INTERNATIONAL STANDARD

ISO 8571-3

First edition
1988-10-01

AMENDMENT 1
1992-12-15

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

Part 3 : File Service Definition

AMENDMENT 1 : Filestore Management

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

Partie 3: Définition du service de fichiers

AMENDEMENT 1 : Gestion du système de fichiers

Reference number
ISO 8571-3:1988/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 8571-3:1988 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

ISO 8571-3 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 3 :
File Service Definition

AMENDMENT 1 : Filestore Management

NOTE – This amendment has additional subclauses and tables to ISO 8571:1988 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.

Section one: General

6.2 File service levels

(amend 1st paragraph's item (a), page 3)

a) the external file service (EFS), in which the user states its FTAM quality of service requirements, but has no awareness of error recovery, delegating such considerations to the service provider. Transfer of file data and other operations on the filestore are modeled 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;

6.3 Regimes of the file service

(amend 1st paragraph, page 4)

Four types of file service regime are defined:

a) the FTAM regime, existing while the application association is used for the FTAM protocol, with which a group of file object complete pathnames is associated;

b) the object selection regime during which a particular object is associated with the FTAM regime;

c) the file open regime during which a particular set of processing mode, presentation contexts and concurrency controls is in operation;

d) the data transfer regime during which a particular bulk data transfer specification and direction of transfer are in force.

(amend 3rd paragraph, page 4)

The file service provides for:

e) a sequence of object selection regimes in an FTAM regime;

f) a sequence of file open regimes in an object selection regime;

g) a sequence of data transfer regimes within a file open regime; the data transfer regimes may each be for either read or write data transfer. Write data transfer permits the operations insert, replace or extend.

7.2 Filestore management

(replace clause, page 4)

Eight services are associated with filestore management:

a) the change current name prefix service (see clause 14a.1) is used by the initiator to control the mapping of incomplete pathnames to complete pathnames during the current association;

b) the list file-directory service (see clause 14a.2) is used by the initiator to interrogate for the attributes of objects which correspond to a given attribute value assertion list in or under a file-directory;

c) the generalized selection service (see clause 14b.1) is used by the initiator to identify a group of pathnames of files with attributes corresponding to a given attribute value assertion list;

d) the group deletion service (see clause 14b.2) is used by the initiator to remove the file objects identified by pathnames in the generalized selection group such that the objects cease to exist.

e) the group move service (see clause 14b.3) is used by the initiator to transfer the file objects identified by pathnames in the generalized selection group to a destination directory.

f) the group copy service (see clause 14b.4) is used by the initiator to duplicate the file objects identified by pathnames in the generalized selection group to a destination directory.

g) the group list service (see clause 14b.5) is used by the initiator to interrogate the pathnames of the file objects in the generalized selection group.

h) the group change attribute service (see clause 14b.6) is used by the initiator to modify the attributes of file objects, identified by complete pathnames within the generalized selection group activity attribute.

(amend title clause 7.3, page 4)

7.3 Object selection regime control

(amend 1st paragraph, page 4)

(amend Figure 2, page 4)

