

AS/NZS 3646.1:1993

ISO 8571-1:1988

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

ISO 8571-1:1988/Amdt 1:1992

Australian/New Zealand Standard

**Information processing systems—
Open Systems Interconnection—
File transfer, access and
management**

Part 1: General introduction

STANDARDS AUSTRALIA

STANDARDS NEW ZEALAND



Documents

Document Sharing Hub

AS/NZS 3646.1:1993

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT/1, Information Systems—Interconnection. It was approved on behalf of the Council of Standards Australia on 17 May 1993 and on behalf of the Council of Standards New Zealand on 26 July 1993. It was published on 16 August 1993.

The following interests are represented on Committee IT/1:

AOTC

Australian Association of Permanent Building Societies
 Australian Bankers Association
 Australian Bureau of Statistics
 Australian Chamber of Commerce and Industry
 Australian Computer Society
 Australian Computer Users Association
 Australian Information Industry Association
 Australian Telecommunications Users Group
 Australian Vice Chancellors Committee
 Department of Defence
 Department of Industry, Technology and Commerce
 Information Exchange Steering Committee
 Institute of Information and Communication Technologies, CSIRO
 Life Insurance Federation of Australia
OPTUS
 Standards New Zealand

Review of Standards. To keep abreast of progress in industry, Joint Australian/New Zealand Standards are subject to periodic review and are kept up to date by the issue of amendments or new editions as necessary. It is important therefore that Standards users ensure that they are in possession of the latest edition, and any amendments thereto.

Full details of all Joint Standards and related publications will be found in the Standards Australia and Standards New Zealand Catalogue of Publications; this information is supplemented each month by the magazines 'The Australian Standard' and 'Standards New Zealand', which subscribing members receive, and which give details of new publications, new editions and amendments, and of withdrawn Standards.

Suggestions for improvements to Joint Standards, addressed to the head office of either Standards Australia or Standards New Zealand, are welcomed. Notification of any inaccuracy or ambiguity found in a Joint Australian/New Zealand Standard should be made without delay in order that the matter may be investigated and appropriate action taken.



Documents

Document Sharing Hub

AS/NZS 3646.1:1993

Australian/New Zealand Standard

Information processing systems— Open Systems Interconnection— File transfer, access and management

Part 1: General introduction

First published as Joint Standard AS/NZS 3646.1:1993.

PUBLISHED JOINTLY BY:

STANDARDS AUSTRALIA
1 The Crescent,
Homebush NSW 2140 Australia

STANDARDS NEW ZEALAND
6th Floor, Wellington Trade Centre,
181-187 Victoria Street,
Wellington 1 New Zealand

PREFACE

This Standard was prepared by the Joint Australia/New Zealand Standards Committee IT/1 on Information Systems—Interconnection. It is identical with, and has been reproduced from, ISO 8571-1:1988, *Information processing systems—Open Systems Interconnection—File Transfer, Access and Management*, Part 1: *General introduction*.

Technical Corrigendum 1 and Amendment 1 have been bound at the end of this Standard. Text affected is marked by a marginal bar.

This Joint Standard is one of a series of Open Systems Interconnection (OSI) Standards which are currently under development. Since OSI Standards are developmental, there may be some minor difficulties encountered in their implementation. For this reason, Standards Australia will be providing, through the OSI Help Desk, a service to coordinate and disseminate information concerning difficulties which are identified in using this Joint Standard.

Under arrangements made between Standards Australia/Standards New Zealand and the international Standards bodies, ISO and IEC, as well as certain other Standards organizations, users of this Joint Standard are advised of the following:

- (a) In Australia, copyright is vested in Standards Australia. In New Zealand, copyright is vested in Standards New Zealand.
- (b) The number of this Joint Standard is not reproduced on each page; its identity is shown only on the cover and title pages.

For the purpose of this Joint Standard, the ISO text should be modified as follows:

- (i) *Terminology* The words ‘Australian Standard’, ‘New Zealand Standard’ or ‘Joint Australian/New Zealand Standard’ should replace the words ‘International Standard’ wherever they appear.
- (ii) *References* The references to International Standards should be replaced by references to the following Australian/New Zealand Standards:

