid	valid	valid	valid
Read			valid	valid	valid	valid	valid (note)
Insert			valid	valid	valid	valid	valid
Replace			valid	valid	valid	valid	valid (note)
Extend			valid	valid	valid	valid	valid
Erase			valid	valid	valid	valid	valid (note)

NOTE - An empty sequence of node names is used to address the root node when performing actions on the whole file.

8 Abstract syntaxes

8.1 Abstract syntax NBS-AS2

Abstract syntax name:

{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2) }

"NBS file directory entry abstract syntax"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 type NBS-AS2.FileDirectoryEntry (see figure 3).

For this abstract syntax the following transfer syntax will be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

```
NBS-AS2 DEFINITIONS ::=
BEGIN
FileDirectoryEntry  ::= [PRIVATE 2] Read-Attributes
Read-Attributes     ::= ISO8571-FTAM.Read-Attributes
END
```

Figure 3 - NBS-AS2 definition

8.2 Abstract syntax INTAP-AS1

Abstract syntax name:

{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as1(1) }

"INTAP abstract syntax AS1"

This is an abstract syntax which defines the file contents data elements as an ASN.1 type INTAP-1-DOCUMENT-TYPE. INTAP-1-Record-Element (see figure 4).

Figure 5 gives an example of INTAP-1 records.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
ISO8571-CONTENTS.File-Contents-Data-Element ::= INTAP-1-DOCUMENT-TYPE.INTAP-1-Record-Element

INTAP-1-DOCUMENT-TYPE DEFINITIONS ::=

                    BEGIN

                        INTAP-1-Record-Element  ::= CHOICE {

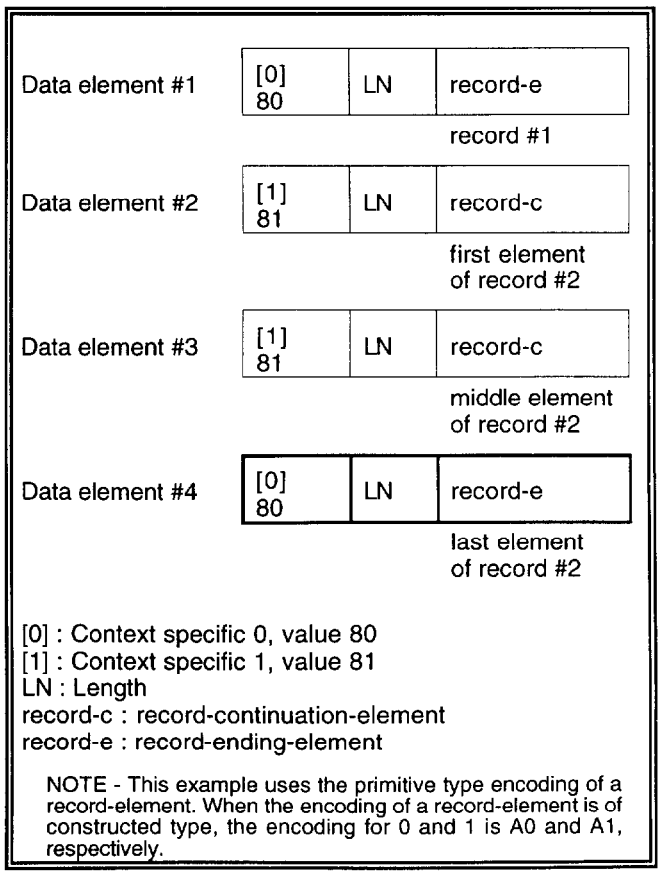
                        record-ending-element           [0] IMPLICIT OCTET STRING,

                        record-continuation-element     [1] IMPLICIT OCTET STRING }

                    -- The last INTAP-1-Record-Element in the bulk data
                    -- transfer must be of format  record-ending-element.
                    --
                    -- <Record of INTAP-1> ::=
                    --      <record-ending-element> |
                    --      <record-continuation-elements><record-ending-element>
                    -- <record-continuation-elements> ::=
                    --      <record-continuation-element> |
                    --      <record-continuation-element><record-continuation-elements>

                    END
```

Figure 4 - INTAP-AS1 definition



NOTE - This example uses the primitive type encoding of a record-element. When the encoding of a record-element is of constructed type, the encoding for 0 and 1 is A0 and A1, respectively.

Figure 5 - Example of INTAP-1 records

8.3 Abstract syntax NBS-AS1

Abstract syntax name:

{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) }

"NBS abstract syntax AS1"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 type NBS-AS1.PrimType (see figure 6).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

NOTES

1 The mantissa is a number in the range (1/2 < mantissa < 1).

2 The value is equal to mantissa * 2 exponent

3 The first bit in the mantissa is most significant.

4 See IEEE 754 for definitions of terminology, such as NaN.

ANSI/IEEE 754:1985, Standard for Binary Floating-Point Arithmetic.

5 A minimum length range (in bits) is required for the components of FloatingPointNumber, as follows : mantissa 1-23 bits, and exponent 0-8 bits.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
NBS-AS1 DEFINITIONS ::=
BEGIN
   PrimType ::= CHOICE {   integer              INTEGER,
                           bitstring            BIT STRING,
                           boolean              BOOLEAN,
                           ia5string            IA5String,
                           graphicstring        GraphicString,
                           generalstring        GeneralString,
                           octetstring          OCTET STRING,
                           utctime              UTCTime,
                           generalizedtime      GeneralizedTime,
                           null                 NULL,
                           floatingpointnumber  FloatingPointNumber    }
   --      The support for IA5String is the ISO/IEC 646 IRV G0 character set and the ISO/IEC 646 IRV C0 set.
   --      The minimum level of support for GraphicString is the ISO/IEC 646 IRV G0 character set and the
   --      ISO 8859-1 G0 and G1 sets.
   --      The minimum level of support for GeneralString is the ISO/IEC 646 IRV G0 character set and the
   --      ISO 8859-1 G0 and G1 character sets, and the ISO/IEC 646 IRV C0 set.
   FloatingPointNumber ::=
         [PRIVATE 0] CHOICE  {   finite          [0] IMPLICIT SEQUENCE
                                                 {     sign  Sign,
                                                       mantissa  BIT STRING,
                                                       -- first bit must be 1
                                                       exponent INTEGER        },
                                 infinity        [1] IMPLICIT Sign,
                                 signalling-nan  [2] IMPLICIT NaN,
                                 quiet-nan       [3] IMPLICIT NaN,
                                 zero            [4] IMPLICIT NULL   }
   Sign  ::= INTEGER { positive (0), negative (1) }
   NaN  ::= INTEGER
END
```

