INTERNATIONAL STANDARD

ISO 8571-1

First edition
1988-10-01

AMENDMENT 1
1992-12-15

Information processing systems – Open Systems Interconnection – File Transfer, Access and Management –

Part 1 : General introduction

AMENDMENT 1 : Filestore Management

Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Transfert, accès et gestion de fichiers –

Partie 1 : Introduction générale

AMENDEMENT 1 : Gestion du système de fichiers

Reference number
ISO 8571-1:1988/Amd.1:1992 (E)

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 and 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 JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 1 to International Standard ISO 8571-1:1988 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

ISO 8571-1 consists of the following parts, under the general title *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management*

- *Part 1 : General introduction*
- *Part 2 : Virtual Filestore Definition*
- *Part 3 : File Service Definition*
- *Part 4 : File Protocol Specification*
- *Part 5 : Protocol Implementation Conformance Statement Proforma*

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Information processing systems – Open Systems Interconnection – File Transfer, Access and Management –

Part 1 : General introduction

AMENDMENT 1 : Filestore Management

NOTE – This amendment has additional subclauses and tables to ISO 8571 which are indicated by the use of lower case Roman letters beginning with "a" and imply ordering alphabetically, following the clause with the same numerical value in ISO 8571. These and all subsequent subclauses, tables, and cross references will be renumbered in subsequent editions.

0 Introduction

(amend 3rd paragraph, page 1)

ISO 8571 defines services for file transfer, access and management. It also specifies a 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, stored and managed within open systems, or passed between application processes.

(amend 4th paragraph, page 1)

ISO 8571 defines a basic file service. It provides sufficient facilities to support file transfer, file access, and management of files stored on open systems. ISO 8571 does not specify the interfaces to a file transfer, access or management facility within the local system.

5 FTAM definitions

(append after clause 5, page 4)

5.8 Filestore Management

5.8.1 object

A file, file-directory, or reference.

5.8.2 file-directory

An object that provides a mechanism for the logical grouping of files, references, and file-directories.

5.8.3 reference

An object that provides a mechanism for the logical linkage of an apparent location in the filestore structure with a file or a file-directory in the filestore structure.

5.8.4 parenthood

An ordered relation between a file-directory object and another object. The file-directory object is the superior object in the relation, and the other object is the inferior object.

5.8.5 childhood

The dual relation to the parent relation.

5.8.6 linkage

An ordered relation between a reference object and a single file or file-directory object. The reference object is the superior object in the relation, and the other object is the inferior object. Every reference object has exactly one inferior object in the linkage relation. A file or file-directory object may have zero or more superiors in the linkage relation.

A reference maintaining a linkage relation to an object is said to ‘refer’ to the object. The object being referred to is called the referent of the relation.

5.8.7 parent file-directory

The superior object of two objects related by parenthood, if no other object lies strictly between the two objects under parenthood.

5.8.8 child object

The inferior object of two objects related by parenthood, if no other object lies strictly between the two objects under childhood.

5.8.9 filestore structure

The structure imposed on the organization of objects in a filestore by the relations of parenthood and linkage.

5.8.10 filestore tree

The tree structure imposed on the organization of objects in a filestore by the relation of parenthood.

5.8.11 filestore root

The unique object in a filestore tree that is superior to all other objects in that tree.

5.8.12 pathname

A sequence of object names identifying a path from an assumed starting file-directory object through zero or more file-directories or references to file-directories, to a target object.

5.8.13 valid pathname

A pathname such that for any two adjacent names the object identified or referenced by the first relates to the object identified by the second by parenthood.

5.8.14 complete pathname

A pathname whose assumed starting file-directory is the filestore root.

5.8.15 incomplete pathname

A pathname whose assumed starting directory is not necessarily the filestore root. See clause 5.8.10.

5.8.16 primary pathname

The single complete pathname of an object where each adjacent pair of objects whose names appear in the pathname are related solely by parenthood.

5.8.17 attribute value assertion

A logical conjunction, disjunction, or negation of assertions about the values of individual attributes.

NOTE – The admissible forms of assertion about each attribute are given under its definition in ISO 8571-2.

Section one: FTAM general concepts

9.1 Control of actions

(amend 1st paragraph, page 7)

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

a) the object attributes (see section two); these indicate the actions appropriate to the object and its contents if any (in terms of the constraint set applied, in the case of files) and the local storage mechanisms (by means of the permitted actions object attribute) and express any access control constraints affecting object access;

aa) the path-access-control object attributes (see section two) of all file-directory and reference objects used to locate the target object; these may express access control constraints affecting the object access as well;

b) the current state of the filestore, particularly, in the case of file objects, the constraints implied by any concurrent access in progress to the same object;

c) in the case of file access, the values of the activity attributes established by parameters of the file service when the data transfer regime was being negotiated.