Reference to International Standard or other Publication		Australian/New Zealand Standard	
ISO		AS	
7498	Information Processing Systems— Open Systems Interconnection— Basic Reference Model	2777	Information processing systems— Open Systems Interconnection— Basic reference model
8326	Information Processing Systems— Open Systems Interconnection— Basic Connection Oriented Session Service Definition	3591	Information processing systems— Open Systems Interconnection— Basic connection oriented session service definition
8571	Information Processing Systems— Open Systems Interconnection— File transfer, access and management	3646	Information processing systems— Open Systems Interconnection— File transfer, access and management
8571-2	Part 2: Virtual Filestore definition	3646.2	Part 2: Virtual filestore definition
8571-3	Part 3: File Service definition	3646.3	Part 3: File service definition
8571-4	Part 4: File Protocol specification	3646.3	Part 4: File protocol specification
8649	Information Processing Systems— Open Systems Interconnection— Service definition for the Association Control Service Element	3683	Information processing systems— Open Systems Interconnection— Service definition for the Association Control Service Element

8822	Information Processing Systems— Open Systems Interconnection— Connection-oriented Presentation Service Definition	3615	Information processing systems— Open Systems Interconnection— Connection oriented presentation service definition
8824	Information Processing Systems— Open Systems Interconnection— Specification of Abstract Syntax Notation One (ASN.1)	3625	Information technology— Open Systems Interconnection— Specification of Abstract Syntax Notation One (ASN.1)
8825	Information Processing Systems— Open Systems Interconnection— Specification of Basic encoding rules for Abstract Syntax Notation One (ASN.1)	3626	Information technology— Open Systems Interconnection— Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)
9804	Information Processing Systems— Open Systems Interconnection— Definition of Application Service Elements—Commitment, Concurrency and Recovery	AS/NZS 4106	Information technology— Open Systems Interconnection— Service definition for the commitment, concurrency and recovery service element
9834	Information Processing Systems— Procedures for specific OSI registration authorities—	—	
9834-2	Part 2: Registration of Document Types		
ISO/TR			
8509	Information Processing Systems— Open Systems Interconnection— Service Conventions	3620	Information processing systems— Open Systems Interconnection— Service conventions

© Copyright – STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

Users of Standards are reminded that copyright subsists in all Standards Australia and Standards New Zealand publications and software. Except where the Copyright Act allows and except where provided for below no publications or software produced by Standards Australia or Standards New Zealand may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from Standards Australia or Standards New Zealand. Permission may be conditional on an appropriate royalty payment. Australian requests for permission and information on commercial software royalties should be directed to the head office of Standards Australia. New Zealand requests should be directed to Standards New Zealand.

Up to 10 percent of the technical content pages of a Standard may be copied for use exclusively in-house by purchasers of the Standard without payment of a royalty or advice to Standards Australia or Standards New Zealand.

Inclusion of copyright material in computer software programs is also permitted without royalty payment provided such programs are used exclusively in-house by the creators of the programs.

Care should be taken to ensure that material used is from the current edition of the Standard and that it is updated whenever the Standard is amended or revised. The number and date of the Standard should therefore be clearly identified.

The use of material in print form or in computer software programs to be used commercially, with or without payment, or in commercial contracts is subject to the payment of a royalty. This policy may be varied by Standards Australia or Standards New Zealand at any time.

CONTENTS

	<i>Page</i>
0 Introduction	7
1 Scope and field of application	7
2 References	7
3 Reference model definitions	7
4 Service conventions definitions	8
5 FTAM definitions	8
5.1 General	8
5.2 Architectural	8
5.3 Filestore schema	9
5.4 Filestore access	9
5.5 File structure	9
5.6 Constraint set	9
5.7 Document types	10
6 Abbreviations	10
Section one: FTAM general concepts	
7 OSI architectural background	11
8 Nature of the file service	12
8.1 Control of file activity	12
8.2 Asymmetry of the dialogue	12
8.3 External file service and internal file service	12
8.4 Service classes and functional units	13
9 Functions associated with the file service	13
9.1 Control of actions	13
9.2 Accounting	14
9.3 Concurrency control	14
9.4 Access control	15
9.5 Commitment	16
10 Service providers supporting FTAM	16
10.1 ACSE - application contexts and the FTAM environment	16
10.2 Presentation Service	16
10.3 Session Service	16
Section two: Virtual Filestore — General Concepts	
11 Virtual filestore	18