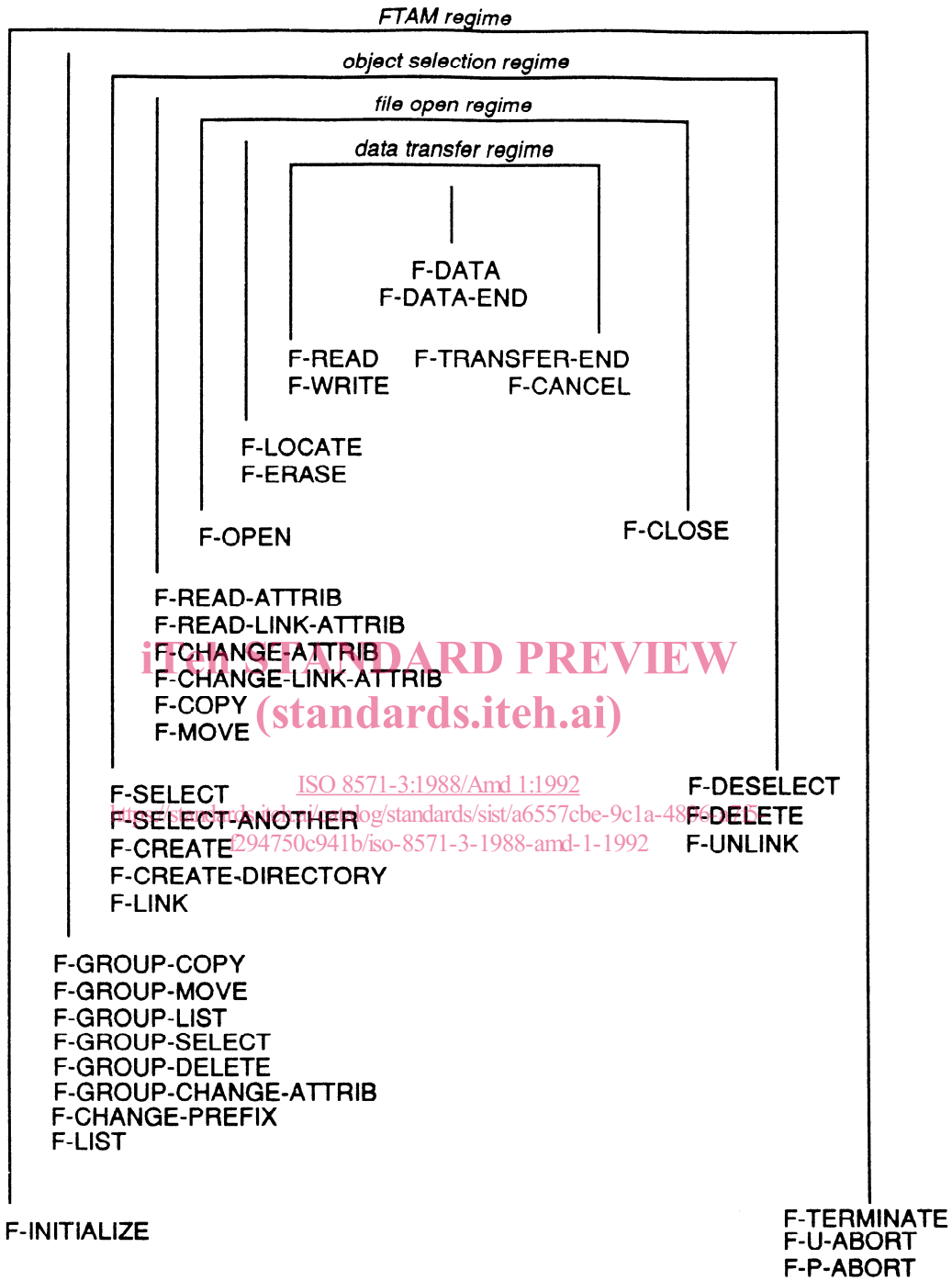


Figure 2 – File service regimes and related primitives

Eight services are associated with object selection regime control:

a) the object selection service (see clause 15.1) is used by the initiator to select a specific object by pathname and to bind the specified object to the FTAM regime;

aa) the select another service (see clause 15.1a) is used by the initiator to bind a previously unselected file from the generalized selection group to the FTAM regime;

b) the object deselection service (see clause 15.2) is used by the initiator to release the binding between the FTAM regime and the specified object;

c) the file creation service (see clause 15.3) is used by the initiator either

1) to create a specified file and to select the newly created file; or

2) depending on the override parameter of F–CREATE, to select an existing file;

and then to bind the specified file to the FTAM regime;

ca) the file-directory creation service (see clause 15.3a) is used by the initiator to create a specified file-directory object and bind the newly created file-directory to the FTAM regime;

cb) the reference creation service (see clause 15.3b) is used by the initiator to create a specified reference object and bind the existing object to which it is linked to the FTAM regime;

d) the object deletion service (see clause 15.4) is used by the initiator to release the binding between the FTAM regime and the specified object in such a way that the previously selected object ceased to exist;

e) the reference deletion service (see clause 15.5) is used to delete an existing reference, leaving the object to which it was linked intact in the filestore, and release the binding between the FTAM regime and the linked object.