Figure 6 - NBS-AS1 definition

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.4 NBS node name abstract syntax

Abstract syntax name:

{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-node-name(3) }

"NBS random access node name abstract syntax"

This is an abstract syntax for the user coded Node-Name in the FTAM FADU abstract syntax (see figure 7).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

```
NBS-AS3 DEFINITIONS ::=
BEGIN
    NBS-Node-Name ::= SEQUENCE      {       starting-fadu       [0] IMPLICIT INTEGER,
                                            fadu-count          [1] IMPLICIT INTEGER
                                    -- A fadu-count of 0 specifies the range of FADUs beginning
                                    -- at starting-fadu and ending at end of file
                                    }
END
```

Figure 7 - NBS-AS3 definition

8.5 NBS random binary access file abstract syntax

Abstract syntax name:

{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-random-binary(4) }

"NBS random binary access file abstract syntax"

This is an abstract syntax for the transfer of file contents for NBS random binary files (see figure 8).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

```
NBS-AS4 DEFINITIONS ::=
BEGIN
    NBS-Random-Binary ::= OCTET STRING
                    -- Contains one or more presentation data values
                    -- concatenated together. Each presentation data
                    -- value is defined as Datatype1 in table 7.
END
```

Figure 8 - NBS-AS4 definition

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.6 NBS simple text abstract syntax

Abstract syntax name:

{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-simple-text(5) }

"NBS simple text abstract syntax"

This is an abstract syntax for the transfer of file contents for NBS simple text files (see figure 9).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

```
NBS-AS5 DEFINITIONS ::=
BEGIN
    NBS-Text ::= CHOICE { ia5string      IA5String,      -- Universal Class 22
                          graphicstring  GraphicString,  -- Universal Class 25
                          visiblestring  VisibleString,  -- Universal Class 26
                          generalstring  GeneralString   -- Universal Class 27
                          }
END
```

Figure 9 - NBS-AS5 definition

8.7 Abstract syntax INTAP-AS2

Abstract syntax name:

{ iso member-body 392 ftam(10) abstract-syntax(3) intap-as2(2) }

"INTAP abstract syntax AS2"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 type INTAP-AS2.PrimType2 (see figure 10).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

NOTES

1 GraphicString, GeneralString and OCTET STRING are optionally supported.

2 By supporting GraphicString and GeneralString, the code extension is available, and the character set can be used beyond the limits defined in IA5String; ISO/IEC 646 IRV G0 and C0 sets.

As for OCTET STRING, it can be used when data are transparently exchanged in FTAM on the assumption that detailed contents of data are mutually recognized between file service users.

3 See ISO 1989 for definitions of COBOL data types.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
INTAP-AS2 DEFINITIONS ::=
BEGIN
    PrimType2   ::=   CHOICE {   integer          INTEGER,
                                 ia5string        IA5String,
                                 graphicstring    GraphicString,
                                 generalstring    GeneralString,
                                 octetstring      OCTET STRING,
                                 decimaltype      DecimalType,
                                 numerictype      NumericType      }

    -- The support for IA5String is the ISO/IEC 646 IRV G0 character set and the ISO/IEC 646 IRV C0 set.
    -- The minimum level of support for GraphicString is the ISO/IEC 646 IRV G0 character set.
    -- The minimum level of support for GeneralString is the ISO/IEC 646 IRV G0 and C0 character set.

    DecimalType ::=   CHOICE      {
                      unsign      [PRIVATE 0] IMPLICIT OCTET STRING,
                      sign        [PRIVATE 1] IMPLICIT OCTET STRING  }

    -- This defines a parameter of type decimal. A decimal field represents a number according to PL/1 notation :
    -- packed and the least significant nibble containing the sign using 1010(2), 1100(2), 1110(2) and 1111(2) for plus,
    -- and 1011(2) and 1101(2) for minus. The length of the field equals the smallest integer greater than (precision/2).

    NumericType ::=   CHOICE      {
                      unsign      [PRIVATE 2] IMPLICIT PrintableString,
                      trailing    [PRIVATE 3] IMPLICIT PrintableString,
                      trailing-with-separate-character  [PRIVATE 4] IMPLICIT PrintableString,
                      leading     [PRIVATE 5] IMPLICIT PrintableString,
                      leading-with-separate-character [PRIVATE 6] IMPLICIT PrintableString     }

    -- This defines a parameter of numeric type. A numeric field represents a number according to COBOL notation :
    -- printable digits with a leading sign + or -. The length of the string therefore equals (precision+1).

END
```

Figure 10 - INTAP-AS2 definition

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.8 Abstract syntax INTAP-AS3

Abstract syntax name:

{ iso member-body 392 ftam(10) abstract-syntax(3) intap-node-name(3) }

"INTAP node name abstract syntax"

This is an abstract syntax for the user coded Node-Name in the FTAM FADU abstract syntax (see figure 11).

For this abstract syntax, the following transfer syntax can be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

```
INTAP-AS3 DEFINITIONS ::=

BEGIN

    INTAP-Node-Name ::= CHOICE {
            node-value    PrimType2,

                -- This value is used to specify the Node-Name in the
                -- Node-Descriptor-Data-Element.

            fadu-identity  SEQUENCE {
                    node-value  [0] PrimType2,
                    access-condition  [1] IMPLICIT Access-condition  DEFAULT equal-to  }  }

                -- This value is used to specify the FADU Identity of name-list in the
                -- FTAM PDUs specifying FADU identity.