	<i>Page</i>
11.1 Need for a filestore model	18
11.2 Mapping the virtual filestore definition	18
11.3 Form of the virtual filestore	19
11.4 Attribute dynamics	20
11.5 Filestore schema	20
12 File structures	20
12.1 Categories of Structure	20
12.2 File access structure	21
12.3 Presentation structure	21
13 Constraint sets	22
14 Document types	22
Section three: Overview of the file service and file protocol	
15 File service	23
15.1 FTAM regime initialization phase	23
15.2 Filestore management phase	23
15.3 File selection phase	23
15.4 File management phase	24
15.5 File open phase	24
15.6 Data access phase	24
15.7 File close phase	24
15.8 File deselection phase	24
15.9 FTAM regime termination phase	24
16 Mechanisms in the file protocol	24
16.1 Protocol state machine	24
16.2 Grouping of protocol data units	24
16.3 Transparency	24
16.4 Checkpoint insertion	25
16.5 Diagnostics and results	25
16.6 Docket handling and non-volatile storage	25
16.7 Error recovery mechanisms	25
Annexes	
A Examples of the use of FTAM	26
B Summary of information objects identified in ISO 8571	31

Figures	<i>Page</i>
1 Flow of information between RSE and OSIE.....	11
2 Logical and actual flows of information in file transfer	12
3 Example of the dialogue between file entities.....	12
4 Structuring a file protocol entity	13
5 Restrictions on the possible actions	14
6 Mapping between real systems and open systems	18
7 Virtual filestore schema	19
8 An example of a file access structure	20
9 Transmission of an unstructured text file	21
10 File service regimes and related primitives.....	23
11 Sending a file to a remote system	27
12 Remote database access	28
13 Use of FTAM in a LAN fileserver	29
14 File management.....	30

Information processing systems—Open Systems Interconnection—File transfer, access and management

Part 1: General introduction

0 Introduction

ISO 8571 is one of a set of International Standards produced to facilitate the interconnection of computer systems. Its relation to other International Standards in the set is defined by the Reference Model for Open Systems Interconnection (ISO 7498). The Reference Model subdivides the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The aim of Open Systems Interconnection is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of computer systems:

- a) from different manufacturers,
- b) under different managements,
- c) of different levels of complexity,
- d) of different ages.

ISO 8571 defines a file service and specifies a file protocol available within the application layer of the Reference Model. The service defined is of the category Application Service Element (ASE). It is concerned with identifiable bodies of information which can be treated as files, and may be stored within open systems or passed between application processes.

ISO 8571 defines a basic file service. It provides sufficient facilities to support file transfer, and establishes a framework for file access and file management. ISO 8571 does not specify the interfaces to a file transfer or access facility within the local system.

ISO 8571 consists of the following four parts:

- Part 1: General introduction
- Part 2: Virtual Filestore definition
- Part 3: File Service definition
- Part 4: File Protocol specification

The definitions in this part of ISO 8571 are used in the subsequent parts of ISO 8571 which specify the virtual filestore, services and protocols.

This part of ISO 8571 contains the following annexes which do not form part of the standard:

- Annex A - Examples of the use of FTAM
- Annex B - Summary of objects identified

1 Scope and field of application

This part of ISO 8571 provides a general introduction to the concepts and mechanisms specified in parts 2 to 4 of ISO 8571.

2 References

ISO 7498, *Information Processing Systems - Open Systems Interconnection - Basic Reference Model*.

ISO 8326, *Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Service Definition*.

ISO/TR 8509, *Information Processing Systems - Open Systems Interconnection - Service Conventions*.

ISO 8571, *Information Processing Systems - Open Systems Interconnection - File transfer, access and management*.

- Part 2: Virtual Filestore definition.
- Part 3: File Service definition.
- Part 4: File Protocol specification.

ISO 8649, *Information Processing Systems - Open Systems Interconnection - Service definition for the Association Control Service Element*.

ISO 8822, *Information Processing Systems - Open Systems Interconnection - Connection-oriented Presentation Service Definition*.

ISO 8824, *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*.

ISO 8825, *Information Processing Systems - Open Systems Interconnection - Specification of Basic encoding rules for Abstract Syntax Notation One (ASN.1)*.

ISO 9804, *Information Processing Systems - Open Systems Interconnection - Definition of Application Service Elements - Commitment, Concurrency and Recovery*.¹⁾

ISO 9834-2, *Information Processing Systems - Procedures for specific OSI registration authorities - Part 2: Registration of Document Types*.¹⁾

3 Reference model definitions

ISO 8571 is based on the concepts developed in ISO 7498 and makes use of the following terms defined in it:

- a) application-entity;
- b) application process;
- c) application service element;
- d) (N)-connection;
- e) open system;
- f) (N)-protocol;

¹⁾ At present at the stage of draft; publication anticipated in due course.