(amend title to clause 7.4, page 5)

7.4 Object management

(amend 1st paragraph, page 5)

Six services are associated with object management:

a) the read attributes service (see clause 16.1) is used by the initiator to interrogate the object attributes of the selected object;

b) the change attributes service (see clause 16.2) is used by the initiator to modify the object attributes of the selected object;

c) the read reference attributes service (see clause 16.3) is used by the initiator to interrogate the object attributes of the reference identified by the current pathname activity attribute;

d) the change reference attributes service (see clause 16.4) is used by the initiator to modify the object attributes of the reference identified by the current pathname activity attribute;

e) the move object service (see clause 16.5) is used by the initiator to place the currently selected object into a specific file-directory by changing its primary pathname object attribute;

f) the copy object service (see clause 16.6) is used by the initiator to create a duplicate of the currently selected object in a specific file-directory.

NOTE – When accessing an object via a reference, the move object and copy object services operate only on the reference object, not the referent object.

7.6 Grouping control

(append after 1st paragraph, page 5)

The set of primitives contained within a group are constrained as follows:

a) FTAM regime control, filestore management and generalized filestore service primitives will never appear within a group;

b) recovery and restart service primitives will never appear within a group;

c) data transfer regime primitives, including F–DATA, F–DATA–END, F–LOCATE and F–ERASE, will never appear within a group;

d) if a regime is requested to be created and requested to be terminated within a single grouped sequence, then every regime requested to be created within the grouped sequence is also requested to be terminated within that grouped sequence;

e) primitives may be present within a group only if they are allowed within the negotiated service class, and the corresponding functional unit was

negotiated during the FTAM regime establishment;

f) any service from list 1 (below) must not appear in the same grouped sequence with a service from list 2:

1) F–CREATE,
F–CREATE–DIRECTORY

2) F–READ–LINK–ATTRIBUTES
F–CHANGE–LINK–ATTRIBUTES
F–UNLINK

Clause 12.2 and associated tables also place constraints on legal grouped sequences.

(append after clause 8.1.6, page 6)

8.1.6a Limited filestore management

The limited filestore management functional unit supports management of the current name prefix, and listing of objects within the filestore. In addition this functional unit supports the selection, deselection, and interrogation of directory and reference attributes.

8.1.6b Enhanced filestore management

The enhanced filestore management functional unit provides for the creation and deletion of file-directory and reference objects, and the modification of reference and directory object attributes.

8.1.6c Object manipulation functional unit

The object manipulation functional unit provides services to manage the position of objects within the filestore, and to duplicate objects within the filestore.

8.1.6d Group manipulation functional unit

The group manipulation functional unit provides for identifying a group of files within the filestore, interrogating the contents of that group, and for manipulation operations on that group of files.

8.2.1 File transfer class

(amend 1st paragraph, page 6)

ea) Optionally, the limited filestore management functional unit;

(amend 2nd paragraph, page 6)

In the file transfer service class, the use of the services is constrained so that there is a sequence of zero or more FTAM events on the application association. Each FTAM event requested by the initiator is a series of requests consisting of:

1) a single,optional, limited filestore management procedure to change current name prefix or list filestore;

2) a single grouped series of requests to establish a file open regime. (See clause 7.6 for valid grouped sequences);

3) a single bulk data transfer procedure, for either a read transfer or a write transfer. The processing mode parameter on the F–OPEN primitive is set to either a read or a valid write action, as defined in the constraint set, but not both.

4) a single grouped series of requests to release the file open and select regimes. (See clause 7.6 for valid grouped series.).