    Access-condition    ::=    INTEGER { equal-to                (1),
                                         greater-than             (2),
                                         greater-than-or-equal-to (3)
                                         }

END
```

Figure 11 - INTAP-AS3 definition

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9 Transfer syntaxes

9.1 Transfer syntax INTAP-TS1

Transfer syntax name:

{ iso member-body 392 ftam(10) transfer-syntax(4) intap-ts1(1) }

"INTAP transfer syntax TS1"

9.1.1 Basic encoding

A File-Contents-Data-Element is encoded using the ASN.1 Basic Encoding Rules, and then the following compression method is applied.

9.1.2 Compression method

The objects for compression are individual File-Contents-Data-Elements (see figure 12 for an example of INTAP-1).

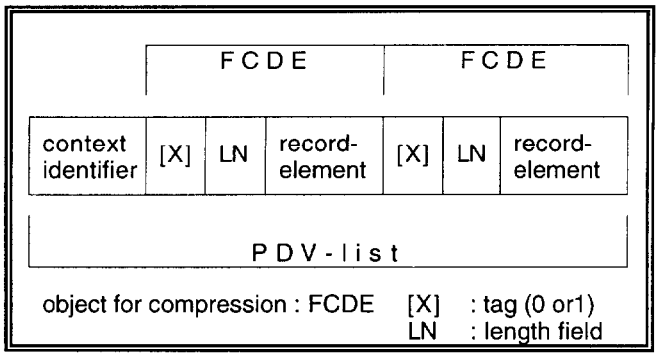
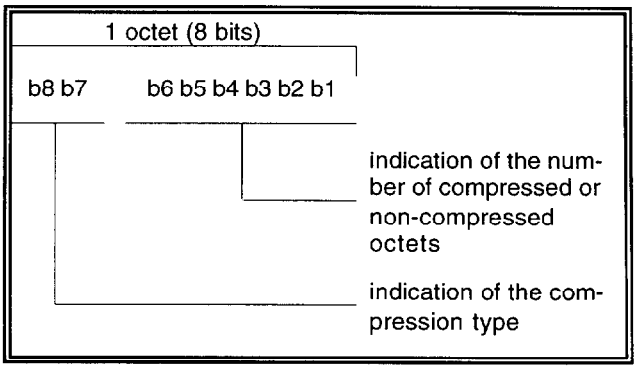


Figure 12 - INTAP-TS1 compression method

When the record-element in FCDE is encoded as a constructed type, the object of compression includes id, LN of the constructed type.

With this method the repetition of octets is compressed, using 1-octet compression control characters to indicate type and length of the compression (figure 13).



bit assignment		description
b8 b7	b6 b5 b4 b3 b2 b1	
00	number of octets not compressed (1-63)	No compression of the following octets.
11	number of compressed octets (1-63)	Compression of octets. In this case, one of the compressed octets follows the compression control character.

Figure 13 - Compression control character

The first octet of a File-Contents-Data-Element is always a compression control character (see also figure 14).

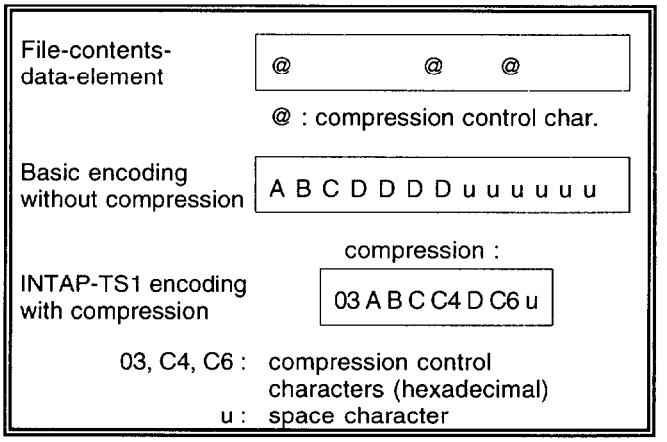


Figure 14 - Example for encoding using INTAP-TS1

NOTE - The compressed data shall be transferred by presentation data values of type octet-aligned.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex A

(informative)

Corrigenda

International Standards are subject to constant review and revision by the ISO/IEC Technical Committee concerned. The following corrigenda are approved by ISO/IEC JTC1, but at the date of publication of this part of ISO/IEC ISP 10607 they were not yet incorporated in the text of the corresponding base standards as referenced in clause 2. The corrigenda as listed below are identified at the time of publication of this part of ISO/IEC ISP 10607.

FTAM

ISO 8571-1:1988/Cor.1:1991

ISO 8571-2:1988/Cor.1:1991

ISO 8571-3:1988/Cor.1:1991

ISO 8571-3:1988/Cor.2:1992

ISO 8571-4:1988/Cor.1:1992

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex B

(informative)

Main differences between the 1st edition (1990/91/94) and the 2nd edition (1995) of this part of ISO/IEC ISP 10607

B.1 The normative references were updated for ISO/IEC 646, ISO/IEC 6429, ISO 8571, ISO/IEC 8824, ISO/IEC 8825, ISO/IEC TR 10000, ISO/IEC ISP 10607.

B.2 ISO/IEC ISP 10607-2:1990/Amd.1:1991, ISO/IEC ISP 10607-2:1990/Amd.2:1994 (COBOL document types) and ISO/IEC ISP 10607-2:1990/DAM.3 (CGM document type, not published) were integrated in the 2nd edition of this part of ISO/IEC ISP 10607.

B.3 The syntax of the ASN.1 definitions in this part of ISO/IEC ISP 10607 was aligned with the 95-edition of ISO/IEC 8824-1, i.e., for all components of ASN.1 types SEQUENCE, CHOICE and SET, identifiers preceeding the component types were added.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex C

(informative)

FTAM-COBOL Language Binding for the INTAP-2, -3, -4, -5 Document Types (Tutorial)

C.1 Aim of COBOL document types

Each enterprise has many resources which are shown as follows:

- An application program which accesses each record of a file in a local system
- The data in a file accessed by an application program

Many of these application programs are made in COBOL language and are used continuously between the past and present. The data used in the enterprise have a long life and new data addition and current data update have been carried out on the old data. These application programs and the data are indispensable resources for the enterprise activities and they are the resources which will be used in the future. Recently, the construction of the system environment often combines products of various vendors in each enterprise. In addition, the operation mode of the process is not only processed in the local system but is often processed in a remote system via networks. And, because implementing the file access in which FTAM is used became possible, access to a record of a file in a remote system is possible. In such situations, each enterprise will change the resources as resources which are not only used in the local system but are shared in each section of each enterprise.

If COBOL document types are used, the user can obtain the following advantages:

- The user is able to access a record of the file in the remote system without the change of a current application program.
- The user need not have knowledge of FTAM even when the user makes the application program which uses the file access function of FTAM for new jobs.

C.2 Support range of COBOL document types

COBOL document types offer the functions of the sequential file, the relative file and the indexed file within the range that level 1 and level 2 of COBOL language specification can be achieved. As for parts other than level 1 and level 2, because the level of support is different for each vendor, it is outside ranges achieved by COBOL document types.

Level 1 : Level 1 element of the functional unit of ISO 1989:1985 is shown.

Level 2 : Level 2 element of the functional unit of ISO 1989:1985 is shown.

[Notation]

- Symbols for Level 1 and Level 2

  X : Indicates that this element is present in the level.

  Z : Indicates that this element is present in the level, but is an obsolete element (an element to be deleted from the next revision of the standard).

  N : Indicates that this element is not present in the level.

- [ ] : This element is related to file access process.

- Symbols for support level of COBOL document types

  √ : support (standpoint is only for the actions taken)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

C.2.1 Sequential file module element

ELEMENT	Level 1	Level 2	COBOL document type
Procedure Division			
Declarative procedures	X	X	
DECLARATIVES option	X	X	
END DECLARATIVES option	X	X	
CLOSE statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
REEL/UNIT phrase	X	X	
FOR REMOVAL phrase	N	X	
WITH NO REWIND phrase	N	X	
WITH LOCK phrase	N	X	
OPEN statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
INPUT phrase	X	X	✓
WITH NO REWIND phrase	N	X	
REVERSED phrase	N	Z	
OUTPUT phrase	X	X	✓
WITH NO REWIND phrase	N	X	
I-O phrase	X	X	✓
EXTEND phrase	N	X	✓
INPUT,OUTPUT and I-O series	X	X	
EXTEND series	N	X	
READ statement	X	X	✓
NEXT phrase	N	X	✓
INTO phrase	X	X	
AT END phrase	X	X	✓
NOT AT END phrase	X	X	
END-READ phrase	X	X	
REWRITE statement	X	X	✓
FROM phrase	X	X	
END-REWRITE phrase	X	X	
USE statement	X	X	
EXCEPTION/ERROR PROCEDURE phrase	X	X	
ON file-name	X	X	
ON file-name series	N	X	
ON INPUT	X	X	
ON OUTPUT	X	X	
ON I-O	X	X	
ON EXTEND	N	X	
WRITE statement	X	X	✓
FROM phrase	X	X	
BEFORE/AFTER ADVANCING phrase	X	X	
Integer LINE/LINES	X	X	
Identifier LINE/LINES	X	X	
mnemonic-name	N	X	
PAGE	X	X	