- g) (N)-protocol-control-information;
- h) (N)-protocol-data-unit;
- i) (N)-service-access-point;
- j) (N)-service-access-point-address;
- k) (N)-service-data-unit;
- l) sublayer;
- m) (N)-user-data;
- n) user element.

4 Service conventions definitions

ISO 8571 makes use of the following terms defined in ISO/TR 8509 as they apply to the file service:

- a) confirm;
- b) indication;
- c) primitive;
- d) request;
- e) response;
- f) service provider;
- g) service user.

5 FTAM definitions

Unless otherwise stated, all terms apply to the view of a system presented for the purpose of open interconnection. This implies that the terms relate to a virtual filestore rather than to any real filestore (see clause 7).

The definitions are grouped into major categories, and ordered alphabetically within each category.

For the purpose of ISO 8571, the following definitions apply:

5.1 General

5.1.1 empty file: A file whose file contents consist of only a root node with no associated data unit, and no node name.

5.1.2 file access: The inspection, modification, replacement or erasure of part of a file's contents.

5.1.3 file contents: The data units, node names and structuring information contained in the file, which may be manipulated during the file open regime; the file attributes do not form part of the file's contents.

5.1.4 file management: The creation and deletion of files, and the inspection or manipulation of the file attributes.

5.1.5 file transfer: A function which moves a part or the whole of a file's contents between open systems.

5.1.6 hierarchical file model: A model of the internal structure of a file which takes the form of a tree of nameable file access data units.

5.1.7 real file: The named collection of information and its attributes which reside in a real system and to which the references to virtual files made in the OSIE are mapped.

5.1.8 real filestore: An organized collection of files, including their attributes and names, which reside in a real system and to which the virtual file references in the OSIE are mapped.

5.1.9 virtual file; file: An unambiguously named collection of structured information having a common set of attributes.

5.1.10 virtual filestore: An abstract model for describing files and filestores, and the possible actions on them. Where no ambiguity exists, the term is shortened in ISO 8571 to "filestore".

5.2 Architectural

5.2.1 accounting regime: The period during which a particular set of accounting information applies.

5.2.2 commitment unit: A set of filestore actions which either succeeds, the effect then being made visible to other processes, or fails completely with no effect visible to other processes.

NOTE - Nullification of the actions in a commitment unit is possible at any time up to the completion of the commitment unit.

5.2.3 docket: That collection of information which can be associated with a file service regime and which must be preserved if error recovery is to be possible.

5.2.4 external file service: File Transfer, Access and Management as seen by the file service user.

5.2.5 file service user: That portion of the application entity which conceptually invokes the FTAM service.

5.2.6 initiator: That file service user which requests FTAM regime establishment.

5.2.7 internal file service: The service used by the file error recovery protocol machine to transmit both file error recovery protocol control information and normal file protocol control information.

5.2.8 open systems interconnection environment: The set of definitions of the standardized services, protocols and data structures which enable the interconnection of systems.

5.2.9 phase: The period of time in which protocol exchanges have a particular purpose, such as establishing or releasing an application context; for each phase a set of valid messages is defined in terms of state transitions.

NOTE - An entity is in one phase at any time.

5.2.10 presentation data value: The unit of information, specified at the abstract syntax level, which is transferred by the presentation service.

5.2.11 real system environment: The implementation aspects which support an application process within a real system.

5.2.12 receiving entity; receiver: The entity which receives part or all of the file's contents during the file data transfer regime.

5.2.13 regime: The period during which the entity is in a subset of its possible states for which particular actions are permitted.

NOTE - Regimes may be nested.

5.2.14 responder: That file service user which accepts an FTAM regime establishment requested by the initiator.

5.2.15 rollback: The nullification of uncommitted actions.

5.2.16 sending entity; sender: The entity which sends part or all of the file's contents during the file data transfer regime.

5.2.17 service element: A unit of standardization specifying a complete group of functions.

5.2.18 service primitive: The smallest defined interaction between the user and the provider of a communication service.

5.2.19 symbiotic service element: A service element which will support the operation of some second service by adopting its semantics and including parts of its abstract syntax at defined points in the abstract syntax of the first service's protocol control information.

5.3 Filestore schema

5.3.1 activity attributes: The attributes describing the activity of using the file service. The attributes are local to one FTAM regime (and any regime nested within it).

5.3.2 attribute: A piece of information stating a property of something, taking one of a set of defined values, each value having a defined meaning.

5.3.3 file attributes: The name and other identifiable properties of a file.

NOTE - The same value of a file attribute is observed at a particular time by any user of the file service, even when more than one user is active at that time.