(amend 3rd paragraph, page 7)

The first diagram shows that the actions appropriate to an object are those which are allowed by the file structure as expressed by the structural constraint set (in the case of file objects), the permitted actions attribute, the access control attribute, and the path access control attributes of all file-directory and reference objects used to locate the target object.

(amend 4th paragraph, page 7)

When performing file access, 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.

9.2 Accounting

(amend 1st paragraph, page 8)

The FTAM service defines a basic mechanism for the carriage of accounting and charging information. Account names may be associated with an object to cover the costs arising from its storage, and accounts may be associated with regimes to cover the costs of actions performed on objects during the course of the regime.

(amend 2nd paragraph, page 8)

Corresponding charging parameters allow the costs incurred against these accounts to be reported when a regime is terminated. An account may be set up when an FTAM regime is initialized, but this may be overridden by nested regimes to allow actions within the nested regimes to be charged against a separate account if necessary.

9.3 Concurrency control

(amend 1st paragraph, page 8)

Concurrency control is only defined for access to file objects. The objective of the concurrency control mechanisms is to ensure that an initiator has a consistent view of the file by restricting shared access. These mechanisms are designed to provide a way for a user to perform a coordinated series of actions without interference from concurrent accesses.

9.4 Access control

(insert after 1st paragraph, page 9)

Access control information is derived from two separate sources. The object being accessed may have access control information associated with it in the form of the access control attribute. In addition, each file-directory or reference object used in the complete pathname identifying the accessed object may have additional access control information in the form of their path access control attributes. To perform an action on an object, an initiator must be granted access to the requested action in the access control list of the path access control attribute of each file-directory or reference object used in the identification of the target object. Additionally, access to the requested action must be granted in the access control list of the target object's access control attribute.

(amend 2nd paragraph, page 9)

In addition to the actions given in the path access control entries and access control entry, allowed concurrency control combinations can also be included when accessing file objects (see clause 9.3). If they are not included, the performance of concurrency control on file objects is determined locally by the filestore.

(amend 6th paragraph, page 9)

In establishing the FTAM regime and the file selection and open regimes, values are established for various activity attributes corresponding to the possible items in the list. In particular, the initiator asks to perform a certain set of actions by setting the current access request activity attribute when establishing a selection regime. When establishing the selection regime, before allowing the requested actions, the responding entity scans each access control list involved to determine if the activity attribute values match any of the entries. If, for each access control list involved, a match for the set of actions is found, and the associated tests are satisfied, the actions can be performed; if no match is found in any of the access control lists, the request is rejected.

(amend 7th paragraph, page 9)

Thus in summary, the path access control list is a permanent property of file-directories and references, and the access control list is a permanent property of all objects. They are stored for as long as the object exists. For an initiator to gain permission to perform a given set of actions, an access check is made against the path access control list of each file-directory or reference listed in the complete pathname used to identify a target object, and an access check is made against the access control list of the target object. These access checks are performed on each file considered for inclusion into the generalized selection group. Only files which allow the requested actions by the requesting initiator are included in the generalized selection group. In addition, these access checks are performed whenever a regime implying access to a single object is set up; the access granted remains valid for the duration of that regime. Access checks are also applied when interrogating the filestore for its contents.

Section two: Virtual Filestore - General Concepts

11.3 Form of the virtual filestore

(amend 1st paragraph, page 13)

The definition of the Virtual Filestore forms a schema for the description of an organization of filed information. In this definition, there are three kinds of objects - files, file-directories, and references (see figure 6a). Each type of object is distinguishable, and has specific characteristics. All objects share the following in common:

a) one or more pathnames that identify a path of access, and allow the object to be referenced without ambiguity;

b) other descriptive object attributes which express properties of the object such as accounting information, history, etc.;

c) object attributes expressing the actions capable of being performed on the object.

(amend 2nd paragraph, page 13)

These are all aspects of the object which can be observed by any authorized initiator. If two observers make the same inquiry about these aspects of a single object, they will obtain the same information about its properties, provided that no modification took place between the inquiries. These properties are called object attributes.