AT END-OF-PAGE/EOP phrase	N	X	
NOT AT END-OF-PAGE/EOP phrase	N	X	
END-WRITE clause	X	X	

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.2.2 Relative file module element

ELEMENT	Level 1	Level 2	COBOL document type
Procedure Division			
Declarative procedures	X	X	
DECLARATIVES option	X	X	
END DECLARATIVES option	X	X	
CLOSE statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
WITH LOCK phrase	N	X	
DELETE statement	X	X	✓
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-DELETE phrase	X	X	
OPEN statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
INPUT phrase	X	X	✓
OUTPUT phrase	X	X	✓
I-O phrase	X	X	✓
EXTEND phrase	N	X	✓
INPUT,OUTPUT and I-O series	X	X	
EXTEND series	N	X	
READ statement	X	X	✓
NEXT phrase	N	X	✓
INTO phrase	X	X	
AT END phrase	X	X	✓
NOT AT END phrase	X	X	
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-READ phrase	X	X	
REWRITE statement	X	X	✓
FROM phrase	X	X	
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-REWRITE phrase	X	X	
START statement	N	X	✓
KEY phrase	N	X	✓
EQUAL TO, =	N	X	✓
GREATER THAN, >	N	X	✓
NOT LESS THAN , NOT <	N	X	✓
GREATER THAN OR EQUAL TO, >=	N	X	✓
INVALID KEY phrase	N	X	
NOT INVALID KEY phrase	N	X	
END-START phrase	N	X	
USE statement	X	X	
EXCEPTION/ERROR PROCEDURE phrase	X	X	
ON file-mane	X	X	
ON file-mane series	N	X	
ON INPUT	X	X	
ON OUTPUT	X	X	
ON I-O	X	X	
ON EXTEND	N	X	
WRITE statement	X	X	✓
FROM phrase	X	X	
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-WRITE phrase	X	X	

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.2.3 Indexed file module element

ELEMENT	Level 1	Level 2	COBOL document type
Procedure Division			
Declarative procedures	X	X	
DECLARATIVES option	X	X	
END DECLARATIVES option	X	X	
CLOSE statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
WITH LOCK phrase	N	X	
DELETE statement	X	X	✓
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-DELETE phrase	X	X	
OPEN statement	X	X	✓
file-name	X	X	✓
file-name series	X	X	
INPUT phrase	X	X	✓
OUTPUT phrase	X	X	✓
I-O phrase	X	X	✓
EXTEND phrase	N	X	✓
INPUT,OUTPUT and I-O series	X	X	
EXTEND series	N	X	
READ statement	X	X	✓
NEXT phrase	N	X	✓
INTO phrase	X	X	
AT END phrase	X	X	✓
NOT AT END phrase	X	X	
KEY phrase	N	X	✓
EQUAL TO, =	N	X	✓
GREATER THAN, >	N	X	✓
NOT LESS THAN , NOT <	N	X	✓
GREATER THAN OR EQUAL TO, >=	N	X	✓
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-READ phrase	X	X	
REWRITE statement	X	X	✓
FROM phrase	X	X	
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-REWRITE phrase	X	X	
START statement	N	X	✓
KEY phrase	N	X	✓
EQUAL TO, =	N	X	✓
GREATER THAN, >	N	X	✓
NOT LESS THAN , NOT <	N	X	✓
GREATER THAN OR EQUAL TO, >=	N	X	✓
INVALID KEY phrase	N	X	
NOT INVALID KEY phrase	N	X	
END-START phrase	N	X	
USE statement	X	X	
EXCEPTION/ERROR PROCEDURE Phrase	X	X	
ON file-name	X	X	
ON file-name series	N	X	
ON INPUT	X	X	
ON OUTPUT	X	X	
ON I-O	X	X	
ON EXTEND	N	X	
WRITE statement	X	X	✓
FROM phrase	X	X	
INVALID KEY phrase	X	X	
NOT INVALID KEY phrase	X	X	
END-WRITE phrase	X	X	

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.3 Achievement method of COBOL document types

COBOL document types are designed according to the policies and the basic methods shown as follows:

C.3.1 Policies

1) Minimization of the influence on the current FTAM products

Current FTAM products with the function of the file access must end only in the support of COBOL document types.

2) No influence on the current user resources

Current application programs made in COBOL language and current data in the file should be able to be used in the FTAM access without change of resources.

3) Adjustment with COBOL language specification

The file access of FTAM should be usable from the statement described by the procedure division of the application program directly.

4) Specifications according to the precedent

If the purpose of the specifications is the same, regulations of the basic standard (ISO 8571) and methods of other document types are adopted when the specifications of COBOL document types are decided.

C.3.2 Basic methods

1) Correspondence of an application program and a remote file

The initiator assigns an application program made in COBOL language and the virtual file of a COBOL document type. The responder assigns the virtual file of a COBOL document type and the real file in the remote system. The application program made in COBOL language can access the real file in the remote system by this assignment of two phases.

2) Composition of COBOL document types

COBOL document types are composed of the following four types.

- INTAP-2 : document type for a sequential file
- INTAP-3 : document type for a relative file
- INTAP-4 : document type for an indexed file (alternate record key)
- INTAP-5 : document type for an indexed file (prime record key)

3) Prime record key and alternate record keys of the indexed file

An indexed file has one prime record key and zero, one or more alternate record keys in one indexed file. Because two or more record keys are occasionally handled in an indexed file, the file model of FTAM referred to by COBOL document types must be a file model with two or more keys. As for the action method, because the basic standard (ISO 8571) does not provide for the file model with two or more keys, the method which is to add a new file model to the basic standard (ISO 8571) is considered. However, it is difficult for the FPM part of the current FTAM product to accept this method. In addition, because a new file model is complex structural, each vendor cannot positively promise the support of this file model. Moreover, because the change of the basic standard (ISO 8571) requires a long discussion period, the user cannot use COBOL document types in the early stage. Consequently, to reduce the influence on the FTAM product, the indexed file of COBOL document types is based on the current file model. The indexed file is a composition of the virtual files with one record key. Because each record key of the indexed file is achieved in the

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

virtual files separately, the FPM part of each product can handle the virtual files of the indexed file as well as other virtual files. Even if COBOL document types are supported, the FPM part need not be changed because each virtual file is a current file model.

4) Restriction in operation of INTAP-4 and INTAP-5

The restriction in the prime record key and the alternate record key operation is described in COBOL language specification. When the indexed file is accessed with the prime record key, the operation of insertion, change or deletion can be executed. When the indexed file is accessed with the alternate record key, only the operation of retrieval can be executed. The prime record key is offered with INTAP-5 and the alternate record key is offered with INTAP-4. The WRITE operation is permitted in INTAP-5 and only the READ operation is permitted in INTAP-4.

5) Relation between INTAP-4 and INTAP-5

The change might occur in the virtual file of INTAP-4 if the WRITE operation is executed by using the prime record key for the virtual file of INTAP-5 by the application program. In that case, the updating of the index of the indexed file is executed by an actual data management system. COBOL document types do not provide for the cooperation method between INTAP-5 and INTAP-4.