5.4 Filestore access

5.4.1 access context: The specification of an algorithm defining a subset of the structuring information and user information in a file's contents, when reading the file for transfer or access.

5.4.2 data element: The smallest piece of data whose identity is necessarily preserved when transferred by the Presentation Service. A data element can convey file contents information, file structuring information or protocol control information.

5.4.3 data unit: The smallest unit of a file's contents which the filestore actions can manipulate. Each data unit is associated with a node of the file access structure. A data unit is a series of data elements.

5.4.4 file access data unit: A unit of the file access structure on which the actions of transfer, deletion, extension, replacement or insertion can be performed. A file access data unit contains zero or more data units.

5.4.5 filestore action: One of the actions specified as part of the definition of the virtual filestore.

5.4.6 file access structure: The data structure of a file that relates the file access data units, allowing their identification, description and manipulation.

5.5 File structure

5.5.1 arc: A directed link between two nodes.

5.5.2 arc length: A positive integer expressing the difference in levels between a child node and its parent node.

5.5.3 child (of a node): A node at which an outbound arc of the node concerned terminates.

5.5.4 leaf: A node of a tree that has no outbound arcs.

5.5.5 level (of a node): The sum of the arc lengths from the root to the node concerned.

5.5.6 long arc: An arc with an arc length greater than one.

5.5.7 node: The elementary component from which a tree is built up.

5.5.8 ordered tree: A tree in which there is a defined ordering of the children of each node in the tree.

5.5.9 parent (of a node): The node from which the inbound arc of the node concerned originates.

5.5.10 path: A sequence of arcs which links one node to another node in such a way that each arc is traversed in its defined direction.

5.5.11 root: The unique node of a tree that has no inbound arcs; it is at level 0.

5.5.12 sister (of a node): A node which shares the same parent node as the node concerned.

5.5.13 subtree: A part of a tree comprising an arbitrary node as the subtree root node and all the other nodes which can be reached by a path from the subtree root node.

5.5.14 traversal sequence: An ordering of the nodes in a tree such that each node occurs once and only once, and which is determined by an algorithm applicable to all possible trees.

NOTE - In general, many different trees may generate the same traversal sequence.

5.5.15 tree: A connected structure in which each node is linked to other nodes by directed arcs in such a way that one node has no inbound arcs and all other nodes have exactly one inbound arc.

5.6 Constraint set

5.6.1 constraint set: A set of restrictions and refinements of a general file model which specifies a less general model tailored to the needs of a particular class of applications.

5.6.2 file model: A model of the access structure of a file's contents.

5.6.3 flat (constraint set): A constraint set which, when applied to the general hierarchical file model, generates an access structure that consists of two levels, at the levels zero and one, and that may have data units at only the leaf nodes and has no data unit at the root node.

5.6.4 general hierarchical file model: a model in which the file access data units are organized in a hierarchical tree.

5.6.5 hierarchical (constraint set): A constraint set which, when applied to the general hierarchical file model, generates an access structure which is still hierarchical, but in which the form of the node descriptions and data units is restricted.

5.6.6 unstructured (constraint set): A constraint set which, when applied to the general hierarchical file model, generates an access structure that consists only of the root node with one data unit.

5.7 Document types

5.7.1 concatenation (of documents): The combination of two documents to form a single resultant document.

5.7.2 document: A collection of information with known abstract syntaxes and partially known semantics, and a known set of possible transfer syntaxes.

5.7.3 document type: The specification of a class of documents, which states their necessary semantics, abstract syntaxes, transfer syntaxes and dynamics.

5.7.4 dynamics (of a document): The concatenation and simplification properties of a document.

5.7.5 relaxation (of a document): The process of deriving one document from another by making the parameters describing it less restrictive.

5.7.6 simplification (of a document): The process of deriving one document from another of a different type by discarding structural information.

6 Abbreviations

The following abbreviations are used in parts 2 to 4 of ISO 8571:

ACSE	association control service element
ASE	application service element
CCR	commitment, concurrency and recovery
DU	data unit
EFS	external file service
FADU	file access data unit
FERPM	file error recovery protocol machine
FPDU	file protocol data unit
FPM	file protocol machine
FTAM	file transfer, access and management
Id	identifier
IFS	internal file service
OSI	open systems interconnection
OSIE	open systems interconnection environment
PCI	protocol control information
PDU	protocol data unit
PDV	presentation data value
PSDU	presentation service data unit
RSE	real system environment

Section one: FTAM general concepts

7 OSI architectural background

The aim in the standardization of a file service is to allow the Open Systems Interconnection of file users, who wish to transfer, access or manage information held by systems which behave as if they store files. Anything which appears as an open system, and conforms to the specified file protocols in the role of responder, is considered to provide a virtual filestore.