(amend 3rd paragraph, page 13)

There are also activity attributes describing the relation between the object and a particular initiator, concerned with things like authentication, data transfer options, accumulated cost, etc.; there is an independent set of values for these activity attributes for each activity in progress. This set is created after the FTAM regime involving the filestore is initialized, maintained while it persists, and destroyed at latest when it is finally released.

(append after 3rd paragraph, page 13)

A pathname is resolved to an object by a series of steps using components of the pathname to locate the intermediate objects along the path in turn (see Part 2, clause 5a.3.2). If the object located when a pathname is resolved is not of the type required for the operation to be performed then an error is reported.

11.3.1 File objects

Characteristics specific to a file object are

a) file attributes describing the logical structure and dimensions of the data stored in the file;

b) any file access data units forming the contents of this file.

(amend 4th paragraph, page 14)

Some file attributes place constraints on the structure of the file's content. This structure is preserved during the lifetime of the file. However, not all users accessing the file are concerned with its full generality. For instance, there may be a need to access a complex hierarchical file as if it were flat in order to construct summary reports, or it may not be necessary to access the smallest structural units of a file independently on all occasions. In addition to the file attributes used in file creation and file management for describing the permanent file access structure, there is a specified access context indicating, when a read data transfer is requested, the subset of the file structuring information and user data from the file access data unit to be transferred.

(append to subclause 11.3, page 14)

11.3.2 File-directory objects

A file-directory object maintains the relation of parenthood between itself and directly subordinate objects. The primary relationship between objects in the filestore is parenthood (the dual relationship of parenthood is childhood). The objects in the filestore form a tree under parenthood, where:

a) files map to nodes with zero or one data unit;

b) directories and references map to nodes without data units;

c) files may only occur as leaf nodes;

d) references may only occur as leaf nodes.

The sequence of parenthood relationships from the root of the filestore to an object is called the primary path to the object, and the sequence of object names on it is the primary pathname.

11.3.3 Reference objects

A reference object is linked to a single target object which is either a file or a file-directory. References may not be linked to other references.

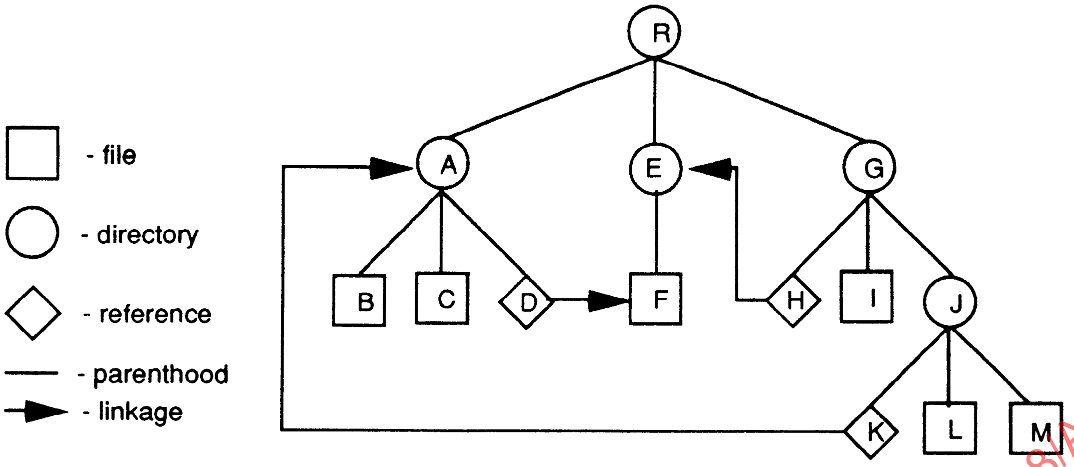

Figure 6a – An example tree structure of a VFS

For any reference, there exists a single linkage relationship from the reference object to another object in the filestore. Linkage is modeled in this standard by the value of the "referenced object" attribute of the reference, which records the primary pathname of the object to which the link is made.

11.4 Attribute dynamics

(amend 1st paragraph, page 14)

The attributes of an object reflect the state of the object as actually stored. Communication between the initiator and the responder builds up an incomplete shared picture of these object attributes, reflecting the knowledge about them which the initiator has gained from this particular communication. Thus in principle, for each object attribute, there exists a corresponding active attribute expressing the information about it which has been passed to the initiator via the parameters on the FTAM service primitives.