(delete note, page 6)

8.2.2 File access class

(amend 1st paragraph, pages 6 and 8)

The file access class consists of:

a) the kernel functional unit;

b) both of the read and write functional units;

c) the file access functional unit;

d) optionally, the grouping functional unit. If the grouping functional unit is successfully negotiated, its valid use in any instance by the initiator is optional but the acceptance by the responder is always mandatory.

e) optionally, the limited file management functional unit;

f) optionally, but only if the limited file management functional unit is present, the enhanced file management functional unit;

fa) optionally, the limited filestore management functional unit;

fb) optionally, but only if the limited filestore management functional unit is present, the enhanced filestore management functional unit;

fc) optionally, the object manipulation functional unit;

fd) optionally, but only if the object manipulation functional unit is present, the group manipulation functional unit;

g) optionally, the FADU locking functional unit;

h) optionally, in the internal file service, the recovery functional unit;

(append after entry U6, page 7)

Table 1 – Services and functional units of the External File Service

U6a Limited filestore management	change current name prefix list file-directory file-directory selection file-directory deselection reference selection reference deselection read file-directory attributes read reference attributes	O	O	O	O	O	14a.1 14a.2 15.1 15.2 15.1 15.2 16.1 16.3
U6b Enhanced filestore management	file-directory creation file-directory deletion reference creation reference deletion change file-directory attributes change reference attributes (requires 6a)		O	O	O	O	15.3a 15.4 15.3b 15.5 16.2 16.4
U6c Object manipulation	move object copy object		O	O	O	O	16.5 16.6
U6d Group manipulation	generalized selection generalized deletion group change attributes group move group copy group list select another		O	O	O	O	14b.1 14b.2 14b.7 14b.3 14b.5 14b.6 15.1a

i) optionally, in the internal file service, the restart data transfer functional unit.

(delete 2nd note, page 8)

8.2.3 File management class

(amend 1st paragraph, page 8)

The file management class consists of:

a) the kernel functional unit;

b) the limited file management functional unit;

c) optionally, the enhanced file management functional unit;

d) the grouping functional unit;

e) optionally, the limited filestore management functional unit;

f) optionally, but only if the limited filestore management functional unit is present, the enhanced filestore management functional unit;

g) optionally, the object manipulation functional unit;

h) optionally, but only if the object manipulation functional unit is present, the group manipulation functional unit.

(amend 2nd paragraph, page 8)

In the file management service class the use of the services is constrained so that there is a series of zero or more FTAM events on the application association. Each FTAM event requested by the initiator is a single grouped series of requests. (See clause 7.6 for valid sets of grouped primitives.) The file management class imposes the following further restrictions on grouped primitives:

a) The F–OPEN and F–CLOSE primitives are excluded from use;

b) if a grouped series of primitives establishes a select regime, it must also terminate the select regime.

8.2.4 File transfer and management class

(amend 1st paragraph, page 8)

The file transfer and management class consists of:

a) the kernel functional unit;

b) the grouping functional unit;

c) one or both of the read or write functional units;

d) the limited file management functional unit;

e) optionally, the enhanced file management functional unit;

ea) optionally, the limited filestore management functional unit;

eb) optionally, but only if the limited filestore management functional unit is present, the enhanced filestore management functional unit;

ec) optionally, the object manipulation functional unit;

ed) optionally, but only if the object manipulation functional unit is present, the group manipulation functional unit;

f) optionally, in the internal file service, the restart data transfer function unit;

g) optionally, in the internal file service, the recovery functional unit.

(amend 2nd paragraph, page 8)

In the file transfer and management service class, the use of the services is constrained so that there is a repeated series of FTAM events on the application association. Each FTAM event is either;

a transfer comprising:

1) a single, optional, limited filestore management procedure to change current name prefix or list filestore;

2) a single grouped series of requests requested by the initiator to establish a file open regime. (See clause 7.6 for valid grouped series).

3) a single bulk data transfer procedure, for either a read transfer or a write transfer. The processing mode parameter on the F–OPEN primitive is set to either a read or a valid write action, as defined in the constraint set, but not both.

4) a single grouped series of requests requested by the initiator to release the file open and select regimes. (See clause 7.6 for valid grouped series.)

or a single grouped series of requests requested by the initiator to effect management. (See clause 7.6 for valid grouped sequences.) The open regime will not be entered. If the select regime is established within a grouped series, then it will also be terminated within the grouped series of requests.