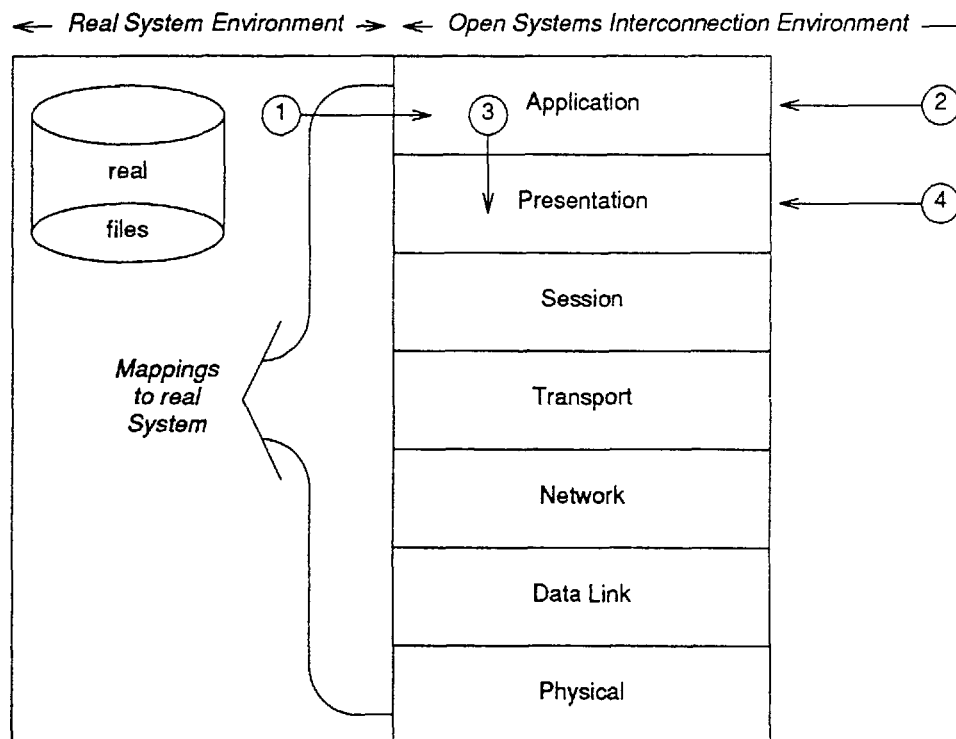
As in all the Open Systems Interconnection standards, the architectural divisions used here reflect classifications of protocol. They do not in any way constrain the way that possible underlying implementations are to be structured. The only point at which conformance can be tested is the point of interconnection.

The various parts of ISO 8571 are part of a larger set of Open Systems Interconnection standards, which are interrelated by the OSI Basic Reference Model (ISO 7498). The aspects of interconnection protocols common to many different fields of use are defined in separate standards, and the logical relation of the various elements is provided by supporting definitions of the services available.

In understanding the intent of OSI, it is important to distinguish carefully between the interrelated set of standards and the implementation, in hardware and software, of a real open system using the protocols specified in the standards. This distinction leads to the identification of two environments:

a) the implementation aspects, in terms of the real facilities and resources which support an application process within a real system, constitute the Real System Environment (RSE);

b) the set of definitions of the standardized services and protocols and data structures which enable the interconnection of systems constitute the Open Systems Interconnection Environment (OSIE). The stylized aspects of the behaviour of an application process visible in the Open Systems Interconnection Environment constitute the application entity; the activities of the application entities are supported by the layered communication functions defined in the OSI reference model.



NOTES

- 1 The flow of activations between the RSE and the OSIE represents the mapping of the intention to communicate onto the application entity user element.
- 2 The flow between application entities (both the application PCI and any user data) is a logical flow of information in an agreed abstract syntax, which is independent of any particular transfer syntax.
- 3 The flow between the application and presentation entities is a logical flow of information, including both user data and application protocol control information in one or more agreed presentation contexts.
- 4 The flow between the presentation entities is of data encoded in the negotiated transfer syntaxes, and of presentation protocol control information.