Section three: Overview of the file service and file protocol

15 File service

(amend 1st paragraph, page 17)

The file service and its supporting protocol are concerned with creating, in a series of stages, a working environment in which the initiator's desired activities can take place. The dialogue, in turn

a) allows the initiator and the responder (filestore) to establish information about each other, including their respective identities in terms of their application entity titles;

b) identifies one or a group of objects which are needed;

c) engages in the management of an object within a group of objects or management of the group of objects as a whole;

d) establishes the attributes describing a particular object and, for file objects, any bulk data transfer which is to take place in this activity;

e) locates the position in the file object's access structure of the FADUs to be accessed;

f) inserts, replaces, extends or erases one or more complete file object FADUs.

15.2 Filestore management phase

(amend 1st paragraph, page 17)

Operations in the filestore management phase allow the file service user to interrogate the collection of objects within the filestore, and determine their attributes. The filestore user may also override the default current name prefix activity attribute assigned by the responder during the FTAM regime initialization phase.

(append after subclause 15.2, page 17)

The file service user may also identify a group of file objects by an attribute value assertion list for subsequent individual selection, or for grouped file operations such as Group Copy or Group Move. The group is maintained as an activity attribute for the duration of the FTAM regime and may be re-established at any time during the filestore management phase. This group is defined to be empty after establishment of the FTAM regime. Note that only file objects are considered for inclusion into the group. Files may be removed from the group by the responder if they are deleted or their access control is changed to exclude this initiator. No notification of removal is made to the initiator.

Access control information may be provided by the initiator as required by the responder, on each individual selection or grouped file operation request. Access control is evaluated on each request and any object which does not grant this initiator the requested access is automatically excluded from the group, on group file operation requests, regardless of the value of its other attributes. The request could fail if either the object's access control is not met or any path access control in the path is not met. Note if an individual object within the group has a requirement for access control passwords then all passwords must be supplied when identifying the group on a grouped file operation. Therefore all objects within the group must have semantically equivalent password(s).

Operations in the filestore management phase allow the file service user to perform filing operations on the entire group of file objects with a single command. For instance, the group of files may be moved or copied to be in a different file-directory within the filestore.

(amend title of subclause 15.3, page 18)

15.3 Object selection phase

(replace the note at the end of subclause 15.3, page 18, with this paragraph)

Alternatively, file object selection may be done by selecting the next file object in the identified group. In this case, the responder identifies the file selected to the initiator via attributes identified by the initiator. No specific access control information is provided by the initiator during this phase. If the access control provided during the establishment of the group is insufficient for the requested access to a given file object, then the file object is excluded from the group, and not identified to the initiator.

(amend title of subclause 15.4, page 18)

15.4 Object management phase

(amend 1st paragraph, page 18)

Operations in the object management phase allow the

(amend Figure 10, page 17)