6) Alternate record key

There are two types of alternate record keys of the indexed file, as follows:

a) Alternate record key with unique value

b) Alternate record key with non unique value

The indexed file with an alternate record key is achieved with INTAP-4. In case of a), the examination whether all alternate record keys have different values is achieved by the local data management system.

7) REWRITE of a record in the sequential file

COBOL language specification offers the REWRITE operation in the sequential file. Because the sequential flat constraint set included in the basic standard (ISO 8571) does not define the replace action, the sequential file of INTAP-2 cannot use this constraint set. The INTAP-2 uses the INTAP sequential flat constraint set newly defined. The INTAP sequential flat constraint set is an addition of the replace action to the sequential flat constraint set.

8) Conditional pointing by a relative record number in the relative file

In the relative file, the application program is able to point to the record with a specified relative record number. The INTAP-Node-Name is newly defined as the application program is able to specify the relative record number using INTAP-3. The INTAP-Node-Name is an abstract syntax to enable the specification of the relative record number and the condition. The INTAP-Node-Name is carried to the remote system by the FADU-ID parameter of F-LOCATE.

9) Conditional pointing and reading by a record key in the indexed file

In the indexed file, the application program is able to point and to read to the record with a specified record key. The INTAP-4 and INTAP-5 enable the specification of the record key and the condition by using the INTAP-Node-Name as well as INTAP-3. The INTAP-Node-Name is carried to the remote system by the FADU-ID parameter of F-LOCATE and F-READ.

10) AT END phrase in sequential access mode

In case of file record access using sequential access mode, an application program made in COBOL language has an end of data procedure (AT END phrase) in the part of read access procedure.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The file access using COBOL document types offers the mapping manner of an error identifier of diagnostic parameter values of FTAM and a COBOL language element. When the responder detects the condition of an end of data in the read access, the responder transfers a "5005" as the error identifier to the initiator.

If the initiator receives a "5005" from the responder, the initiator needs to understand the condition of the end of data in the read access and to perform the end of data procedure (see figure C.1).

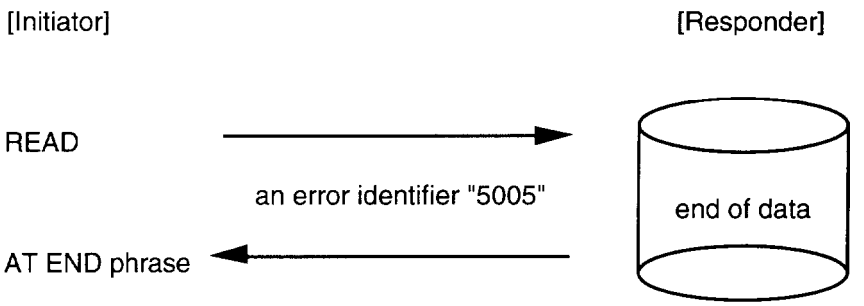


Figure C.1 - AT END procedure

11) Data type of COBOL

As for the character and the numeric of the data type of COBOL language, the representation in the local system might be different with each vendor. Therefore, a standard representation is defined as PrimType2. Each vendor's FTAM product can communicate with other systems by converting the local representation to the data type PrimType2.

12) Creation and deletion of the indexed file

The creation and the deletion of the indexed file are made beyond the limits of the rule provided by COBOL document types. The indexed file is achieved in COBOL document types by two or more virtual files. However, because the relationships between two or more virtual files is not defined in COBOL document types, it is necessary to offer a local procedure by each system as for the creation and the deletion of the indexed file. Generally, the completion of a making application program and the completion of a making file are the same points in the job constructed with COBOL language. Therefore, the operation of the creation and the deletion is limited while executing the job.

C.4 Mapping of COBOL data-type

C.4.1 Purpose of Mapping

In COBOL programs data are precedingly defined by data description entry in the process of programming. Data description entry defines whether the data item is nonnumeric or numeric, and when it is numeric it provides the following information: the number of digits, the position of the decimal point, whether signed and the position of the sign, whether the numeric is binary, packed decimal or display type numeric strings and others.

As for the character code used for nonnumeric item/display type numeric strings or sign code of signed numeric, each operating system defines its own specification. Therefore, when these data are transferred between different systems, it is necessary to transfer the data generally by transforming them into standard data form in order not to cause any problem, even if each system employs a different specification.

(As an exception, when both systems confirm that there is no difference in the specification between the two, there is no problem in exchanging the data transparently as binary.)

To cope with it, COBOL document type defines the data types in INTAP-AS2 as standard one. By defining and using a virtual file which has this data type as a file attribute, data exchange with standard data type becomes possible. That is to make each DU (Data Unit) in virtual file correspond to 1 logical record defined in COBOL program and also each data item to

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

each DE (Data Element). Besides defining the data type and length of each DE on document type parameter and keeping them as file attributes, data exchange is carried out.

The following figure C.2 shows an outline of data type mapping. In this figure, User 1 carries out file access to Virtual File A' on system 2. Assuming that Real File A on system 2 is actually accessed, the data of real file A, domestically through (1) and (2) paths, is transformed to the data type accessible on system 1 and processed in COBOL application which is User 1.

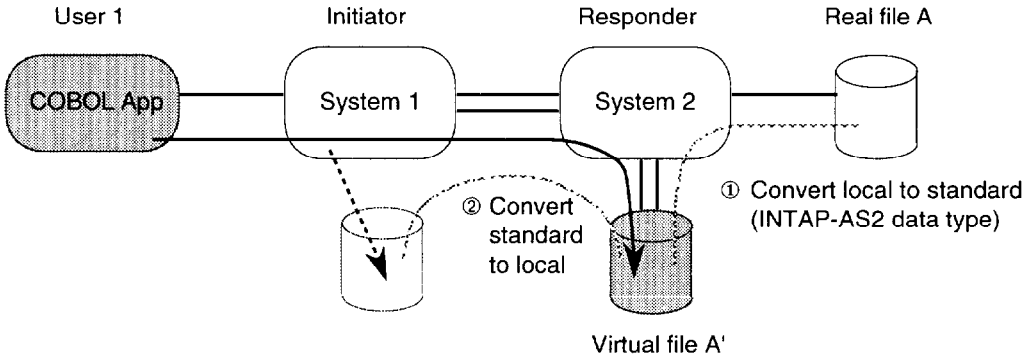


Figure C.2 - Outline of data type mapping

C.4.2 Method of mapping

(a) Correspondence to data type of INTAP-AS2

Table C.1 shows correspondence of data type defined in COBOL PICTURE clause with USAGE to data type of INTAP-AS2.

Table C.1 - Correspondence of data types

COBOL datatype		Contents	INTAP-AS2 datatype	Note
BINARY		binary digit	INTEGER	1
COMP		binary digit	INTEGER	1
DISPLAY	alphanumeric	alphanumeric	IA5, •••	
	numeric	decimal number	NumericType	2
PACKED-DECIMAL		decimal number	DecimalType	

Note 1: If numeric item is binary, the field size of numeric value domestically maintained in each system generally depends on the number of digits.

The maximum-string-length of INTEGER type data element(DE), a document type parameter, tells the information about field size of the value.

When the number of digits of the numeric item of the concerned binary is "n", the maximum-string-length of DE is:

$1 \leq n \leq 4$ ⇨ maximum-string-length = 2 octets
$5 \leq n \leq 9$ ⇨ maximum-string-length = 4 octets
$10 \leq n \leq 18$ ⇨ maximum-string-length = 8 octets

Note 2: In the Numeric Type defined in INTAP-AS2, there are five choices of types depending on whether it is signed or where it is located. These five types correspond to each COBOL SIGN clause as follows.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

NumericType of INTAP-AS2