Figure 1 — Flow of information between RSE and OSIE

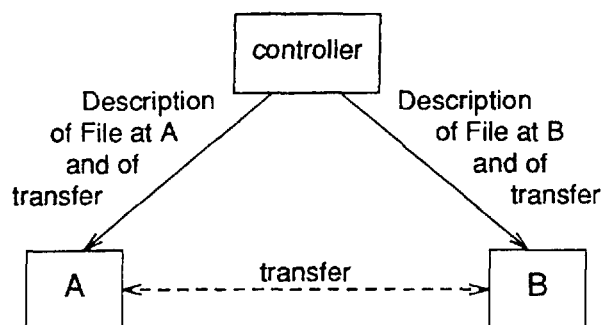


Figure 2a - Logical flows of information in file transfer

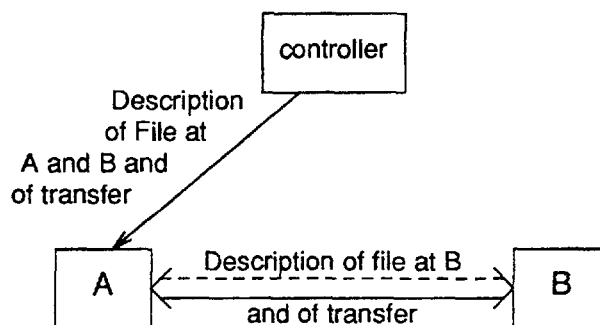


Figure 2b - Actual flows of information in FTAM

Figure 2 — Logical and actual flows of information in file transfer

Part of the work to be done in designing an implementation is the choice of a mapping between terms in all levels of the set of definitions and aspects of the implementation. This process is shown in figure 1.

8 Nature of the file service

8.1 Control of file activity

The way file activities are controlled needs to be explained in order to clarify the aims of ISO 8571. Consider a file transfer or file access; for any transfer or access, there are three entities involved: an entity which takes the controlling initiative, an entity which accesses the source virtual file and an entity which accesses the destination virtual file. From the controller, there are two flows of information:

- information, concerned with the specification of the source virtual file and with constraints on the way the transfer is to be performed, sent to the entity accessing the source file, and
- information, concerned with the specification of the destination virtual file and with constraints on the way the transfer is to be performed, sent to the entity accessing the destination file.

These flows and their relation to the transfer are indicated in figure 2a.

To simplify the co-ordination and control of the transfer, it is assumed that the controller will channel these flows via one of the two file protocol entities, which will act as its agent in performing the transfer (see figure 2b). In many cases this will arise naturally, because the controller and the initiating file entity will be within the same system.

The service defined here is the one which supports the interconnection of the elements at A and B in figure 2b, and A and B are the users of the service referred to in the service definition.

8.2 Asymmetry of the dialogue

The actions to be supported by the file protocol show some important asymmetries which are reflected in the service

and protocol structures. The asymmetries take the form of master to slave relationships.

Firstly, each activity is started by one file service user (the initiator, A in figure 2b), who has some established aim to achieve. The entity associated with the filestore (the responder, B in figure 2b) merely reacts to this initiative in a passive role. This applies even when a file is being transferred from one filestore to another, because the protocol does not need to carry information about the filestore at the initiator. The file protocol carries only information about the filestore at the responder. The act of transferring file access data units to the filestore at the responder can be considered as being performed by a copying application which has local access to one file and remote access to the other (see figure 3).

The second asymmetry is the more basic one that, when transferring file access data units, one particular entity is the sender, and the other is the receiver; at any instant during data transfer there is a preferred direction of data flow.

8.3 External file service and internal file service

The communication identified between the initiator and the responder in 8.2 can itself be divided in a modular fashion by separating out the error recovery mechanisms, yielding two distinct service levels (see figure 4).

These are

- an external file service, in which the user states its quality of service requirements, but has no awareness of error recovery, delegating such considerations to the service provider. Transfer of file data is modelled in the external file service as a series of error-free operations. Thus, within the external file service there is no visibility of recoverable errors or the error recovery actions.
- an internal file service used by the file error recovery protocol machine. This service includes primitives giving its users facilities for error recovery and control of the checkpointing mechanisms. The protocol specification which relates the external to the internal file service therefore consists of a standard set of procedures for error recovery and the protocol machine which executes