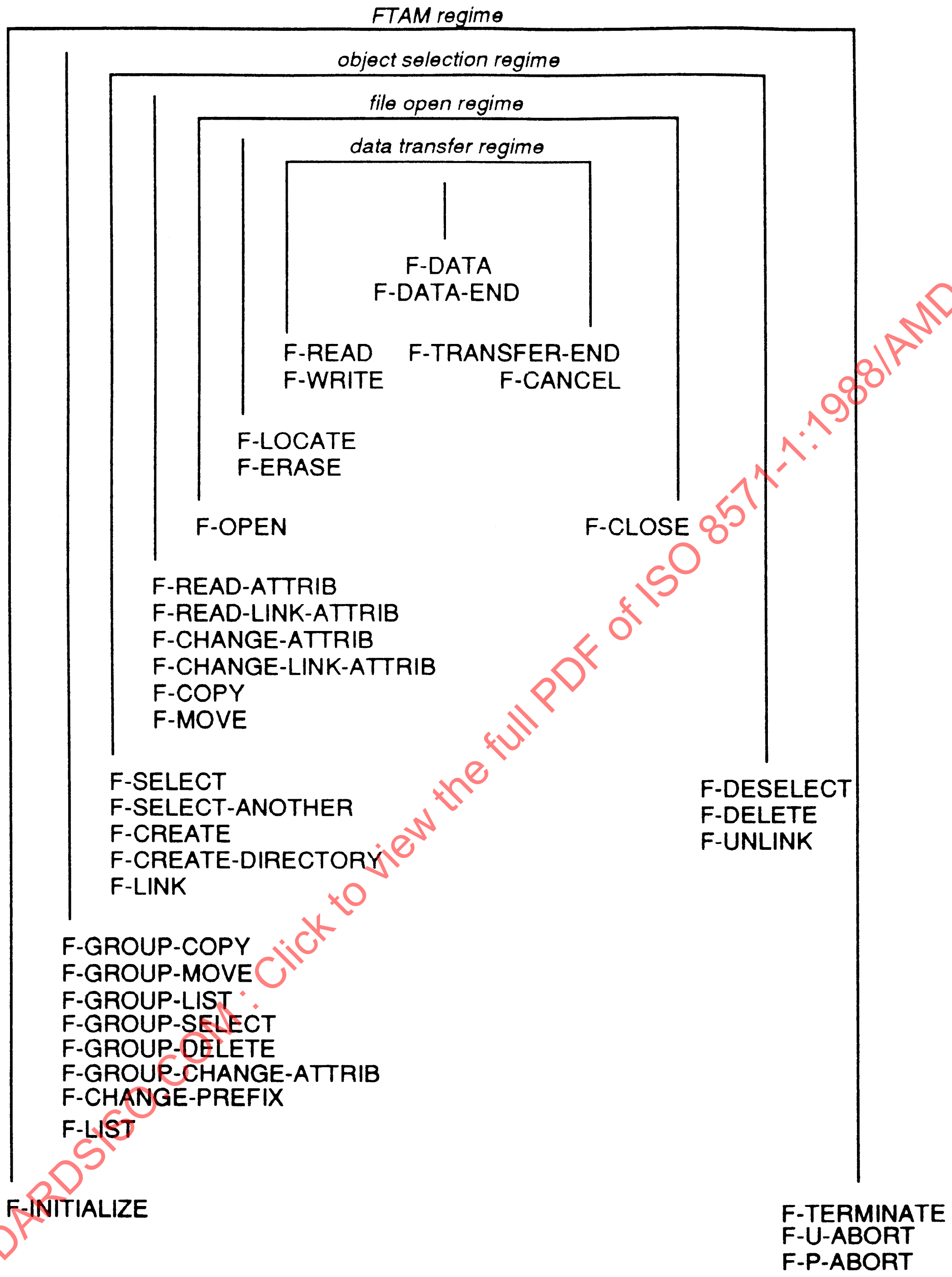

Figure 10 - File service regimes and related primitives

file service user to perform object management actions on the selected object. For instance, object attributes may be read or updated, and reference links identified or changed.

(amend title of subclause 15.8, page 18)

15.8 Object deselection phase

(amend 1st paragraph, page 18)

The deselection phase releases the object selection established by the selection phase. The deselection phase may include standard accounting sequences. It leaves in effect any authentication of account identification established before the object selection phase. The deselection phase may involve the deletion of the selected object.

Annex A
Examples of the use of FTAM

(This annex does not form part of the standard.)

(add to Annex A, page 24)

A.6 Filestore management activity

Suppose a filestore manager wished to preserve a copy of a specific set of files in their current condition. The copy is to remain on the remote system, but in a new file-directory. The steps which are taken are as follows (see figure 15):

a) the initialization of the association is performed, selecting the object access class, with the enhanced filestore management and group manipulation functional units.

b) the controlling system sends an instruction to modify the logical position in the filestore directory, with the limited filestore management functional unit.

c) the fileserver confirms the change in logical location.

d) the controlling system sends instructions to create the new directory, providing appropriate file-directory attributes.

e) the fileserver creates the directory and confirms the action.

f) the controlling system sends instructions to deselect the directory.

g) the fileserver confirms the action.

h) the controlling system sends instructions to form a group of file objects with attributes that identify the specific set of files to be preserved.

i) the fileserver searches the appropriate parts of the filestore structure and forms a private list of files with attributes meeting the identified criteria. It then confirms the action.

j) the controlling system sends instructions to copy that group of files to a new directory.

k) the fileserver duplicates the identified group of files into the new directory, and confirms the action.

l) the controlling system sends instructions to the filestore provider to enumerate a group of files within the filestore directory based on a set of selection criteria (for example, all objects of type directory). This operation requires the limited filestore management functional unit.

m) the fileserver confirms and returns the enumerated list of objects.

n) the controlling system selects an object, for example a file directory, within the filestore. When selecting a directory, or a reference, the limited filestore management functional unit is required.

o) the fileserver confirms the selection of the file directory.

p) the controlling system reads the attributes of the selected object, for example the children in the selected directory. When reading attributes of a directory, or a reference, the limited filestore management functional unit is required.

q) the fileserver returns the child object of the selected file directory.

r) the controlling system then deselects the directory.

s) the fileserver confirms the deselection.

t) the controlling system terminates the association and displays the results to the filestore manager.