```
NumericType ::=
    CHOICE {  unsign [PRIVATE 2] IMPLICIT PrintableString,                          see (1)
              trailing [PRIVATE 3] IMPLICIT PrintableString,                        see (2)
              trailing-with-separate-character [PRIVATE 4] IMPLICIT PrintableString, see (3)
              leading [PRIVATE 5] IMPLICIT PrintableString,                         see (4)
              leading-with-separate-character [PRIVATE 6] IMPLICIT PrintableString } see (5)
```

Correspondence of data types defined in COBOL;

(1)	:	PIC	9(n)	
(2)	:	PIC	S9(n)	[SIGN IS TRAILING]
(3)	:	PIC	S9(n)	SIGN IS TRAILING SEPARATE CHARACTER
(4)	:	PIC	S9(n)	SIGN IS LEADING
(5)	:	PIC	S9(n)	SIGN IS LEADING SEPARATE CHARACTER

(b) Example of mapping from data type of COBOL to INTAP-AS2

(1) Definition of data items in COBOL

```
FD FILE1
01  DATARECORD
    10  FIELDA
        20  NUMBER          PIC   S9 (6)     BINARY
        20  NAME            PIC    X (20)
    10  FIELDB
        20  SUBFIELD1
            30  TYPEA       PIC    9 (3)
            30  TYPEB       PIC    9 (3)
            30  TYPEC       PIC    9 (3)
            30  TOTAL1      PIC   S9 (4)
        20  SUBFIELD2
            30  TYPEX       PIC    9 (3)
            30  TYPEY       PIC    9 (3)
            30  TOTAL2      PIC   S9 (4)
        20  TOTALSUM        PIC   S9 (5)     PACKED-DECIMAL
    10  FILLER              PIC    X (27)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