(delete the node, page 8)

(append after table 3, page 9)

limited filestore management enhanced filestore management	limited filestore management enhanced filestore management

Section two: Definition of file service primitives

12.2 Constraints on the issue of primitives

(amend 1st paragraph, pages 11 & 14; this includes the list of grouped sequences)

The primitives may be issued in any sequence consistent with the constraints given in tables 7 to 10. The sequences of primitives are defined under the individual services. Individual series of requests requested by the initiator may be interleaved to form grouped sequences. See clause 7.6 for valid groups.

(amend the notes, page 14)

NOTES

1) Not all sequences are allowed in the file transfer, file management, and file transfer and management classes (see clauses 8.2.1, 8.2.3, 8.2.4). In these classes, the threshold parameter is set so that the sequences either succeed or fail as a whole, i.e. set to the number of primitives between the begin group and end group primitives.

2) Other constraints will affect the ability of the file service user or file service provider to invoke the various procedures, such as flow control constraints on sending data or constraints on the ability of a file service user to accept spontaneous F-P-ABORT indications from the file service provider.

12.3.2 Conventions for tables 9 and 10

(amend 1st paragraph, page 14)

In tables 9 and 10, the entries indicate the functional units required for the succession to occur. The entries are:

Kernel	kernel functional unit
Grouping	grouping functional unit
Lmgt	limited management functional unit
Emgt	enhanced management functional unit
Recover	recover functional unit
Access	file access functional unit
G–Lmgt	grouping and limited file management functional units
G–Emgt	grouping and enhanced management functional units
Lfmg	limited filestore management functional unit
Efmg	enhanced filestore management functional unit
G–Lfmg	grouping and limited filestore management functional units
G–Efmg	grouping and enhanced filestore management functional units
Obmn	object manipulation functional unit
G–Obmn	grouping and object manipulation functional units
Grmn	group manipulation functional unit
G–Grmn	grouping and group manipulation functional units

(insert after 1st paragraph, page 14)

Tables 9 and 10 assume that completion of a previous file service event implies that all functional units required by the corresponding file service primitive were negotiated.

(insert after clause 12.3.2, page 14)

12.3.3 Notes from tables 9 and 10

[1] This transition is only allowed if no regimes other than the FTAM regime are currently established.

[2] This transition is only allowed if the select regime is the innermost regime established.

[3] This transition is only allowed if the open regime is the innermost regime established.

12.4 Confirmed services

(amend 2nd paragraph, page 20)

A request to establish a new regime (F–INITIALIZE, F–SELECT, F–SELECT–ANOTHER, F–CREATE, F–CREATE–DIRECTORY, F–LINK OR F–OPEN) may be rejected by use of a response with state result parameter indicating failure (see clause 13.1).

Table 6, page 11 modify as follows:

⋮	⋮	⋮	⋮
F-SELECT	Yes	Initiator	State result Action result Attributes Requested access Referent indicator Access passwords Concurrency control Shared ASE information Account Diagnostic

(append to table 6, pages 11 and 12)