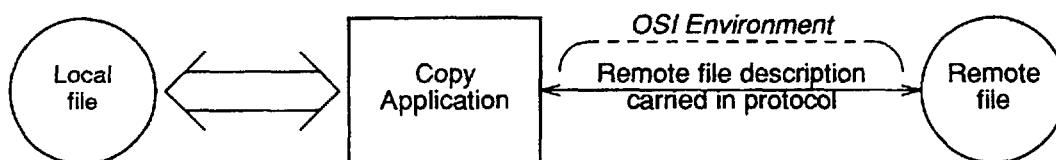


Figure 3 — Example of the dialogue between file entities

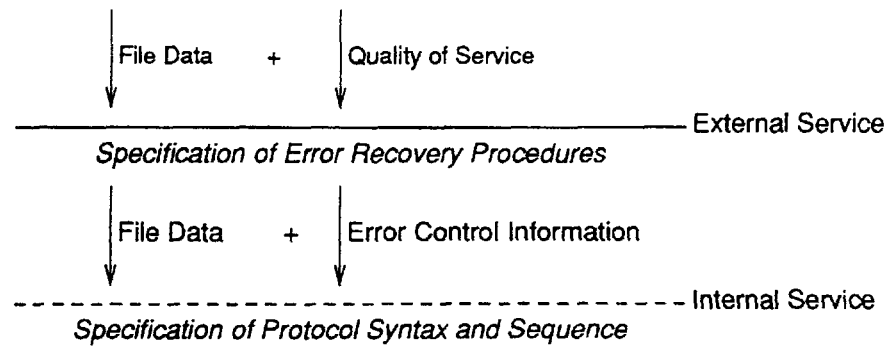


Figure 4 — Structuring a file protocol entity

these procedures is the user of the internal file service. The choice of the error recovery procedures to be used is based on analysis of cost from the FTAM and Communication Qualities of Service requested in the external service and on local management information.

8.4 Service classes and functional units

The FTAM protocol and service are very rich in function. A wide range of types of application can be supported, but the cost of implementing all parts of the protocol on all occasions would be correspondingly high.

If implementors were given arbitrary choice as to which of the functions were to be provided, there would, on the other hand, be little chance that similar choices would be made, and so little probability of communication.

To avoid these problems, the FTAM service defines two types of functional selection. At the most basic level, the functions defined are grouped into functional units. An implementation must support a functional unit either completely or not at all; the number of choices to be made are thus limited. Mechanisms are defined which allow negotiation of functional units when the FTAM regime is initialized, and these allow the two communicating entities to reach a common understanding of the set of functional units available.

This in itself reduces variety, but still leaves considerable freedom. Further convergence is achieved by defining service classes, each of which support broad categories of use. These classes are

- a) the transfer class, which allows for the movement of files or parts of files between systems, placing emphasis on simple operation with a minimum of protocol overhead before and after the data transfer;
- b) the management class, which allows control of the virtual filestore by a series of independent confirmed service exchanges, but does not include file transfer mechanisms.
- c) the transfer and management class, which combines the features of the transfer class and the management class;
- d) the access class, which allows the initiating entity to perform a sequence of operations on the file access data units, providing for the manipulation of remote data.
- e) the unconstrained class, which leaves the selection of functional units to the designer of the distributed application, giving full flexibility for optimization, but no guarantee of a common functional kernel.

The classes of service which are needed are negotiated when the FTAM regime is initialized.

9 Functions associated with the file service

9.1 Control of actions

The virtual filestore defines the actions which can be performed on a file. In any particular situation, only a subset of these actions are available. This subset is determined by

- a) the file attributes (see section two); these indicate the actions appropriate to the file contents (in terms of the constraint set applied) and the local storage mechanisms (by means of the permitted actions file attribute) and express any access control constraints affecting file access;
- b) the current state of the filestore, particularly the constraints implied by any concurrent access in progress to the same file;
- c) the values of the activity attributes established by parameters of the file service when the data transfer regime was being negotiated.

Each of these can restrict the set of actions available. The process of refinement, leading to the set of actions which can actually be performed, is illustrated by the Venn diagrams in figure 5.

The first diagram shows that the actions appropriate to a file are those which are allowed by the file structure as expressed by the structural constraint set, the permitted actions attribute and the access control attribute. The result is a set of actions which could in principle be performed on the file.

The resultant set of actions is then further reduced to take account of any system constraints applying to the filestore; the most important of these for file access are the constraints imposed by the concurrent activities of other users.

Finally, the initiator requests a set of actions while building up the data transfer regime, and at each stage the negotiation of parameters may result in a restriction either of the set of actions requested, or of the set the system would allow, or both. When the association is initialized, actions are limited by the service class and the set of functional units negotiated. When the file is selected, a subset of these actions may result from the permitted actions agreed. Lastly, when the file is opened, the actions to be used are declared in the stated processing mode.