(2) Figure of each data field defined above (figure C.3)

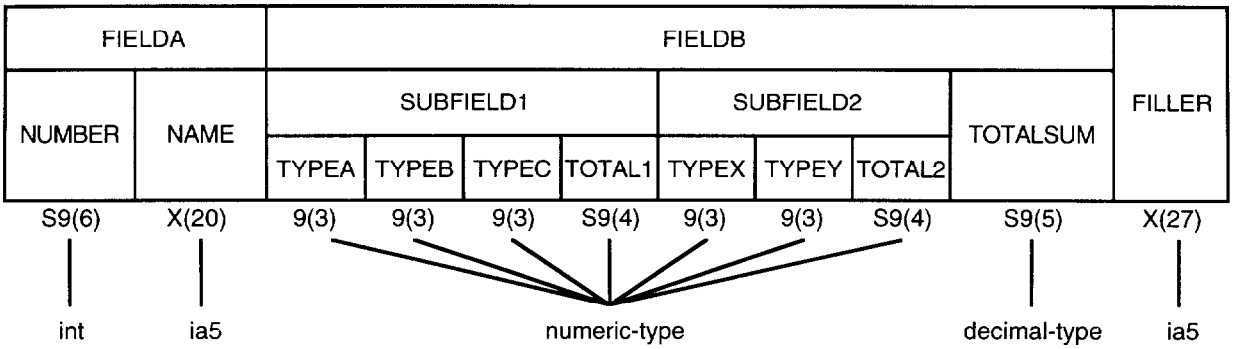


Figure C.3 - DATARECORD

(3) The parameter value of COBOL Document type

```
PARAMETERS : : =    {  fixed,    {
                    {  int,4  } ,                    ;  NUMBER
                    {  ia5,20  } ,                   ;  NAME
                    {  numeric-type,3,4  } ,         ;  TYPEA
                    {  numeric-type,3,4  } ,         ;  TYPEB
                    {  numeric-type,3,4  } ,         ;  TYPEC
                    {  numeric-type,4,5  } ,         ;  TOTAL1
                    ...
                    {  decimal-type,5,3  } ,         ;  TOTALSUM
                    {  ia5,27  }  }  }               ;  FILLER
```

C.4.3 Restriction

(a) The file which can be mapped to COBOL document type must have only one 01 level-number record description. If the user treats a record as an octet string, the user is able to map the record to a multiple of level-number records. The user is responsible for mapping a multiple of level-number records.

(b) The data item which is defined by the DEPENDING ON phrase of the OCCURS clause must be at the end of the record.

C.5 Details of sequential file, relative file and indexed file

The points which should be considered in the record access to the sequential file, the relative file or indexed file is achieved by using the COBOL document types as described.

1) All data replaced in the sequential file

In the file transfer (AFT12, AP.112), a replacement of all data of the sequential file can work according to the INTAP sequential flat constraint set. The initiator prepares the data which are the format when the READ is done with the access context of HA. The initiator transmits the REPLACE which points to the BEGIN. Then, all data in the virtual file can be replaced.

2) The record number of the relative file

The record number of the relative file exists in the leaf node of the ordered flat constraint set with unique names. The INTAP-Node-Name resides in the leaf node and the INTAP-Node-Name is expressed by the integer which is equivalent to the record number.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

3) Addition of the record to the relative file

The method of adding a record to the relative file depends on the restriction in the ordered flat constraint set with unique names. In regulations of the ordered flat constraint set with unique names, the data can be added to the virtual file by pointing at the BEGIN of the virtual file and executing the INSERT by the data of the same format when the user reads a record from the virtual file with the access context of FA. Because the node is included in the data of the format obtained with the access context of FA, it is necessary to specify the place where the record should be added to the relative file by record number. The place where the record should be added is the following number of the maximum record number of the virtual file. To know the maximum record number of the virtual file, the virtual file is accessed by the following sequence. First is to point to the END of the virtual file. Following is continuous and points at the PREVIOUS and reads the record with the access context of FA. Because FADU with the maximum record number is obtained by doing the READ, it is necessary to refer to the maximum record number in the node part of FADU.

4) The record key of the indexed file

The INTAP-5 uses the ordered flat constraint set with unique names and the INTAP-4 uses the ordered flat constraint set. In each constraint set, the record key is in the leaf node and the DU.