Table 6 – File service primitives

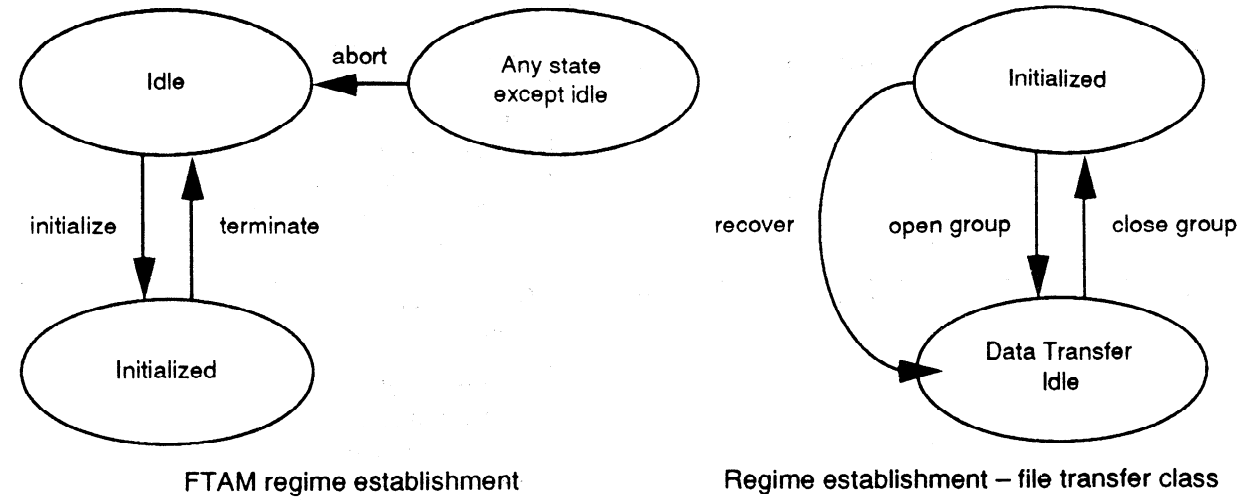
F-CHANGE-PREFIX	Yes	Initiator	Action result Reset Destination file-directory Access passwords Diagnostic
F-LIST	Yes	Initiator	Action result Attribute value assertions Scope Access passwords Attribute names Objects attributes list Diagnostic
F-GROUP-SELECT	Yes	Initiator	Action result State result Attribute value assertions Requested access Scope Access passwords Concurrency control Maximum set size Account Shared ASE information Diagnostic
F-GROUP-DELETE	Yes	Initiator	Action result Charging Request operation result Operation result Shared ASE information Diagnostic
F-GROUP-COPY	Yes	Initiator	Action result Destination file-directory Override Error action Create password Access passwords Attributes Request operation result Operation result Diagnostic
F-GROUP-MOVE	Yes	Initiator	Action result Destination file-directory Override Error action Create password Attributes Request operation result Operation result Diagnostic
F-GROUP-LIST	Yes	Initiator	Action result Attribute names Objects attributes list Diagnostic
F-GROUP-CHANGE-ATTRIB	Yes	Initiator	Action result Attributes Error action Shared ASE information Diagnostic
F-SELECT-ANOTHER	Yes	Initiator	State result Action result Last member indicator Shared ASE information referent indicator Diagnostic

(append to table 6, pages 11 and 12)

Table 6 (continued) – File service primitives

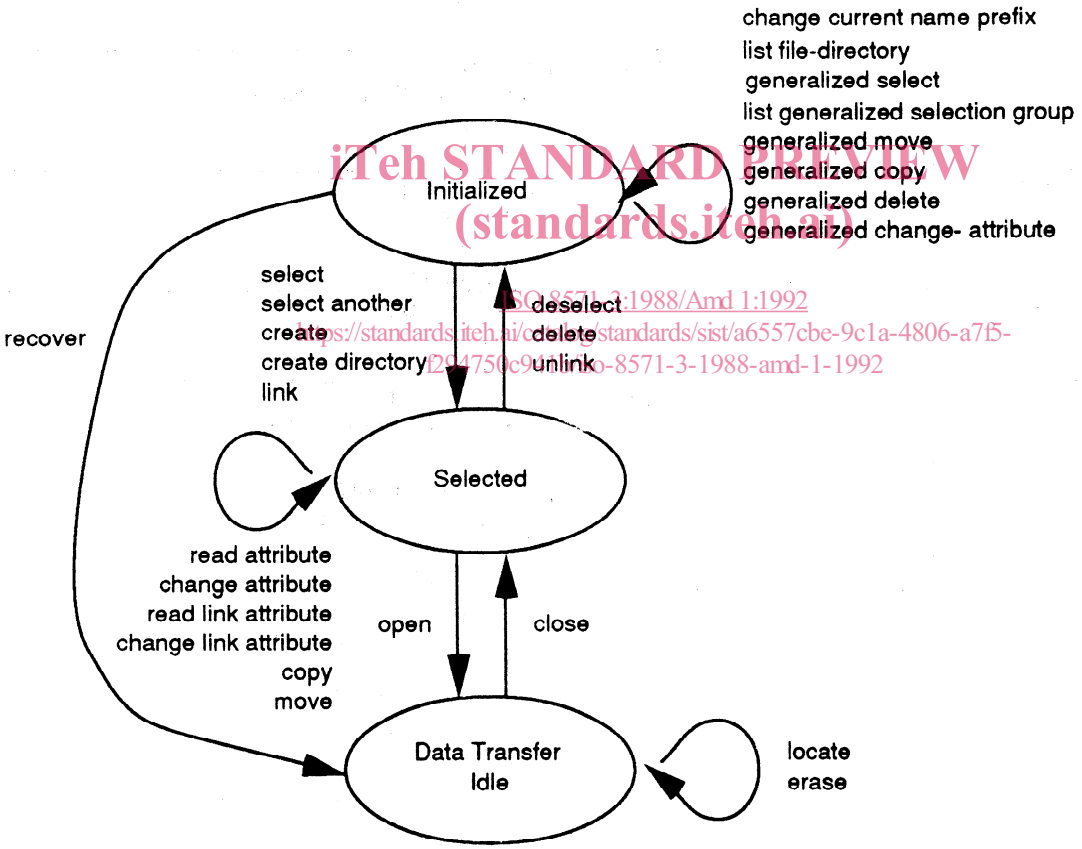
F-CREATE-DIRECTORY	Yes	Initiator	State result Action result Attributes Create password Requested access Shared ASE information Account Diagnostic
F-LINK	Yes	Initiator	State result Action result Attributes Target object Create password Requested access Access passwords Concurrency control Shared ASE information Account Diagnostic
F-UNLINK	Yes	Initiator	Action result Shared ASE information Charging Diagnostic
F-READ-LINK-ATTRIB	Yes	Initiator	Action result Attribute names Attributes Diagnostic
F-CHANGE-LINK-ATTRIB	Yes	Initiator	Action result Attributes Diagnostic
F-COPY	Yes	Initiator	Action result Destination file-directory Override Create password Access passwords Attributes Diagnostic
F-MOVE	Yes	Initiator	Action result Destination file-directory Override Create password Access passwords Attributes Diagnostic

(amend Figure 3, page 13)



FTAM regime establishment

Regime establishment – file transfer class



File regime establishment – file access class

Figure 3 – Simplified State Diagram for successful activity (see Annex E)

(amend Tables 9 and 10, pages 16 to 19)

Table 9 – Sequence of service primitives for file service regimes – initiator

Previous file service event	F-CHANGE-PREFIX.request	F-LIST .request	F-GROUP-SELECT .request	F-GROUP-DELETE .request	F-GROUP-COPY.request	F-GROUP-MOVE.request
FTAM regime established	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-CHANGE-PREFIX.request						
F-CHANGE-PREFIX.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-LIST.request						
F-LIST.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-SELECT.request						
F-GROUP-SELECT.confirm (+ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-SELECT.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-DELETE.request						
F-GROUP-DELETE.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-COPY.request						
F-GROUP-COPY.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-MOVE.request						
F-GROUP-MOVE.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-LIST.request						
F-GROUP-LIST.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-GROUP-CHANGE-ATTRIBUTE.request						
F-GROUP-CHANGE-ATTRIBUTE.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-SELECT.request						
F-SELECT.confirm (+ve)						
F-SELECT.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-SELECT-ANOTHER.request						
F-SELECT-ANOTHER.confirm (+ve)						
F-SELECT-ANOTHER.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-CREATE.request						
F-CREATE.confirm (+ve)						
F-CREATE.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-CREATE-DIRECTORY.request						
F-CREATE-DIRECTORY.confirm (+ve)						
F-CREATE-DIRECTORY.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-LINK.request						
F-LINK.confirm (+ve)						
F-LINK.confirm (-ve)	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-DESELECT.request						
F-DESELECT.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-DELETE.request						
F-DELETE.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-UNLINK.request						
F-UNLINK.confirm	Lfmg	Lfmg	Grmn	Grmn	Grmn	Grmn
F-READ-ATTRIB.request						
F-READ-ATTRIB.confirm						
F-CHANGE-ATTRIB.request						
F-CHANGE-ATTRIB.confirm						
F-READ-LINK-ATTRIB.request						