C.6 Mapping of FTAM service primitive and COBOL statement

C.6.1 Example of the sequential file

COBOL statement	service primitive	explanations
OPEN	F-OPEN	open the file
CLOSE	F-CLOSE	close the file
READ	F-READ F-DATA F-DATA-END F-TRANSFER-END	read 1 record from the file
REWRITE	F-WRITE (REPLACE) F-DATA F-DATA-END F-TRANSFER-END	rewrite 1 record in the file
WRITE	F-WRITE (INSERT) F-DATA F-DATA-END F-TRANSFER-END	write 1 record to the file

OPEN mode	processing mode
INPUT	READ
OUTPUT	INSERT/ERASE
I/O	READ/REPLACE
EXTEND	INSERT

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.6.2 Example of the relative file

COBOL statement	service primitive	explanations
OPEN	F-OPEN	open the file
CLOSE	F-CLOSE	close the file
READ	F-READ F-DATA F-DATA-END F-TRANSFER-END	read 1 record from the file
REWRITE	F-WRITE (REPLACE) F-DATA F-DATA-END F-TRANSFER-END	rewrite 1 record in the file
WRITE	F-WRITE (INSERT) F-DATA F-DATA-END F-TRANSFER-END	write 1 record to the file
START	F-LOCATE	locate the record of the file
DELETE	F-ERASE	delete 1 record from the file

OPEN mode	processing mode
INPUT	READ
OUTPUT	INSERT/ERASE
I/O	READ/REPLACE
EXTEND	INSERT

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.6.3 Example of the indexed file

COBOL statement	service primitive	explanations
OPEN	F-OPEN	open the file
CLOSE	F-CLOSE	close the file
READ	F-READ F-DATA F-DATA-END F-TRANSFER-END	read 1 record from the file
REWRITE	F-WRITE (REPLACE) F-DATA F-DATA-END F-TRANSFER-END	rewrite 1 record in the file
WRITE	F-WRITE (INSERT) F-DATA F-DATA-END F-TRANSFER-END	write 1 record to the file
START	F-LOCATE	locate the record of the file
DELETE	F-ERASE	delete 1 record from the file

OPEN mode	processing mode
INPUT	READ
OUTPUT	INSERT/ERASE
I/O	READ/REPLACE
EXTEND	INSERT

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.7 Example of the COBOL mapping

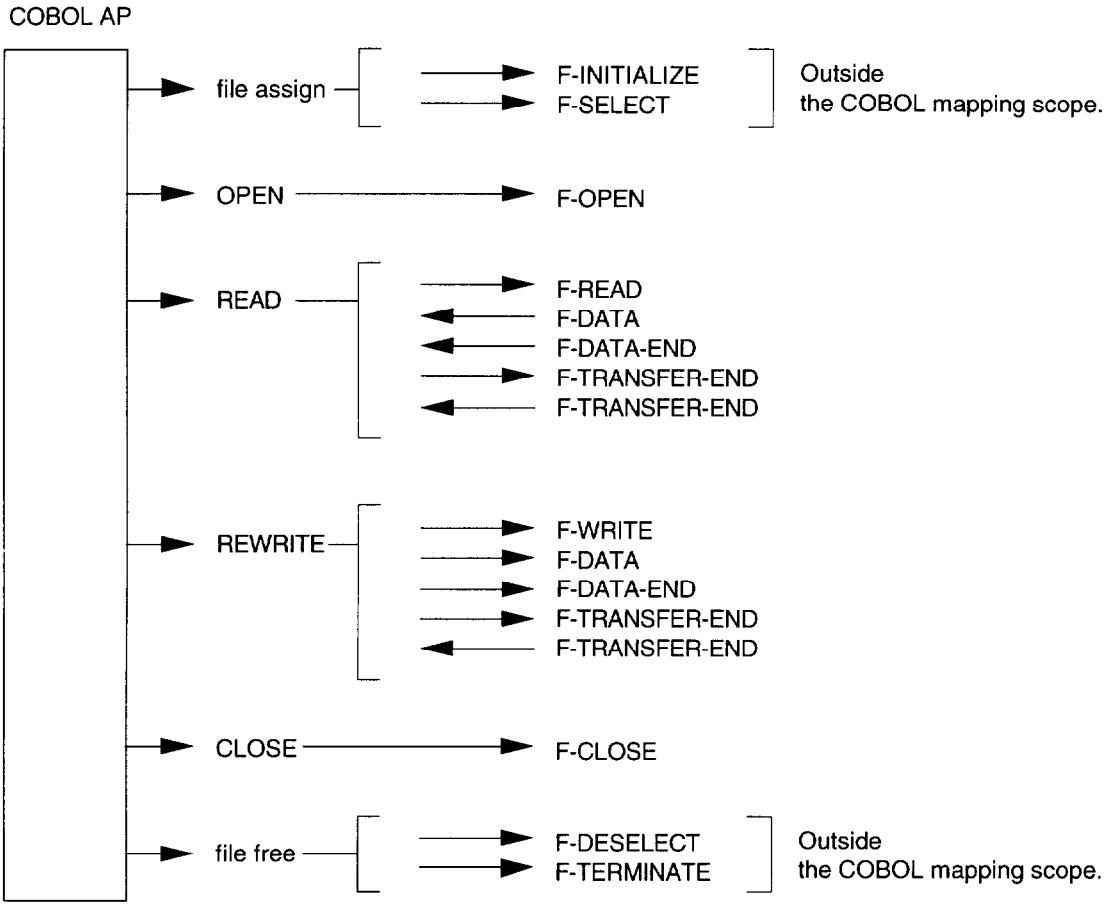


Figure C.4 - Example of COBOL mapping

The example performs the following mappings:

- In case of mapping the OPEN statement to F-OPEN, the association establishment and file selection must be performed by the trigger except COBOL statements.
- In case of mapping the CLOSE statement to F-CLOSE, the file deselection and the association termination must be performed by the trigger except COBOL statements.
- In case of OPEN with OUTPUT mode, and when the file is selected using an F-SELECT, the entire file must be erased before new data is written to the file.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

© ISO/IEC

ICS 35.100

Descriptors: data processing, information interchange, network interconnection, open systems interconnection, data transmission, files, access, management, communication procedure, profiles.

Price based on 68 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale