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)*

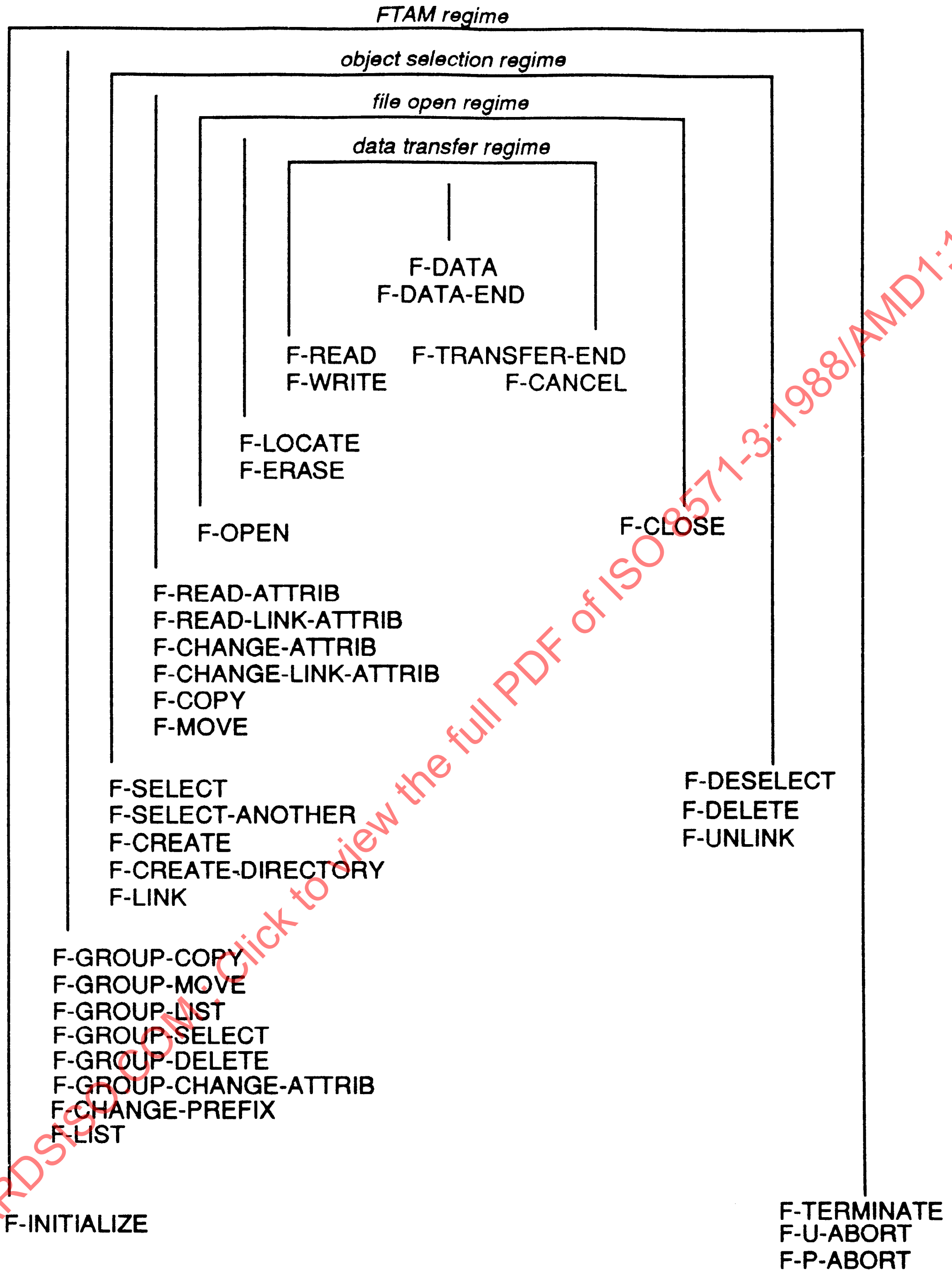


**Figure 2 – File service regimes and related primitives**

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<br>list file-directory<br>file-directory selection<br>file-directory deselection<br>reference selection<br>reference deselection<br>read file-directory attributes<br>read reference attributes | O | O | O | O | O | 14a.1<br>14a.2<br>15.1<br>15.2<br>15.1<br>15.2<br>16.1<br>16.3 |
| U6b Enhanced filestore management | file-directory creation<br>file-directory deletion<br>reference creation<br>reference deletion<br>change file-directory attributes<br>change reference attributes<br>(requires 6a) | | O | O | O | O | 15.3a<br>15.4<br>15.3b<br>15.5<br>16.2<br>16.4 |
| U6c Object manipulation | move object<br>copy object | | O | O | O | O | 16.5<br>16.6 |
| U6d Group manipulation | generalized selection<br>generalized deletion<br>group change attributes<br>group move<br>group copy<br>group list<br>select another | | O | O | O | O | 14b.1<br>14b.2<br>14b.7<br>14b.3<br>14b.5<br>14b.6<br>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<br>enhanced filestore management | limited filestore management<br>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<br>Action result<br>Attributes<br>Requested access<br>Referent indicator<br>Access passwords<br>Concurrency control<br>Shared ASE information<br>Account<br>Diagnostic |

*(append to table 6, pages 11 and 12)*

**Table 6 – File service primitives**

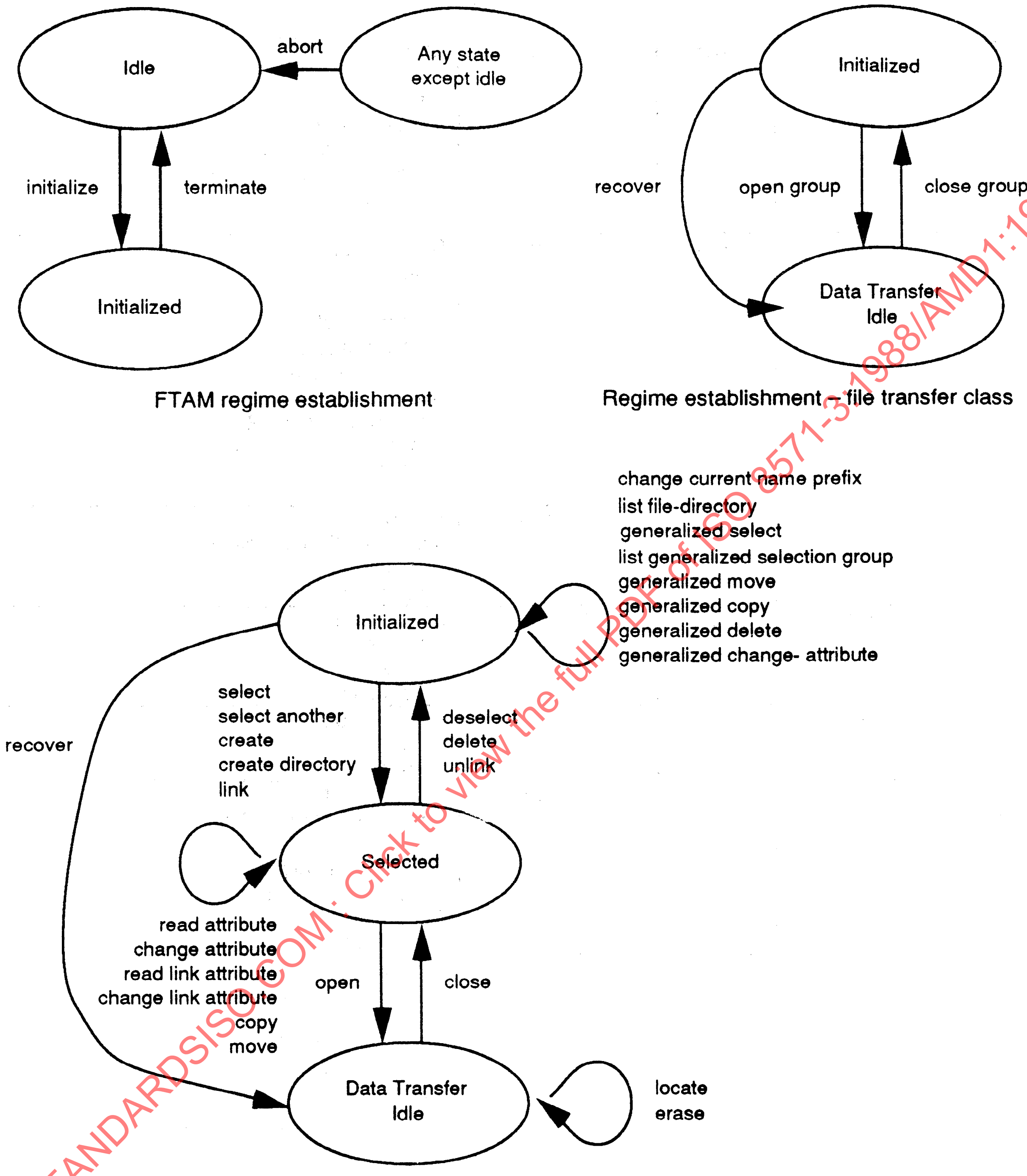
| | | | |
|---|---|---|---|
| F-CHANGE-PREFIX | Yes | Initiator | Action result<br>Reset<br>Destination file-directory<br>Access passwords<br>Diagnostic |
| F-LIST | Yes | Initiator | Action result<br>Attribute value assertions<br>Scope<br>Access passwords<br>Attribute names<br>Objects attributes list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

*(append to table 6, pages 11 and 12)*

**Table 6 (continued) – File service primitives**

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

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| 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 |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.confirm | | | | | | |
| F-CHANGE-LINK-ATTRIB.request | | | | | | |
| F-CHANGE-LINK-ATTRIB.confirm | | | | | | |
| F-COPY.request | | | | | | |
| F-COPY.confirm | | | | | | |
| F-MOVE.request | | | | | | |
| F-MOVE.confirm | | | | | | |
| F-OPEN.request | | | | | | |
| F-OPEN.confirm (+ve) | | | | | | |
| F-OPEN.confirm (-ve) | | | | | | |
| F-CLOSE.request | | | | | | |
| F-CLOSE.confirm | | | | | | |
| F-BEGIN-GROUP.request | | | | | | |
| F-BEGIN-GROUP.confirm | | | | | | |
| F-END-GROUP.request | | | | | | |
| F-END-GROUP.confirm | Lfmg | Lfmg | Grmn | Grmn | Grmn | Grmn |
| F-RECOVER.request | | | | | | |
| F-RECOVER.confirm (+ve) | | | | | | |
| F-RECOVER.confirm (-ve) | Lfmg | Lfmg | Grmn | Grmn | Grmn | Grmn |
| Bulk data transfer | | | | | | |
| F-LOCATE.request | | | | | | |
| F-LOCATE.confirm | | | | | | |
| F-ERASE.request | | | | | | |
| F-ERASE.confirm | | | | | | |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

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

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-GROUP-CHANGE-ATTRIB.request | F-GROUP-LIST.request | F-SELECT .request | F-SELECT-ANOTHER .request | F-CREATE .request | F-CREATE-DIRECTORY .request | F-LINK .request |
|---|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.confirm | | | | | | | |
| F-CHANGE-LINK-ATTRIB.request | | | | | | | |
| F-CHANGE-LINK-ATTRIB.confirm | | | | | | | |
| F-COPY.request | | | | | | | |
| F-COPY.confirm | | | | | | | |
| F-MOVE.request | | | | | | | |
| F-MOVE.confirm | | | | | | | |
| F-OPEN.request | | | | | | | |
| F-OPEN.confirm (+ve) | | | | | | | |
| F-OPEN.confirm (-ve) | | | | | | | |
| F-CLOSE.request | | | | | | | |
| F-CLOSE.confirm | | | | | | | |
| F-BEGIN-GROUP.request | | | Kernel or G-Lfmg | Grmn | Lmgt | Efmg | Efmg |
| F-BEGIN-GROUP.confirm | | | | | | | |
| F-END-GROUP.request | | | | | | | |
| F-END-GROUP.confirm | Grmn | Grmn | Kernel or G-Lfmg | Grmn | Lmgt | Efmg | Efmg |
| F-RECOVER.request | | | | | | | |
| F-RECOVER.confirm (+ve) | | | | | | | |
| F-RECOVER.confirm (-ve) | Grmn | Grmn | Kernel or Lfmg | Grmn | Lmgt | Efmg | Efmg |
| Bulk data transfer | | | | | | | |
| F-LOCATE.request | | | | | | | |
| F-LOCATE.confirm | | | | | | | |
| F-ERASE.request | | | | | | | |
| F-ERASE.confirm | | | | | | | |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-DESELECT .request | F-DELETE .request | F-UNLINK .request | F-READ-ATTRIB.request | F-CHANGE-ATTRIB .request | F-READ-LINK-ATTRIB .request |
|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | |
| F-CHANGE-PREFIX.request | | | | | | |
| F-CHANGE-PREFIX.confirm | | | | | | |
| F-LIST.request | | | | | | |
| F-LIST.confirm | | | | | | |
| F-GROUP-SELECT.request | | | | | | |
| F-GROUP-SELECT.confirm (+ve) | | | | | | |
| F-GROUP-SELECT.confirm (-ve) | | | | | | |
| F-GROUP-DELETE.request | | | | | | |
| F-GROUP-DELETE.confirm | | | | | | |
| F-GROUP-COPY.request | | | | | | |
| F-GROUP-COPY.confirm | | | | | | |
| F-GROUP-MOVE.request | | | | | | |
| F-GROUP-MOVE.confirm | | | | | | |
| F-GROUP-LIST.request | | | | | | |
| F-GROUP-LIST.confirm | | | | | | |
| F-GROUP-CHANGE-ATTRIB.request | | | | | | |
| F-GROUP-CHANGE-ATTRIB.confirm | | | | | | |
| F-SELECT.request | Grouping\|G-Lfmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Lmgt\|G-Lfmg | G-Lfmg |
| F-SELECT.confirm (+ve) | Kernel or Lfmg | Lmgt or Efmg | Efmg | Lmgt or Lfmg | Lmgt or Lfmg | Lfmg |
| F-SELECT.confirm (-ve) | | | | | | |
| F-SELECT-ANOTHER.request | Grouping\|G-Lfmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Lmgt\|G-Lfmg | G-Lfmg |
| F-SELECT-ANOTHER.confirm (+ve) | Kernel or Lfmg | Lmgt or Efmg | Efmg | Lmgt or Lfmg | Lmgt or Lfmg | Lfmg |
| F-SELECT-ANOTHER.confirm (-ve) | | | | | | |
| F-CREATE.request | Grouping | G-Lmgt | | G-Lmgt | G-Emgt | |
| F-CREATE.confirm (+ve) | Kernel | Lmgt | | Lmgt | Emgt | |
| F-CREATE.confirm (-ve) | | | | | | |
| F-CREATE-DIRECTORY.request | G-Efmg | G-Efmg | | G-Efmg | G-Efmg | |
| F-CREATE-DIRECTORY.confirm (+ve) | Kernel or Efmg | Efmg | | Efmg | Efmg | |
| F-CREATE-DIRECTORY.confirm (-ve) | | | | | | |
| F-LINK.request | G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Lfmg |
| F-LINK.confirm (+ve) | Efmg | Efmg | Efmg | Efmg | Efmg | Lfmg |
| F-LINK.confirm (-ve) | | | | | | |
| F-DESELECT.request | | | | | | |
| F-DESELECT.confirm | | | | | | |
| F-DELETE.request | | | | | | |
| F-DELETE.confirm | | | | | | |
| F-UNLINK.request | | | | | | |
| F-UNLINK.confirm | | | | | | |
| F-READ-ATTRIB.request | Grouping or Lfmg | G-Lmgt\|G-Lfmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-READ-ATTRIB.confirm | Kernel or Lfmg | Lmgt or Lfmg | Efmg | Lmgt or Lfmg | Emgt or Efmg | Lfmg |
| F-CHANGE-ATTRIB.request | Grouping or Efmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Efmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-CHANGE-ATTRIB.confirm | Kernel or Efmg | Lmgt or Efmg | Efmg | Lmgt or Efmg | Emgt or Efmg | Lfmg |
| F-READ-LINK-ATTRIB.request | Grouping | G-Efmg | G-Efmg | G-Lmgt\|G-Efmg | G-Lfmg\|G-Efmg | G-Lfmg |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-DESELECT .request | F-DELETE .request | F-UNLINK .request | F-READ-ATTRIB.request | F-CHANGE-ATTRIB .request | F-READ-LINK-ATTRIB .request |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.confirm | Lfmg | Efmg | Efmg | Lmgt\|Lfmg | Efmg | Lfmg |
| F-CHANGE-LINK-ATTRIB.request | G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Efmg |
| F-CHANGE-LINK-ATTRIB.confirm | Efmg | Efmg | Efmg | Efmg | Efmg | Efmg |
| F-COPY.request | G-Lfmg\|Grouping | G-Lmgt\|G-Efmg | G-Efmg | G-Lfmg\|G-Lmgt | G-Efmg\|G-Emgt | G-Lfmg |
| F-COPY.confirm | Lfmg\|Kernel | Lmgt\|Efmg | Efmg | Lmgt\|Lfmg | Efmg\|Emgt | Lfmg |
| F-MOVE.request | G-Lfmg\|Grouping | G-Lmgt\|G-Efmg | G-Efmg | G-Lfmg\|G-Lmgt | G-Efmg\|G-Emgt | G-Lfmg |
| F-MOVE.confirm | Lfmg\|Kernel | Lmgt\|Efmg | Efmg | Lmgt\|Lfmg | Efmg\|Emgt | Lfmg |
| F-OPEN.request | | | | | | |
| F-OPEN.confirm (+ve) | | | | | | |
| F-OPEN.confirm (-ve) | Kernel | Lmgt | Efmg | Lmgt | Emgt | Lfmg |
| F-CLOSE.request | Grouping | G-Lmgt | G-Efmg | G-Lmgt | G-Emgt | G-Lfmg |
| F-CLOSE.confirm | Kernel | Lmgt | Efmg | Lmgt | Emgt | Lfmg |
| F-BEGIN-GROUP.request | Lfmg\|Kernel [2] | Efmg\|Lmgt [2] | Efmg [2] | Lfmg\|Lmgt[2] | Efmg\|Emgt [2] | Lfmg [2] |
| F-BEGIN-GROUP.confirm | | | | | | |
| F-END-GROUP.request | | | | | | |
| F-END-GROUP.confirm | Lfmg\|Kernel [2] | Efmg\|Lmgt [2] | Efmg [2] | Lfmg\|Lmgt [2] | Efmg\|Emgt [2] | Lfmg [2] |
| F-RECOVER.request | | | | | | |
| F-RECOVER.confirm (+ve) | | | | | | |
| F-RECOVER.confirm (-ve) | | | | | | |
| Bulk data transfer | | | | | | |
| F-LOCATE.request | | | | | | |
| F-LOCATE.confirm | | | | | | |
| F-ERASE.request | | | | | | |
| F-ERASE.confirm | | | | | | |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-CHANGE-LINK-ATTRIB .request | F-COPY .request | F-MOVE .request | F-OPEN .request | F-CLOSE .request | F-BEGIN-GROUP.request |
|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | Grouping |
| F-CHANGE-PREFIX.request | | | | | | |
| F-CHANGE-PREFIX.confirm | | | | | | Grouping |
| F-LIST.request | | | | | | |
| F-LIST.confirm | | | | | | Grouping |
| F-GROUP-SELECT.request | | | | | | |
| F-GROUP-SELECT.confirm (+ve) | | | | | | |
| F-GROUP-SELECT.confirm (-ve) | | | | | | Grouping |
| F-GROUP-DELETE.request | | | | | | |
| F-GROUP-DELETE.confirm | | | | | | Grouping |
| F-GROUP-COPY.request | | | | | | |
| F-GROUP-COPY.confirm | | | | | | Grouping |
| F-GROUP-MOVE.request | | | | | | |
| F-GROUP-MOVE.confirm | | | | | | Grouping |
| F-GROUP-LIST.request | | | | | | |
| F-GROUP-LIST.confirm | | | | | | Grouping |
| F-GROUP-CHANGE-ATTRIB.request | | | | | | |
| F-GROUP-CHANGE-ATTRIB.confirm | | | | | | Grouping |
| F-SELECT.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-SELECT.confirm (+ve) | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-SELECT.confirm (-ve) | | | | | | Grouping |
| F-SELECT-ANOTHER.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-SELECT-ANOTHER.confirm (+ve) | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-SELECT-ANOTHER.confirm (-ve) | | | | | | Grouping |
| F-CREATE.request | | G-Obmn | G-Obmn | Grouping | | |
| F-CREATE.confirm (+ve) | | Obmn | Obmn | Kernel | | Grouping |
| F-CREATE.confirm (-ve) | | | | | | Grouping |
| F-CREATE-DIRECTORY.request | | G-Obmn | G-Obmn | | | |
| F-CREATE-DIRECTORY.confirm (+ve) | | Obmn | Obmn | | | Grouping |
| F-CREATE-DIRECTORY.confirm (-ve) | | | | | | Grouping |
| F-LINK.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-LINK.confirm (+ve) | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-LINK.confirm (-ve) | | | | | | Grouping |
| F-DESELECT.request | | | | | | |
| F-DESELECT.confirm | | | | | | Grouping |
| F-DELETE.request | | | | | | |
| F-DELETE.confirm | | | | | | Grouping |
| F-UNLINK.request | | | | | | |
| F-UNLINK.confirm | | | | | | Grouping |
| F-READ-ATTRIB.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-READ-ATTRIB.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-CHANGE-ATTRIB.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-CHANGE-ATTRIB.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-READ-LINK-ATTRIB.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-CHANGE-LINK-ATTRIB .request | F-COPY .request | F-MOVE .request | F-OPEN .request | F-CLOSE .request | F-BEGIN-GROUP.request |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-CHANGE-LINK-ATTRIB.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-CHANGE-LINK-ATTRIB.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-COPY.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-COPY.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-MOVE.request | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-MOVE.confirm | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-OPEN.request | | | | | Grouping | |
| F-OPEN.confirm (+ve) | | | | | Kernel | Grouping |
| F-OPEN.confirm (-ve) | Efmg | Obmn | Obmn | Kernel | | Grouping |
| F-CLOSE.request | G-Efmg | G-Obmn | G-Obmn | | | |
| F-CLOSE.confirm | Efmg | Obmn | Obmn | Kernel | Kernel | Grouping |
| F-BEGIN-GROUP.request | Efmg [2] | Obmn [2] | Obmn [2] | | Kernel [3] | |
| F-BEGIN-GROUP.confirm | | | | | | |
| F-END-GROUP.request | | | | | | |
| F-END-GROUP.confirm | Efmg [2] | Obmn [2] | Obmn [2] | Kernel [2] | Kernel [3] | Grouping |
| F-RECOVER.request | | | | | | |
| F-RECOVER.confirm (+ve) | | | | | Kernel | Grouping |
| F-RECOVER.confirm (-ve) | | | | | | Grouping |
| Bulk data transfer | | | | | Kernel | Grouping |
| F-LOCATE.request | | | | | | |
| F-LOCATE.confirm | | | | | Kernel | Grouping |
| F-ERASE.request | | | | | | |
| F-ERASE.confirm | | | | | Kernel | Grouping |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-END-GROUP .request | F-RECOVER .request | Bulk data transfer | F-LOCATE .request | F-ERASE .request | Terminate FTAM Regime |
|---|---|---|---|---|---|---|
| FTAM regime established | | Recover | | | | Kernel |
| F-CHANGE-PREFIX.request | | | | | | |
| F-CHANGE-PREFIX.confirm | | Recover | | | | Kernel |
| F-LIST.request | | | | | | |
| F-LIST.confirm | | Recover | | | | Kernel |
| F-GROUP-SELECT.request | | | | | | |
| F-GROUP-SELECT.confirm (+ve) | | | | | | |
| F-GROUP-SELECT.confirm (-ve) | | Recover | | | | Kernel |
| F-GROUP-DELETE.request | | | | | | |
| F-GROUP-DELETE.confirm | | Recover | | | | Kernel |
| F-GROUP-COPY.request | | | | | | |
| F-GROUP-COPY.confirm | | | | | | |
| F-GROUP-MOVE.request | | | | | | |
| F-GROUP-MOVE.confirm | | | | | | |
| F-GROUP-LIST.request | | | | | | |
| F-GROUP-LIST.confirm | | | | | | |
| F-GROUP-CHANGE-ATTRIB.request | | | | | | |
| F-GROUP-CHANGE-ATTRIB.confirm | | | | | | |
| F-SELECT.request | Grouping | | | | | |
| F-SELECT.confirm (+ve) | | | | | | |
| F-SELECT.confirm (-ve) | | Recover | | | | Kernel |
| F-SELECT-ANOTHER.request | G-Grmn | | | | | |
| F-SELECT-ANOTHER.confirm (+ve) | | | | | | |
| F-SELECT-ANOTHER.confirm (-ve) | | | | | | |
| F-CREATE.request | Grouping | | | | | |
| F-CREATE.confirm (+ve) | | | | | | |
| F-CREATE.confirm (-ve) | | Recover | | | | Kernel |
| F-CREATE-DIRECTORY.request | G-Efmg | | | | | |
| F-CREATE-DIRECTORY.confirm (+ve) | | | | | | |
| F-CREATE-DIRECTORY.confirm (-ve) | | Recover | | | | Kernel |
| F-LINK.request | G-Efmg | | | | | |
| F-LINK.confirm (+ve) | | | | | | |
| F-LINK.confirm (-ve) | | Recover | | | | Kernel |
| F-DESELECT.request | Grouping | | | | | |
| F-DESELECT.confirm | | Recover | | | | Kernel |
| F-DELETE.request | Grouping | | | | | |
| F-DELETE.confirm | | Recover | | | | Kernel |
| F-UNLINK.request | G-Efmg | | | | | |
| F-UNLINK.confirm | | Recover | | | | Kernel |
| F-READ-ATTRIB.request | Grouping | | | | | |
| F-READ-ATTRIB.confirm | | | | | | |
| F-CHANGE-ATTRIB.request | Grouping | | | | | |
| F-CHANGE-ATTRIB.confirm | | | | | | |
| F-READ-LINK-ATTRIB.request | G-Efmg | | | | | |

Table 9 – Sequence of service primitives for file service regimes – initiator (continued)

| Previous file service event | F-END-GROUP .request | F-RECOVER .request | Bulk data transfer | F-LOCATE .request | F-ERASE .request | Terminate FTAM Regime |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.confirm | | | | | | |
| F-CHANGE-LINK-ATTRIB.request | G-Efmg | | | | | |
| F-CHANGE-LINK-ATTRIB.confirm | | | | | | |
| F-COPY.request | G-Obmn | | | | | |
| F-COPY.confirm | | | | | | |
| F-MOVE.request | G-Obmn | | | | | |
| F-MOVE.confirm | | | | | | |
| F-OPEN.request | Grouping | | | | | |
| F-OPEN.confirm (+ve) | | | Kernel | Access | Access | |
| F-OPEN.confirm (-ve) | | | | | | |
| F-CLOSE.request | | | | | | |
| F-CLOSE.confirm | | | | | | |
| F-BEGIN-GROUP.request | | | | | | |
| F-BEGIN-GROUP.confirm | | | | | | |
| F-END-GROUP.request | | | | | | |
| F-END-GROUP.confirm | | Recover [1] | Kernel [3] | Access [3] | Access [3] | Kernel [1] |
| F-RECOVER.request | | | | | | |
| F-RECOVER.confirm (+ve) | | | Kernel | Access | Access | |
| F-RECOVER.confirm (-ve) | | Recover | | | | Kernel |
| Bulk data transfer | | | Kernel | Access | Access | |
| F-LOCATE.request | | | | | | |
| F-LOCATE.confirm | | | Kernel | Access | Access | |
| F-ERASE.request | | | | | | |
| F-ERASE.confirm | | | Kernel | Access | Access | |

Table 10 – Sequence of service primitives for file service regimes – responder

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

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-CHANGE-PREFIX .response | F-LIST .response | F-GROUP-SELECT .response | F-GROUP-DELETE .response | F-GROUP-COPY.response | F-GROUP-MOVE.response |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.response | | | | | | |
| F-CHANGE-LINK-ATTRIB.indication | | | | | | |
| F-CHANGE-LINK-ATTRIB.response | | | | | | |
| F-COPY.indication | | | | | | |
| F-COPY.response | | | | | | |
| F-MOVE.indication | | | | | | |
| F-MOVE.response | | | | | | |
| F-OPEN.indication | | | | | | |
| F-OPEN.response (+ve) | | | | | | |
| F-OPEN.response (-ve) | | | | | | |
| F-CLOSE.indication | | | | | | |
| F-CLOSE.response | | | | | | |
| F-BEGIN-GROUP.indication | | | | | | |
| F-BEGIN-GROUP.response | | | | | | |
| F-END-GROUP.indication | | | | | | |
| F-END-GROUP.response | | | | | | |
| F-RECOVER.indication | | | | | | |
| F-RECOVER.response (+ve) | | | | | | |
| F-RECOVER.response (-ve) | | | | | | |
| Bulk data transfer | | | | | | |
| F-LOCATE.indication | | | | | | |
| F-LOCATE.response | | | | | | |
| F-ERASE.indication | | | | | | |
| F-ERASE.response | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-GROUP-LIST.response | F-GROUP-CHANGE-ATTRIB.response | F-SELECT .response | F-SELECT-ANOTHER .response | F-CREATE .response | F-CREATE-DIRECTORY .response | F-LINK .response |
|---|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | | |
| F-CHANGE-PREFIX.indication | | | | | | | |
| F-CHANGE-PREFIX.response | | | | | | | |
| F-LIST.indication | | | | | | | |
| F-LIST.response | | | | | | | |
| F-GROUP-SELECT.indication | | | | | | | |
| F-GROUP-SELECT.response (+ve) | | | | | | | |
| F-GROUP-SELECT.response (-ve) | | | | | | | |
| F-GROUP-DELETE.indication | | | | | | | |
| F-GROUP-DELETE.response | | | | | | | |
| F-GROUP-COPY.indication | | | | | | | |
| F-GROUP-COPY.response | | | | | | | |
| F-GROUP-MOVE.indication | | | | | | | |
| F-GROUP-MOVE.response | | | | | | | |
| F-GROUP-LIST.indication | Grmn | | | | | | |
| F-GROUP-LIST.response | | | | | | | |
| F-GROUP-CHANGE-ATTRIB.indication | | Grmn | | | | | |
| F-GROUP-CHANGE-ATTRIB.response | | | | | | | |
| F-SELECT.indication | | | Kernel\|\|Lfmg | | | | |
| F-SELECT.response (+ve) | | | | | | | |
| F-SELECT.response (-ve) | | | | | | | |
| F-SELECT-ANOTHER.indication | | | | Grmn | | | |
| F-SELECT-ANOTHER.response (+ve) | | | | | | | |
| F-SELECT-ANOTHER.response (-ve) | | | | | | | |
| F-CREATE.indication | | | | | Lmgt | | |
| F-CREATE.response (+ve) | | | | | | | |
| F-CREATE.response (-ve) | | | | | | | |
| F-CREATE-DIRECTORY.indication | | | | | | Efmg | |
| F-CREATE-DIRECTORY.response (+ve) | | | | | | | |
| F-CREATE-DIRECTORY.response (-ve) | | | | | | | |
| F-LINK.indication | | | | | | | Efmg |
| F-LINK.response (+ve) | | | | | | | |
| F-LINK.response (-ve) | | | | | | | |
| F-DESELECT.indication | | | | | | | |
| F-DESELECT.response | | | | | | | |
| F-DELETE.indication | | | | | | | |
| F-DELETE.response | | | | | | | |
| F-UNLINK.indication | | | | | | | |
| F-UNLINK.response | | | | | | | |
| F-READ-ATTRIB.indication | | | | | | | |
| F-READ-ATTRIB.response | | | | | | | |
| F-CHANGE-ATTRIB.indication | | | | | | | |
| F-CHANGE-ATTRIB.response | | | | | | | |
| F-READ-LINK-ATTRIB.indication | | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-GROUP-CHANGE-ATTRIB.response | F-GROUP-LIST.response | F-SELECT .response | F-SELECT-ANOTHER .response | F-CREATE .response | F-CREATE-DIRECTORY .response | F-LINK .response |
|---|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.response | | | | | | | |
| F-CHANGE-LINK-ATTRIB.indication | | | | | | | |
| F-CHANGE-LINK-ATTRIB.response | | | | | | | |
| F-COPY.indication | | | | | | | |
| F-COPY.response | | | | | | | |
| F-MOVE.indication | | | | | | | |
| F-MOVE.response | | | | | | | |
| F-OPEN.indication | | | | | | | |
| F-OPEN.response (+ve) | | | | | | | |
| F-OPEN.response (-ve) | | | | | | | |
| F-CLOSE.indication | | | | | | | |
| F-CLOSE.response | | | | | | | |
| F-BEGIN-GROUP.indication | | | | | | | |
| F-BEGIN-GROUP.response | | | Grouping | G-Grmn | G-Lmgt | G-Efmg | G-Efmg |
| F-END-GROUP.indication | | | | | | | |
| F-END-GROUP.response | | | | | | | |
| F-RECOVER.indication | | | | | | | |
| F-RECOVER.response (+ve) | | | | | | | |
| F-RECOVER.response (-ve) | | | | | | | |
| Bulk data transfer | | | | | | | |
| F-LOCATE.indication | | | | | | | |
| F-LOCATE.response | | | | | | | |
| F-ERASE.indication | | | | | | | |
| F-ERASE.response | | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-DESELECT .response | F-DELETE .response | F-UNLINK .response | F-READ-ATTRIB .response | F-CHANGE-ATTRIB .response | F-READ-LINK-ATTRIB .response |
|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | |
| F-CHANGE-PREFIX.indication | | | | | | |
| F-CHANGE-PREFIX.response | | | | | | |
| F-LIST.indication | | | | | | |
| F-LIST.response | | | | | | |
| F-GROUP-SELECT.indication | | | | | | |
| F-GROUP-SELECT.response (+ve) | | | | | | |
| F-GROUP-SELECT.response (-ve) | | | | | | |
| F-GROUP-DELETE.indication | | | | | | |
| F-GROUP-DELETE.response | | | | | | |
| F-GROUP-COPY.indication | | | | | | |
| F-GROUP-COPY.response | | | | | | |
| F-GROUP-MOVE.indication | | | | | | |
| F-GROUP-MOVE.response | | | | | | |
| F-GROUP-LIST.indication | | | | | | |
| F-GROUP-LIST.response | | | | | | |
| F-GROUP-CHANGE-ATTRIB.indication | | | | | | |
| F-GROUP-CHANGE-ATTRIB.response | | | | | | |
| F-SELECT.indication | | | | | | |
| F-SELECT.response (+ve) | Grouping\|G-Lfmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-SELECT.response (-ve) | | | | | | |
| F-SELECT-ANOTHER.indication | | | | | | |
| F-SELECT-ANOTHER.response (+ve) | Grouping\|G-Lfmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-SELECT-ANOTHER.response (-ve) | | | | | | |
| F-CREATE.indication | | | | | | |
| F-CREATE.response (+ve) | Grouping | G-Lmgt | | G-Lmgt | G-Emgt | |
| F-CREATE.response (-ve) | | | | | | |
| F-CREATE-DIRECTORY.indication | | | | | | |
| F-CREATE-DIRECTORY.response(+ve) | Grouping\|G-Efmg | G-Efmg | | G-Efmg | G-Efmg | |
| F-CREATE-DIRECTORY.response (-ve) | | | | | | |
| F-LINK.indication | | | | | | |
| F-LINK.response (+ve) | Grouping\|G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Efmg |
| F-LINK.response (-ve) | | | | | | |
| F-DESELECT.indication | Kernel\|Lfmg | | | | | |
| F-DESELECT.response | | | | | | |
| F-DELETE.indication | | Lmgt\|Efmg | | | | |
| F-DELETE.response | | | | | | |
| F-UNLINK.indication | | | Efmg | | | |
| F-UNLINK.response | | | | | | |
| F-READ-ATTRIB.indication | | | | Lmgt\|Lfmg | | |
| F-READ-ATTRIB.response | Grouping\|G-Lfmg | G-Lmgt\|G-Efmg | G-Efmg | | G-Emgt\|G-Efmg | G-Lfmg |
| F-CHANGE-ATTRIB.indication | | | | | Emgt\|Efmg | |
| F-CHANGE-ATTRIB.response | Grouping\|G-Efmg | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Efmg | G-Emgt\|G-Efmg | G-Efmg |
| F-READ-LINK-ATTRIB.indication | | | | | | Lfmg |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-DESELECT .response | F-DELETE .response | F-UNLINK .response | F-READ-ATTRIB .response | F-CHANGE-ATTRIB .response | F-READ-LINK-ATTRIB .response |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.response | G-Lfmg\|G-Efmg | G-Lmgt\|G-Lfmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-CHANGE-LINK-ATTRIB.indication | | | | | | |
| F-CHANGE-LINK-ATTRIB.response | G-Lfmg\|G-Efmg | G-Efmg | G-Efmg | G-Efmg | G-Emgt | G-Efmg |
| F-COPY.indication | | | | | | |
| F-COPY.response | G-Lfmg\|Grouping | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-MOVE.indication | | | | | | |
| F-MOVE.response | G-Lfmg\|Grouping | G-Lmgt\|G-Efmg | G-Efmg | G-Lmgt\|G-Lfmg | G-Emgt\|G-Efmg | G-Lfmg |
| F-OPEN.indication | | | | | | |
| F-OPEN.response (+ve) | | | | | | |
| F-OPEN.response (-ve) | | | | | | |
| F-CLOSE.indication | | | | | | |
| F-CLOSE.response | Grouping | G-Lmgt | G-Efmg | G-Lmgt | G-Emgt | G-Lfmg |
| F-BEGIN-GROUP.indication | | | | | | |
| F-BEGIN-GROUP.response | G-Lfmg\|Group[2] | G-Lmgt[2]\|G-Efmg | G-Efmg [2] | G-Lmgt[6]\|G-lfmg | G-emgt[6]\|G-efmg | G-Lfmg [2] |
| F-END-GROUP.indication | | | | | | |
| F-END-GROUP.response | | | | | | |
| F-RECOVER.indication | | | | | | |
| F-RECOVER.response (+ve) | | | | | | |
| F-RECOVER.response (-ve) | | | | | | |
| Bulk data transfer | | | | | | |
| F-LOCATE.indication | | | | | | |
| F-LOCATE.response | | | | | | |
| F-ERASE.indication | | | | | | |
| F-ERASE.response | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-CHANGE-LINK-ATTRIB .response | F-COPY .response | F-MOVE .response | F-OPEN .response | F-CLOSE .response | F-BEGIN-GROUP .response |
|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | |
| F-CHANGE-PREFIX.indication | | | | | | |
| F-CHANGE-PREFIX.response | | | | | | |
| F-LIST.indication | | | | | | |
| F-LIST.response | | | | | | |
| F-GROUP-SELECT.indication | | | | | | |
| F-GROUP-SELECT.response (+ve) | | | | | | |
| F-GROUP-SELECT.response (-ve) | | | | | | |
| F-GROUP-DELETE.indication | | | | | | |
| F-GROUP-DELETE.response | | | | | | |
| F-GROUP-COPY.indication | | | | | | |
| F-GROUP-COPY.response | | | | | | |
| F-GROUP-MOVE.indication | | | | | | |
| F-GROUP-MOVE.response | | | | | | |
| F-GROUP-LIST.indication | | | | | | |
| F-GROUP-LIST.response | | | | | | |
| F-GROUP-CHANGE-ATTRIB.indication | | | | | | |
| F-GROUP-CHANGE-ATTRIB.response | | | | | | |
| F-SELECT.indication | | | | | | |
| F-SELECT.response (+ve) | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-SELECT.response (-ve) | | | | | | |
| F-SELECT-ANOTHER.indication | | | | | | |
| F-SELECT-ANOTHER.response (+ve) | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-SELECT-ANOTHER.response (-ve) | | | | | | |
| F-CREATE.indication | | | | | | |
| F-CREATE.response (+ve) | | G-Obmn | G-Obmn | Grouping | | |
| F-CREATE.response (-ve) | | | | | | |
| F-CREATE-DIRECTORY.indication | | | | | | |
| F-CREATE-DIRECTORY.response (+ve) | | G-Obmn | G-Obmn | | | |
| F-CREATE-DIRECTORY.response (-ve) | | | | | | |
| F-LINK.indication | | | | | | |
| F-LINK.response (+ve) | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-LINK.response (-ve) | | | | | | |
| F-DESELECT.indication | | | | | | |
| F-DESELECT.response | | | | | | |
| F-DELETE.indication | | | | | | |
| F-DELETE.response | | | | | | |
| F-UNLINK.indication | | | | | | |
| F-UNLINK.response | | | | | | |
| F-READ-ATTRIB.indication | | | | | | |
| F-READ-ATTRIB.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-CHANGE-ATTRIB.indication | | | | | | |
| F-CHANGE-ATTRIB.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-READ-LINK-ATTRIB.indication | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-CHANGE-LINK-ATTRIB .response | F-COPY .response | F-MOVE .response | F-OPEN .response | F-CLOSE .response | F-BEGIN-GROUP .response |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-CHANGE-LINK-ATTRIB.indication | Efmg | | | | | |
| F-CHANGE-LINK-ATTRIB.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-COPY.indication | | Obmn | | | | |
| F-COPY.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-MOVE.indication | | | Obmn | | | |
| F-MOVE.response | G-Efmg | G-Obmn | G-Obmn | Grouping | | |
| F-OPEN.indication | | | | Kernel | | |
| F-OPEN.response (+ve) | | | | | Grouping | |
| F-OPEN.response (-ve) | | | | | | |
| F-CLOSE.indication | | | | | Kernel | |
| F-CLOSE.response | G-Efmg | G-Obmn | G-Obmn | | | |
| F-BEGIN-GROUP.indication | | | | | | |
| F-BEGIN-GROUP.response | G-Efmg [2] | G-Obmn [2] | G-Obmn [2] | | Grouping [3] | |
| F-END-GROUP.indication | | | | | | Grouping |
| F-END-GROUP.response | | | | | | |
| F-RECOVER.indication | | | | | | |
| F-RECOVER.response (+ve) | | | | | | |
| F-RECOVER.response (-ve) | | | | | | |
| Bulk data transfer | | | | | | |
| F-LOCATE.indication | | | | | | |
| F-LOCATE.response | | | | | | |
| F-ERASE.indication | | | | | | |
| F-ERASE.response | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (continued)

| Previous file service event | F-END-GROUP .response | F-RECOVER .response | Bulk data transfer | F-LOCATE .response | F-ERASE .response | Terminate FTAM Regime |
|---|---|---|---|---|---|---|
| FTAM regime established | | | | | | |
| F-CHANGE-PREFIX.indication | | | | | | |
| F-CHANGE-PREFIX.response | | | | | | |
| F-LIST.indication | | | | | | |
| F-LIST.response | | | | | | |
| F-GROUP-SELECT.indication | | | | | | |
| F-GROUP-SELECT.response (+ve) | | | | | | |
| F-GROUP-SELECT.response (-ve) | | | | | | |
| F-GROUP-DELETE.indication | | | | | | |
| F-GROUP-DELETE.response | | | | | | |
| F-GROUP-COPY.indication | | | | | | |
| F-GROUP-COPY.response | | | | | | |
| F-GROUP-MOVE.indication | | | | | | |
| F-GROUP-MOVE.response | | | | | | |
| F-GROUP-LIST.indication | | | | | | |
| F-GROUP-LIST.response | | | | | | |
| F-GROUP-CHANGE-ATTRIB.indication | | | | | | |
| F-GROUP-CHANGE-ATTRIB.response | | | | | | |
| F-SELECT.indication | | | | | | |
| F-SELECT.response (+ve) | Grouping | | | | | |
| F-SELECT.response (-ve) | Grouping | | | | | |
| F-SELECT-ANOTHER.indication | | | | | | |
| F-SELECT-ANOTHER.response (+ve) | Grouping | | | | | |
| F-SELECT-ANOTHER.response (-ve) | Grouping | | | | | |
| F-CREATE.indication | | | | | | |
| F-CREATE.response (+ve) | Grouping | | | | | |
| F-CREATE.response (-ve) | Grouping | | | | | |
| F-CREATE-DIRECTORY.indication | | | | | | |
| F-CREATE-DIRECTORY.response (+ve) | Grouping | | | | | |
| F-CREATE-DIRECTORY.response (-ve) | Grouping | | | | | |
| F-LINK.indication | | | | | | |
| F-LINK.response (+ve) | Grouping | | | | | |
| F-LINK.response (-ve) | Grouping | | | | | |
| F-DESELECT.indication | | | | | | |
| F-DESELECT.response | Grouping | | | | | |
| F-DELETE.indication | | | | | | |
| F-DELETE.response | Grouping | | | | | |
| F-UNLINK.indication | | | | | | |
| F-UNLINK.response | Grouping | | | | | |
| F-READ-ATTRIB.indication | | | | | | |
| F-READ-ATTRIB.response | Grouping | | | | | |
| F-CHANGE-ATTRIB.indication | | | | | | |
| F-CHANGE-ATTRIB.response | Grouping | | | | | |
| F-READ-LINK-ATTRIB.indication | | | | | | |

Table 10 – Sequence of service primitives for file service regimes – responder (concluded)

| Previous file service event | F-END-GROUP .response | F-RECOVER .response | Bulk data transfer | F-LOCATE .response | F-ERASE .response | Terminate FTAM Regime |
|---|---|---|---|---|---|---|
| F-READ-LINK-ATTRIB.response | Grouping | | | | | |
| F-CHANGE-LINK-ATTRIB.indication | | | | | | |
| F-CHANGE-LINK-ATTRIB.response | Grouping | | | | | |
| F-COPY.indication | | | | | | |
| F-COPY.response | Grouping | | | | | |
| F-MOVE.indication | | | | | | |
| F-MOVE.response | Grouping | | | | | |
| F-OPEN.indication | | | | | | |
| F-OPEN.response (+ve) | Grouping | | | | | |
| F-OPEN.response (-ve) | Grouping | | | | | |
| F-CLOSE.indication | | | | | | |
| F-CLOSE.response | | | | | | |
| F-BEGIN-GROUP.indication | | | | | | |
| F-BEGIN-GROUP.response | | | | | | |
| F-END-GROUP.indication | | | | | | |
| F-END-GROUP.response | | | | | | |
| F-RECOVER.indication | | Recover | | | | |
| F-RECOVER.response (+ve) | | | | | | |
| F-RECOVER.response (-ve) | | | | | | |
| Bulk data transfer | | | | | | |
| F-LOCATE.indication | | | | Access | | |
| F-LOCATE.response | | | | | | |
| F-ERASE.indication | | | | | Access | |
| F-ERASE.response | | | | | | |

# 13 Common file service parameters

## 13.5 Attributes

*(amend 1st paragraph, page 20)*

The attributes parameter conveys a list of object attribute names and object attribute values associated with the object. It also may convey a pathname used to access the object. F–INITIALIZE negotiates the attribute groups which are available for the duration of the FTAM regime. Subsequent primitives only use those attributes which were negotiated.

NOTE – The pathname element (not to be confused with the primary pathname element) of the attributes parameter is not an object attribute. An object may be accessible by several pathnames. The pathname element conveys the actual pathname identifying the object, and is used in conjunction with the current pathname activity attribute. It is used to change the object name attribute, and may effect the primary pathname attribute.

The object name attribute may be derived from the last graphicstring element in the pathname vector. If the object is being accessed by a reference, the object name will be that of the reference object, not the target object.

*(amend 2nd paragraph, page 20)*

On primitives other than F–SELECT issued by the initiator, the parameter conveys new values to be assigned to the object attributes. On primitives issued by the responder, the parameter reports the current values of the object attributes, or indications that there is no value available.

*(amend 3rd paragraph, page 20)*

The pathname element does not always correspond to the primary pathname object attribute. For existing objects, this element identifies a given object by any valid pathname, and sets the current pathname activity identifier. The pathname element on F–SELECT identifies the object to be selected, or on F–CREATE the pathname of the object to be created and selected. In the request and indication primitives it indicates the object required, and in the response and confirm primitives it indicates the object actually selected.

NOTES

1) If, for example, the object pathname requested gave a generic name or a generation name, the name selected might differ from that requested.

2) The pathname specified on create actions will not necessarily be the primary pathname of the object being created.

*(amend 4th paragraph, page 20)*

The object attributes, the range of values they may take and the actions to be taken are defined in the virtual filestore definition (ISO 8571–2). Annex B specifies which object attributes may be manipulated by each primitive.

## 13.6 Requested access

*(amend 1st paragraph, page 20)*

The requested access parameter indicates how objects affected by the operation will be accessed. The value gives, as a vector, the actions to be performed on objects affected by the service. The elements of the vector correspond to the read, insert, replace, extend, erase, read attribute, change attribute, and delete object actions, and each element indicates whether the action is required or not.

*(amend 2nd paragraph's note, page 20)*

NOTE – When objects are targeted for selection, either individually by a select regime establishment or recovery actions, or by a generalized selection regime, the requested access parameter is used to state the maximum facilities the current user will require during the whole select regime. These access requirements may be password protected by the access control object attribute which is matched to the access passwords parameter (see clause 13.7).

*(amend 3rd paragraph, page 21)*

The requested access may be further qualified by any concurrency control requested, which specifies how multiple users are to be restricted when requiring access to the same file object. The requested access parameter includes only those actions to be performed by the user; any action not included is not available later in the select regime.

*(amend 4th paragraph, page 21)*

The relationship between attributes and actions is detailed in ISO 8571–2. This parameter is used to set the current access request activity attribute, or in a generalized selection service, the generalized access request activity attribute.

## 13.7 Access passwords

*(amend 1st paragraph, page 21)*

The access passwords parameter provides passwords associated with the actions specified in the requested access parameter. Passwords for both the object's access controls and the specified pathname's path access controls may be specified. This parameter is only available if the security attribute group has been negotiated. The range of values the parameter takes is equal to that defined for the current access passwords activity attribute. This parameter is used to set the current access passwords activity attribute, or in a generalized selection service, the generalized access passwords activity attribute.

### 13.8 Concurrency control

*(append after 1st paragraph, page 21)*

NOTE – Concurrency control in this standard is only defined for actions on file object contents.

*(append after clause 13.13, page 22)*

### 13.14 Attribute names

The attribute names parameter indicates which object attributes from the kernel or negotiated attribute groups, given in the virtual filestore definition, are to be read. The parameter is a list, each element of which names an object attribute defined for any object type, as in ISO 8571–2. This parameter is not used to set any activity attributes.

### 13.15 Attribute value assertions

The attribute value assertions parameter consists of an attribute value assertion list. Its value is a set of legally related lists of attribute patterns that are used for matching objects. When used in conjunction with the generalized selection mechanism, the results of this parameter are used to set the generalized selection group activity attributes.

### 13.16 Create password

The create password is an optional parameter which may be required to establish that the user identified by the initiator identity parameter of the FTAM regime establishment service has permission to create objects in the current filestore. This parameter is not used to set any activity attributes.

### 13.17 Destination file-directory

The destination file-directory parameter is used to identify a target destination for one or more operations. The range of values this parameter may take is that for a valid pathname. The pathname must identify an existing file-directory object within the current filestore. This parameter does not set any activity attributes except when used with the change current name prefix service, in which case it affects the current name prefix activity attribute.

### 13.18 Objects attributes list

The objects attributes list parameter consists of a set of object attribute lists, as defined in clause 13.5. This parameter is returned as a part of a response or confirm primitive. It does not set any activity attributes.

### 13.19 Override

The override parameter defined the action to be taken if the named object already exists. The values of the parameter are:

a) fail the create action if the object already exists;

b) select the object if it already exists;

c) delete the object if it already exists and create a new object using the old object's attributes (effectively erase any contents of the existing object and select it);

d) delete the object if it already exists and create a new object using the initial attributes provided on the filestore service primitive.

The values (c) and (d) are dependent upon the provision of a delete password in the access passwords parameter where required by the access control attribute of the existing object. Not every creation service supports every override option. This parameter is not used to set any activity attributes.

### 13.20 Error Action

The error action parameter is used by services dealing with more than one object at a time via the generalized selection group. It carries instructions as to how to proceed when an error is detected by the responder. This parameter may take on the following values:

a) terminate – this value indicates that if an error is detected during the service, that all processing should cease, the action result should indicate the error encountered, and suitable diagnostics be returned. Any actions taken up to the point of error detection are not rolled back.

b) continue – this value indicates that if an error is detected during the service, that the service continue with the next object in the generalized selection list. Upon completion of the service, the action result should contain an informational message noting that one or more errors had occurred, and one or more diagnostic messages should be generated for each object for which an error was encountered.

### 13.21 Last member indicator

The Last Member indicator parameter is used by the responder on F-SELECT-ANOTHER primitive, to indicate that the currently selected object is last in the group of objects selected through generalized selection. This parameter can be set by the responder to indicate to the initiator that the next select request may result in a selection of an object with a previously selected pathname from the pathname group.

### 13.22 Scope

The scope parameter is used by the initiator to indicate the scope of objects for retrieving, on F-LIST and F-GROUP-SELECT services. It specifies the root file-directory of the retrieval scope, which may be a subtree in the whole filestore, by means of a complete or incomplete pathname (a reference object linked to a file-directory is allowed, in which case the referenced file-directory is a root of the retrieval scope subtree). It also indicates whether the retrieval scope is direct child objects of the specified root file-directory or all descendants within the whole subtree.

### 13.23 Referent Indicator

The referent indicator parameter is used by the responder in F-SELECT-ANOTHER (and F-SELECT) primitives to indicate that the currently selected object has been selected such that the last object in the pathname is a reference. This parameter may be set to true by the responder to indicate that the current pathname used is a reference. Note that the parameter is optional in all cases to allow for responders to conceal the existence of references for security purposes.

### 13.24 Request operation result

The request operation result parameter is used by the initiator on the F-GROUP-DELETE, F-GROUP-MOVE and F-GROUP-COPY primitives to indicate that result of the operations within the grouped services be returned. The value of this parameter is set by the initiator to request either an overall summary of the operation or the list of original pathnames successfully acted upon in the service.

### 13.25 Operation result

The operation result parameter is used by the responder on F-GROUP-DELETE, F-GROUP-MOVE and F-GROUP-COPY primitives to retum the result of the operations within these group services. The value will be a choice of either the number of the files successfully acted upon or the complete list of the original pathnames.

### 14.1.2.12 Functional units

*(amend 1st paragraph, page 24)*

The requested functional units parameter negotiates the set of file service functional units, excluding the kernel, to be available from the negotiated service class on the application association (see clauses 8.1 and 10.3). On the request and indication primitives the parameter carries the full functional unit capability of the initiator. On the response and confirm primitives it carries all the functional units to be available on the association. When the required service class permits, the set may contain the optional functional units "read", "write", "file access", "limited file management", "enhanced file management", "limited filestore management", "enhanced filestore management", "object manipulation", "group manipulation", "grouping", and "FADU locking". For use by the internal file service, if the service class permits, it may also contain the "recovery" and "restart" functional units. The recovery and restart functional units are only visible in the IFS and are thus never explicitly requested in the external file service. They are indirectly requested via the FTAM quality of service parameter (see clause 10.2). This parameter is not used to set any activity attributes

*(insert prior to clause 15, page 27)*

## 14a Filestore services

These services provide filestore management capability to the file service user. They are not concerned with the establishment of regimes. The services provide facilities for interrogation of pathnames and attributes of objects in the filestore, and modification of the current name prefix activity attribute.

### 14a.1 Change current name prefix

#### 14a.1.1 Function

This service changes the responder's value of the current name prefix activity attribute by specifying a pathname of a file-directory object within the virtual filestore. This primitive may only be issued outside of any object selection regime. Upon receiving the F–CHANGE–PREFIX indication, the responder verifies that the indicated file-directory exists, the initiator has "pass" access through the pathname specified, and the initiator has at least one object access permission ("read", "insert", "replace", "extend", "erase", "read attribute", "change attribute", "delete", or "link") when using that pathname. If these three conditions are met, the responder changes its current name prefix activity attribute to that pathname. The result of the operation is then returned to the initiator by means of the F–CHANGE–PREFIX response primitive's action result parameter.

#### 14a.1.2 Types of primitives and parameters

Table 14a indicates the types of primitives and the parameters needed for changing the current name prefix.

##### 14a.1.2.1 Action result

The action result parameter is defined in clause 13.2.

##### 14a.1.2.2 Destination file-directory

The destination file-directory parameter is defined in clause 13.17. It identifies an existing file-directory object to which the current name prefix activity attribute is to be set. The responding application may optionally return a value confirming the newly set current name prefix.

##### 14a.1.2.3 Reset

The reset parameter is a boolean value indicating whether a destination file-directory in the format of an incomplete pathname should be resolved from the existing current name prefix, or from the current name prefix assigned at the creation of the FTAM regime. When the destination file directory is specified in the form of a complete pathname the reset parameter is not significant. A value of true implies that the current name prefix should be reset to the current name prefix assigned at the creation of the FTAM regime prior to attempting to set it from the supplied destination file-directory parameter.

Note that a reset value of true and a destination file-directory value of an incomplete pathname containing no elements resets the current name prefix to the value assigned at the creation of the FTAM regime.

##### 14a.1.2.4 Access passwords

The access passwords parameter is defined in clause 13.7. This parameter is used to establish access rights to the specified directory by the filestore user identified on the F–INITIALIZE service primitive. This parameter does not set any activity attributes, nor is any access assumed in any later services based on of the value of this parameter.

##### 14a.1.2.5 Diagnostic

The diagnostic parameter is defined in clause 13.13.

### 14a.2 List file-directory

**Table 14a – F–CHANGE–PREFIX parameters**

| Parameter | F-CHANGE-PREFIX request | F-CHANGE-PREFIX indication | F-CHANGE-PREFIX response | F-CHANGE-PREFIX confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Destination file-directory | Mandatory | Mandatory(=) | Optional | Optional(=) |
| Reset | Mandatory | Mandatory(=) | | |
| Access passwords | Optional | Optional(=) | | |
| Diagnostic | | | Optional | Optional |

**Table 14b – F–LIST parameters**

| Parameter | F-LIST request | F-LIST indication | F-LIST response | F-LIST confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attribute value assertions | Mandatory | Mandatory(=) | | |
| Access passwords | Optional | Optional(=) | | |
| Attribute names | Mandatory | Mandatory(=) | | |
| Scope | Mandatory | Mandatory(=) | | |
| Objects attributes list | | | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional |

### 14a.2.1 Function

This service enumerates objects whose attributes satisfy the attribute value assertions given. The request and indication service primitives convey an attribute value assertion list. The action fails if

a) no object exists as specified by the attribute value assertion list;

b) the attribute value assertion list references attributes from attribute groups that were not successfully negotiated.

Upon receiving the F–LIST indication, the filestore provider searches the filestore for objects conforming to the attribute value assertion list, and reads the requested attributes of each object identified. The results are then reported by issuing the F–LIST response primitive.

NOTES

1) According to the definition of the attribute value assertion list search mechanism (ISO 8571-2, clause 9a), objects to which this filestore user does not have read-attribute access are not reported, as well as any objects under file-directories to which the filestore user does not have pass and read access permission.

2) This service does not detect duplications of objects due to multiple pathnames. If the same object is accessible by multiple pathnames, and more than one of the pathnames conform to the attribute value assertion list, then the object will be reported on multiple times, once by each conforming pathname.

3) It should not be considered a requirement that the entire real filestore be searched for objects conforming to an attribute value assertion list. Real implementations may incorporate search optimizations, such as limiting the scope of the search to file-directories described in the attribute value assertion list.

### 14a.2.2 Types of primitives and parameters

Table 14b indicates the types of primitives and the parameters needed for listing the contents of directories.

#### 14a.2.2.1 Action result

The action result parameter is defined in clause 13.2.

#### 14a.2.2.2 Attribute value assertions

The attribute value assertions parameter is defined in clause 13.15. It is used to restrict the list of objects to those which match the attribute value assertion list.

#### 14a.2.2.3 Access passwords

The access passwords parameter is defined in clause 13.7. It is used to establish read-attribute access permission to the objects being listed, and read access permission to the file-directories containing them.

#### 14a.2.2.4 Attribute names

The attribute names parameter is defined in clause 13.14.

#### 14a.2.2.5 Objects attributes list

The objects attributes list parameter is defined in clause 13.18. It is used to return the list of objects conforming to the attribute value assertion list, and their corresponding attributes. The presence of the object attributes list parameter is conditional on the inclusion in the attribute names parameter of at least one attribute with a returnable value.

#### 14a.2.2.6 Scope

This parameter is described in clause 13.22.

#### 14a.2.2.7 Diagnostic

The diagnostic parameter is defined in clause 13.13.

## 14b Generalized filestore services

These services provide the mechanism for identifying a group of object pathnames by attribute value assertions, then operating on the objects identified by the group as a whole. They can be moved to a different location within the filestore, duplicated within the filestore, their attributes can be modified, or the pathnames of the group can be interrogated.

Members of the identified group can also be operated on individually under the object select regime.

### 14b.1 Generalized selection

#### 14b.1.1 Function

The generalized selection service identifies a group of pathnames matching a given attribute value assertions list. Only existing objects of type 'file' and references to objects of type 'file' are considered for generalized selection. The group of pathnames identified consists of all pathnames:

a) whose object attributes satisfy the given attribute value assertion list; and

b) for which the initiating user demonstrates authorization for access to the identified object. Generalized selection fails if there are no objects that satisfy the given attribute value assertions.

NOTES

1) According to the definition of the attribute value assertion list search mechanism (ISO 8571-2, clause 9a), objects to which this filestore user does not have read-attribute access are not included, as well as any objects under file-directories to which the filestore user does not have pass and read access permission.

2) This service does not detect duplications of objects due to multiple pathnames. If the same object is accessible by multiple pathnames, and more than one of the pathnames conform to the attribute value assertion list, then the object will be included multiple times, once by each conforming pathname.

3) It should not be considered a requirement that the entire real filestore be searched for objects conforming to an attribute value assertion list. Real implementations may incorporate search optimizations, such as limiting the scope of the search to file-directories described in the attribute value assertion list.

These primitives may only be issued within an FTAM regime. No other regimes may be currently established when issuing this service. The filestore provider performs the generalized selection action after receiving the F-GROUP-SELECT indication and before issuing an F-GROUP-SELECT response primitive with an action result indicating success.

#### 14b.1.2 Types of primitives and parameters

Table 14c indicates the types of primitives and the parameters needed for generalized selection.

##### 14b.1.2.1 Action result

The action result parameter is defined in clause 13.2.

##### 14b.1.2.2 State result

The state result parameter is defined in clause 13.1.

##### 14b.1.2.3 Attribute value assertions

The attribute value assertions parameter is defined in clause 13.15. This attribute value assertion list identifies the group of pathnames used to set the generalized select group activity attribute.

##### 14b.1.2.4 Requested access

The Requested access parameter is defined in clause 13.6. This parameter is used to identify actions that may be performed on the objects identified by the pathnames considered for the generalized select group. When taken in conjunction with the access passwords parameter, pathnames included in the group must identify objects for which the initiating filestore user has permissions for the access requested.

This parameter also sets the generalized access request activity attribute, used to establish select regimes on objects taken from the generalized select group.

##### 14b.1.2.5 Access passwords

The access passwords parameter is defined in clause 13.7. It's presence is conditional on the security attribute group being successfully negotiated. This parameter is used to verify access permission to the objects identified by the generalized select group. It also sets the generalized access passwords activity attribute, used to establish select regimes on objects taken from the generalized select group.

##### 14b.1.2.6 Concurrency control

The concurrency control parameter is defined in clause 13.8. No concurrency control is set during the establishment of the generalized selection regime.

Table 14c – F–GROUP–SELECT parameters

| Parameter | F-GROUP-SELECT request | F-GROUP-SELECT indication | F-GROUP-SELECT response | F-GROUP-SELECT confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attribute value assertions | Mandatory | Mandatory(=) | | |
| Requested access | Mandatory | Mandatory(=) | | |
| Access passwords | Conditional | Conditional(=) | | |
| Concurrency control | Optional | Optional(=) | | |
| Maximum set size | Optional | Optional(=) | | |
| Scope | Mandatory | Mandatory(=) | | |
| Account | Optional | Optional(=) | | |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |

This parameter is only used to set the generalized concurrency control activity attribute, used to establish select regimes on objects taken from the generalized select group.

14b.1.2.7 Maximum set size

The maximum set size parameter gives the maximum number of pathnames in the generalized select group that the initiator will handle. Its value is an integer. If no value is supplied, then no restriction is to be placed on the group size. If the number of pathnames in the generalized select group exceeds this bound, then the generalized selection regime is not established, no generalized select group is established, and an error is reported to the initiator.

14b.1.2.8 Account

The account parameter is defined in clause 13.3. The value given overrides any value previously set by the F–INITIALIZE request for the duration of the generalized selection regime. This account may be explicitly overridden in nested regimes.

14b.1.2.9 Shared ASE information

The shared ASE information parameter is defined in clause 13.10.

14b.1.2.10 Diagnostic

The diagnostic parameter is defined in clause 13.13.

14b.1.2.11 Scope

This parameter is described in clause 13.22.

## 14b.2 Generalized deletion

### 14b.2.1 Function

Table 14d – F–GROUP–DELETE parameters

| Parameter | F-GROUP-DELETE request | F-GROUP-DELETE indication | F-GROUP-DELETE response | F-GROUP-DELETE confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Charging | | | Conditional | Conditional(=) |
| Request operation result | Optional | Optional(=) | | |
| Operation result | | | Conditional | Conditional(=) |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |

The F–GROUP–DELETE service removes the file objects and resets to null the generalized select group activity attribute value. The file objects are removed in such a way that the objects cease to exist and hence are not available for selection.

NOTE – Implementations should be aware that this is a destructive service and will delete all the files whose pathnames are in the current selection group from the filestore.

This primitive may only be used within the FTAM regime.

The filestore provider performs the generalized deletion action after receiving the F–GROUP–DELETE indication primitive, and before issuing the F–GROUP–DELETE response primitive.

#### 14b.2.2 Types of primitives and parameters

Table 14d indicates the types of primitives and the parameters needed for generalized deletion.

##### 14b.2.2.1 Action result

The action result parameter is defined in clause 13.2.

##### 14b.2.2.2 Charging

The charging parameter is defined in clause 13.4. Charges are only reported against an overriding account set up at regime establishment.

##### 14b.2.2.3 Request operation result

The request opertaion result parameter is defined in 13.24. This parameter is optional.

##### 14b.2.2.4 Operation result

The operation result parameter is defined in 13.25. This parmeter is conditional on the presence of the request operation result parameter.

##### 14b.2.2.5 Shared ASE information

The shared ASE information parameter is defined in clause 13.10.

##### 14b.2.2.6 Diagnostic

The diagnostic parameter is defined in clause 13.13.

### 14b.3 Group move

#### 14b.3.1 Function

The F–GROUP–MOVE service moves the file objects identified by the generalized select group by changing their location in the filestore structure and resets to null the generalized select group activity attribute value.. The destination is specified by the destination file-directory parameter. This parameter changes the primary pathname object attribute of all objects in the group. Other object attributes may also be changed by use of the attributes parameter.

Moving an object consists of entering a parenthood relationship for the given object into a named file-directory, removing the parenthood relationship for that object from the old parent file-directory for that object, and updating the primary pathname attribute of the object to reflect the new primary pathname.

When references to files exist in the generalized selection group, the file object is moved to the new location, not the reference.

Prior to moving each object, the filestore user identified in the F–INITIALIZE service (along with the access passwords in the generalized access passwords activity attribute) must have change-attribute permission to the object via the corresponding pathname. Also, the filestore user (along with the create password) must have permission to create objects in this filestore, and insert permission in the destination file-directory.

The F–GROUP–MOVE service is not an atomic action. Move operations are performed one at a time, until either an error occurs or the list of pathnames is exhausted. The error action parameter determines whether or not an error terminates the service. A shared ASE information parameter is provided for use should the use of some other ASE such as CCR be desired to form an atomic action from this service.

#### 14b.3.2 Types of primitives and parameters

Table 14e indicates the types of primitives and the parameters needed for group move operations.

##### 14b.3.2.1 Action result

The action result parameter is defined in clause 13.2.

##### 14b.3.2.2 Destination file-directory

The destination file-directory parameter is defined in clause 13.17. It identifies the file-directory that should maintain the parenthood relationship to the objects identified by the pathnames in the generalized select group.

##### 14b.3.2.3 Override

The override parameter is defined in clause 13.19 and only applies to the create part of the action. Only the following override options are supported for the F–GROUP–MOVE service:

a) fail the action if the file already exists;

b) delete the object if it already exists and create a new object using the initial attributes provided on the filestore service primitive.

#### 14b.3.2.4 Error action

The error action parameter is described in clause 13.20.

#### 14b.3.2.5 Request operation result

The request operation result parameter is defined in 13.24. This parameter is optional.

#### 14.b.3.2.6 Operation result

The operation result parameter is defined in 13.25. This parameter is conditional on the presence of the request operation result parameter.

#### 14b.3.2.7 Create password

The create password parameter is described in clause 13.16.

#### 14b.3.2.8 Access passwords

The access passwords parameter is described in clause 13.7. It is used to establish delete access to any objects which may be overridden by this service. The presence of the access passwords parameter is conditional on the successful negotiation of the security attribute group during the establishment of the FTAM regime.

#### 14b.3.2.9 Attributes

The attributes parameter is described in clause 13.5. This parameter is used to set any additional attributes (other than primary pathname) on the objects being moved.

#### 14b.3.2.10 Diagnostic

The diagnostic parameter is described in clause 13.13.

### 14b.4 Group copy

#### 14b.4.1 Function

The F–GROUP–COPY service copies the file objects identified by the generalized select group by creating new objects with duplicate contents in the specified destination location in the filestore structure and the value of the generalized select group activity attribute remains unchanged. The destination is specified by the destination file-directory parameter. Object attributes may also be changed by use of the attributes parameter.

When references to files exist in the generalized selection group, the file object is duplicated into the new location, not the reference.

Prior to copying each object, the filestore user identified in the F–INITIALIZE service (along with the access passwords in the generalized access passwords activity attribute) must have read-attributes and read permission to the object via the corresponding pathname. Also, the filestore user (along with the create password) must have permission to create objects in this filestore, and insert permission in the destination file-directory.

The F–GROUP–COPY service is not an atomic action. Copy operations are performed one at a time, until either an error occurs or the list of pathnames is exhausted. The error action parameter determines whether or not an error terminates the service. A shared ASE information parameter is provided for use

**Table 14e – F–GROUP–MOVE parameters**

| Parameter | F-GROUP-MOVE request | F-GROUP-MOVE indication | F-GROUP-MOVE response | F-GROUP-MOVE confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Destination file-directory | Mandatory | Mandatory(=) | Optional | Optional(=) |
| Override | Mandatory | Mandatory(=) | | |
| Error action | Mandatory | Mandatory(=) | | |
| Request operation result | Optional | Optional(=) | | |
| Operation result | | | Conditional | Conditional(=) |
| Create password | Optional | Optional(=) | | |
| Access passwords | Conditional | Conditional(=) | | |
| Attributes | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |

Table 14f – F–GROUP–COPY parameters

| Parameter | F-GROUP-COPY request | F-GROUP-COPY indication | F-GROUP-COPY response | F-GROUP-COPY confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Destination file-directory | Mandatory | Mandatory(=) | Optional | Optional(=) |
| Override | Mandatory | Mandatory(=) | | |
| Error action | Mandatory | Mandatory(=) | | |
| Request operation result | Optional | Optional(=) | | |
| Operation result | | | Conditional | Conditional(=) |
| Create password | Optional | Optional(=) | | |
| Access passwords | Conditional | Conditional(=) | | |
| Attributes | Optional | Optional(=) | | |
| Diagnostic | | | Optional | Optional |

should the use of some other ASE such as CCR be desired to form an atomic action from this service.

### 14b.4.2 Types of primitives and parameters

Table 14f indicates the types of primitives and the parameters needed for group copy operations.

14b.4.2.1 Action result

The action result parameter is defined in clause 13.2.

14b.4.2.2 Destination file-directory

The destination file-directory parameter is defined in clause 13.17. It identifies the file-directory in which the duplicate objects are to be created.

14b.4.2.3 Override

The override parameter is defined in clause 13.19 and only applies to the create part of the action. Only the following override options are supported for the F–GROUP–COPY service:

a) fail the action if the file already exists;

b) delete the object if it already exists and create a new object using the initial attributes provided on the filestore service primitive.

14b.4.2.4 Error action

The error action parameter is described in clause 13.20.

14b.4.2.5 Request operation result

The request operation result parameter is defined in 13.24. This parameter is optional.

14.b.4.2.6 Operation result

The operation result parameter is defined in 13.25. This parameter is conditional on the presence of the request operation result parameter.

14b.4.2.7 Create password

The create password parameter is described in clause 13.16.

14b.4.2.8 Access passwords

The access passwords parameter is described in clause 13.7. It is used to establish delete access to any objects which may be overridden by this service. The presence of the access passwords parameter is conditional on the successful negotiation of the security attribute group during the establishment of the FTAM regime.

14b.4.2.9 Attributes

The attributes parameter is described in clause 13.5. This parameter is used to set any additional attributes (other than primary pathname) on the object being copied.

14b.4.2.10 Diagnostic

The diagnostic parameter is described in clause 13.13.

### 14b.5 Group list

14b.5.1 Function

The F–GROUP–LIST service returns the set of pathnames maintained by the filestore provider in the generalized select group activity attribute, and selected object attributes of the represented objects.

Table 14g – F–GROUP–LIST

| Parameter | F-GROUP-LIST request | F-GROUP-LIST indication | F-GROUP-LIST response | F-GROUP-LIST confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attribute names | Mandatory | Mandatory(=) | | |
| Objects attributes list | | | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional |

Upon receiving the F–GROUP–LIST indication, the filestore provider reads the requested attributes of each object identified by pathname in the generalized select group. Any pathname for which access permission is denied is removed from the list without notification of the filestore user. The results are then reported by issuing the F–GROUP–LIST response primitive.

### 14b.5.2 Types of primitives and parameters

Table 14g indicates the types of primitives and the parameters needed for listing the generalized selection group.

#### 14b.5.2.1 Action result

The action result parameter is described in clause 13.2.

#### 14b.5.2.2 Attribute names

The attribute names parameter is described in clause 13.14.

#### 14b.5.2.3 Objects attributes list

The objects attributes list parameter is described in clause 13.18. The presence of the object attributes list parameter is conditional on the inclusion in the attribute names parameter of at least one attribute with a returnable value.

#### 14b.5.2.4 Diagnostic

The diagnostic parameter is described in clause 13.13.

## 14b.6 Group change attribute

### 14.b.6.1 Function

The F-GROUP-CHANGE-ATTRIB service modifies the attributes of the file objects in the generalized select group. The primitive conveys a list of attribute names and values.

The filestore provider performs the change attribute action and sets the two object attributes, date and time of last attribute modifications and identity of last attribute modifier, after receiving the F-GROUP-CHANGE-ATTRIB indication, and before issuing the F-GROUP-CHANGE-ATTRIB response primitive. If any change causes an action result other than success to be generated, none of the requested changes against the file are made. The action result for that file is "unsuccessful".

The F-GROUP-CHANGE-ATTRIB service is not an atomic action. Change attribute operations are performed one at a time, until either an error occurs or the list of pathnames is exhausted. The error action parameter determines whether or not an error terminates the service. A shared ASE information parameter is provided, should some other ASE, such as CCR, desire to form an atomic action using this service.

### 14b.6.2 Type of primitives and parameters

Table 14h indicates the types of primitives and the parameters needed to change attributes of grouped files.

#### 14b.6.2.1 Action result

The action result parameter is defined in clause 13.2.

#### 14b.6.2.2 Error Action

The error action parameter is defined in clause 13.20.

#### 14b.6.2.3 Attribute

The attribute parameter, defined in 13.5, indictes on the request and indication primitives which object attributes given in the virtual filestore definition are to be changed and what the new values are. The parameter is a list, each element of which names a file attribute defined in ISO 8571-2 and provides a new value for it.

#### 14b.6.2.4 Request operation result

The request opertaion result parameter is defined in 13.24. This parameter is optional.

#### 14b.6.2.5 Operation result

The operation result parameter is defined in 13.25. This parmeter is conditional on the presence of the re-

Table 14h – F–GROUP–CHANGE–ATTRIB parameters

| Parameter | F-GROUP-CHANGE-ATTRIB request | F-GROUP-CHANGE-ATTRIB indication | F-GROUP-CHANGE-ATTRIB response | F-GROUP-CHANGE-ATTRIB confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attributes | Mandatory | Mandatory(=) | | |
| Error action | Mandatory | Mandatory(=) | | |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Request operation result | Optional | Optional(=) | | |
| Operation result | | | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional(=) |

quest operation result parameter.

14b.6.2.6 Shared ASE information

The shared ASE information is defined in clause 13.10.

14b.6.2.7 Diagnostic

The diagnostic parameter is described in clause 13.13.

**Append the following parameter to Table 15**

| Parameter | F-SELECT request | F-SELECT indication | F-SELECT response | F-SELECT confirm |
|---|---|---|---|---|
| .<br>.<br>.<br>Referent indicator | | | .<br>.<br>.<br>Optional | .<br>.<br>.<br>Optional |

*(amend title to clause 15, page 27)*

## 15 Object selection regime control

*(amend 1st paragraph)*

These services control the regime which binds an identified object to an FTAM regime. The object selection service is pertinent to files only, unless the limited filestore management is available. This is because selection of objects other than files is permitted if and only if the limited filestore management functional unit is available.

Different services are used depending on whether the object exists before the selection or not, the kind of object to be selected or created, and on whether it is to exist after the selection or not.

*(amend title to clause 15.1, page 27)*

### 15.1 Object selection service

### 15.1.1 Function

*(amend 1st paragraph, page 27)*

This service selects an object which already exists, by specifying an existing pathname in the attributes parameter to identify the object unambiguously. These primitives may only be issued if there is no current object selection within the current FTAM regime. The filestore provider performs the select object action after receiving the F–SELECT indication and before issuing an F–SELECT response primitive with an action result indicating success.

The object selection service does not affect, and is not affected by, any existing activity attributes that may have been established by a generalized select.

### 15.1.2 Types of primitives and parameters

*(amend 1st paragraph, page 27)*

Table 15 indicates the types of primitives and the parameters needed for object selection.

15.1.2.3 Attributes

*(amend 1st paragraph, pages 27 to 28)*

The attributes parameter is described in clause 13.5. On F–SELECT the parameter carries only a single object pathname, which may or may not be a primary pathname. The pathname is used to identify the object in the object selection process. On the F–SELECT response and confirm primitives only the pathname is returned which may not be identical to that issued on the request and indication.

15.1.2.8 Account

*(amend 1st paragraph, page 28)*

The account parameter is described in clause 13.3. The value given overrides any value previously set by the F–INITIALIZE or F–GROUP–SELECT requests for the duration of this select regime.

*(append after subclause 15.1.2.9, page 28)*

15.1.2.9 Referent indicator

The Referent indicator parameter is described in 13.23, and can be used if and only if the limited filestore management functional unit is available.

*(append after clause 15.1, page 28)*

### 15.1a Select another

### 15.1a.1 Function

The F–SELECT–ANOTHER service selects an object from the set of previously unselected pathnames in the generalized select group, setting the current access request, current concurrency control, current access passwords, and current access control activity attributes from the generalized access request, generalized concurrency control, generalized access passwords, and generalized access control activity attributes, respectively.

If no pathnames exist in the generalized select group, then an error indication is returned. A filestore user may cycle through the set of pathnames multiple times. A last member indicator parameter is returned on the F–SELECT–ANOTHER response and confirm primitives, to indicate that the currently selected object is the last pathname in the group.

**Table 15a – F-SELECT-ANOTHER parameters**

| Parameter | F-SELECT-ANOTHER request | F-SELECT-ANOTHER indication | F-SELECT-ANOTHER response | F-SELECT-ANOTHER confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| State result | | | Mandatory | Mandatory |
| Last member indicator | | | Optional | Optional |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |
| Referent indicator | | | Optional | Optional |

NOTES

1) No order is implied in the generalized select group, and there is no guarantee that pathnames will be chosen in the same order as a previous pass through the group, or as listed by an F–GROUP–LIST service.

2) No concurrency control is implied by the generalized selection service. Objects may be removed from the generalized select group due to object deletion or changes in access control during a given FTAM regime without notification. Once an object is selected by the F–SELECT–ANOTHER service, the concurrency control specified in the generalized concurrency control activity attribute is in force for that object only during the life of this select regime.

### 15.1a.2 Types of primitives and parameters

Table 15a indicates the types of primitives and the parameters needed for selecting another pathname from the generalized selection group.

#### 15.1a.2.1 Action result

The action result parameter is described in clause 13.2.

#### 15.1a.2.2 State result

The state result parameter is described in clause 13.1.

#### 15.1a.2.3 Last Member indicator

The Last Member indicator is described in clause 13.21.

#### 15.1a.2.4 Referent indicator

The referent indicator is described in clause 13.23.

#### 15.1a.2.5 Shared ASE information

The shared ASE information parameter is described in clause ]13.10.

#### 15.1a.2.6 Diagnostic

The diagnostic parameter is described in clause 13.13.

*(amend title to clause 15.2, page 28)*

## 15.2 Object deselection service

### 15.2.1 Function

*(amend 1st paragraph, page 28)*

The F–DESELECT service releases the binding between the select regime and the object. The object continues to exist and is available for subsequent selection. The primitives may only be used while an object is selected. The object select regime is always terminated following an F–DESELECT response or confirm for all values of the action result parameter

*(amend 2nd paragraph, page 28)*

The filestore provider performs the deselect object action after receiving the F–DESELECT indication primitive, and before issuing the F–DESELECT response primitive. In the external service all concurrency control is released when the object is deselected, except when performed within a CCR atomic action. The current account activity attribute is reset to that of the FTAM regime.

### 15.2.2 Types of primitives and parameters

*(amend 1st paragraph, page 28)*

Table 16 indicates the types of primitives and the parameters needed for object deselection.

#### 15.2.2.2 Charging

Table 17a – F–CREATE–DIRECTORY parameters

| Parameter | F-CREATE-DIRECTORY request | F-CREATE-DIRECTORY indication | F-CREATE-DIRECTORY response | F-CREATE-DIRECTORY confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| State result | | | Mandatory | Mandatory |
| Attributes | Mandatory | Mandatory(=) | Mandatory | Mandatory(=) |
| Create password | Optional | Optional(=) | | |
| Requested access | Mandatory | Mandatory(=) | | |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Account | Optional | Optional(=) | | |
| Diagnostic | | | Optional | Optional |

*(amend 1st paragraph)*

The charging parameter is described in clause 13.4. Charges are only reported on deselection against an overriding account set up at object selection.

15.3.2.3 Override

*(amend 1st paragraph, page 29)*

The override parameter is defined in clause 13.19.

15.3.2.5 Create password

*(amend 1st paragraph, page 29)*

The create password parameter is defined in clause 13.16.

*(add after clause 15.3, page 30)*

### 15.3a File-directory creation

### 15.3a.1 Function

The F–CREATE–DIRECTORY service causes a file-directory to be created and establishes a selection of the newly created file-directory. If an object with the specified pathname already exists then the request is rejected. The file-directory create service may only be used if there is no currently selected object.

The filestore provider performs the create file-directory action after receiving the F–CREATE–DIRECTORY indication primitive, and before issuing the F–CREATE–DIRECTORY response primitive with action result parameters indicating success.

### 15.3a.2 Types of primitives and parameters

Table 17a indicates the types of primitives and the parameters needed for creating a file-directory.

15.3a.2.1 Action result

The action result parameter is described in clause 13.2.

15.3a.2.2 State result

The state result parameter is described in clause 13.1.

15.3a.2.3 Attributes

The attributes parameter is described in clause 13.5. For those attribute groups negotiated on F–INITIALIZE, the attribute values given are associated with the newly created file-directory. If, for any particular attribute a value is not given, a default is determined locally by the responder; this value may be "no value available". For attributes other than pathname and permitted actions, the responder may change a value proposed by the initiator to "no value available" but may not assign any different value to it. The responder does report any local modification to the values of the pathname or permitted actions attributes. In addition to the attributes in the attribute groups negotiated on F–INITIALIZE the attributes in the kernel attribute group are also associated with the newly created file-directory.

15.3a.2.4 Create password

The create password parameter is described in clause 13.16.

15.3a.2.5 Requested access

The requested access parameter is described in clause 13.6. The parameter is used to establish the access available in the select regime following the create action. The requested access during the select regime need not be the full capability of the file-directory. Hence the access requested must be a subset of those capabilities established with the attributes parameter. The established capabilities include all responder provided defaults which are

**Table 17b – F–LINK parameters**

| Parameter | F-LINK request | F-LINK indication | F-LINK response | F-LINK confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| State result | | | Mandatory | Mandatory |
| Attributes | Mandatory | Mandatory(=) | Mandatory | Mandatory(=) |
| Target object | Mandatory | Mandatory(=) | Mandatory | Mandatory(=) |
| Create password | Optional | Optional(=) | | |
| Requested access | Mandatory | Mandatory(=) | | |
| Access passwords | Conditional | Conditional(=) | | |
| Concurrency control | Optional | Optional(=) | | |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Account | Optional | Optional(=) | | |
| Diagnostic | | | Optional | Optional |

within the kernel attribute group or the other attribute groups negotiated at F–INITIALIZE.

If the requested access parameter is incompatible with the newly created file-directory, the select will fail even though the file-directory was created.

15.3a.2.6 Shared ASE information

The shared ASE information parameter is described in clause 13.10.

15.3a.2.7 Account

The account parameter is described in clause 13.3. The value given overrides any value previously set on the F–INITIALIZE or F–GROUP–SELECT request for the duration of this selection.

15.3a.2.8 Diagnostic

The diagnostic parameter is described in clause 13.13.

### 15.3b Reference creation

### 15.3b.1 Function

The F–LINK service causes a reference to be created and establishes a selection of the referenced object. If an object with the specified reference pathname already exists then the request is rejected. This service also fails if the object to which the reference is to be created does not exist. The create reference service may only be used if there is not a currently selected object.

The filestore provider performs the create reference action after receiving the F–LINK indication primitive, and before issuing the F–LINK response primitive with action result parameters indicating success.

### 15.3b.2 Types of primitives and parameters

Table 17b indicates the types of primitives and the parameters needed for creating a reference.

15.3b.2.1 Action result

The action result parameter is described in clause 13.2.

15.3b.2.2 State result

The state result parameter is described in clause 13.1.

15.3b.2.3 Attributes

The attributes parameter is described in clause 13.5. For those attribute groups negotiated on F–INITIALIZE, the attribute values given are associated with the newly created reference. If, for any particular attribute a value is not given, a default is determined locally by the responder; this value may be “no value available”. For attributes other than pathname and permitted actions, the responder may change a value proposed by the initiator to “no value available” but may not assign any different value to it. The responder does report any local modification to the values of the pathname or permitted actions attributes. In addition to the attributes in the attribute groups negotiated on F–INITIALIZE the attributes in the kernel attribute group are also associated with the newly created reference.

15.3b.2.4 Target object

The target object parameter identifies an existing file or file-directory object within the filestore by pathname. This is the object to which the reference will be linked. The filestore user must have been granted link permission for this object for the service to succeed.

15.3b.2.5 Create password

The create password parameter is described in clause 13.16.

15.3b.2.6 Requested access

The requested access parameter is described in clause 13.6. The parameter is used to establish the access available to the target object in the select regime following the create reference action. The requested access during the select regime must be a subset of the access permitted to this filestore user by the target object.

If the requested access parameter is incompatible with the existing target object, the select will fail even though the reference was created.

15.3b.2.7 Access passwords

The access passwords parameter is described in clause 13.7. It is conditional on the security group being selected. If the link member of the set of access passwords does not match any of the non-empty password strings in the link member of the target object's access control attribute, the reference is not created and the select regime is not established. If the remaining members of the set of access passwords do not match any of the non-empty password strings in the access control attribute of the target object, the reference may be created, but the select regime is not established. This parameter is used to set the current access passwords activity attribute.

15.3b.2.8 Concurrency control

The concurrency control parameter is described in clause 13.8. This parameter is used to establish concurrency controls on the target object during the select regime. It is only valid when the reference links to a file object. If the required concurrency control is not available, the select regime is not established, although the reference may be created.

15.3b.2.9 Shared ASE information

The shared ASE information parameter is described in clause 13.10.

15.3b.2.10 Account

The account parameter is described in clause 13.3. The value given overrides any value previously set on the F–INITIALIZE or F–GROUP–SELECT request for the duration of this selection.

15.3b.2.11 Diagnostic

The diagnostic parameter is described in clause 13.13.

*(amend title to clause 15.4, page 30)*

### 15.4 Object deletion service

**15.4.1 Function**

*(amend 1st paragraph, page 30)*

The F–DELETE service releases an existing object select regime in such a way that the selected object ceases to exist, and is not available for reselection. The primitives may only be issued while an object is selected. The select regime is always released following an F–DELETE response or confirm, for all possible values of the action result parameter. File-directories may not be deleted while they maintain any parenthood relationships. In this case, the select regime is terminated, but the file-directory is not deleted. The action result parameter will reflect this condition. If access to the selected object was by reference, then both the referenced object and the reference are deleted.

*(amend 2nd paragraph, page 30)*

The filestore provider performs the delete object action after receiving the F–DELETE indication, and before issuing the F–DELETE response primitive, with an action result parameter of success. The delete object action can be performed only if the initiating entity has the “delete” access control permission (see clause 13.6). This access control permission is established by the requested access, concurrency control and access passwords parameters on the F–SELECT, F–CREATE, F–CREATE–DIRECTORY, or F–LINK primitives, or by the generalized requested access, generalized concurrency control, and generalized access passwords activity attributes on an F–SELECT–ANOTHER primitive, whichever established the object selection regime.

**15.4.2 Types of primitives and parameters**

*(amend 1st paragraph, page 30)*

Table 18 indicates the types of primitives and the parameters needed for object deletion.

15.4.2.1 Action result

*(amend 1st paragraph, page 30)*

The action result parameter, defined in 13.2, indicates success or failure of the deletion action. The object is

Table 18a – F–UNLINK parameters

| Parameter | F-UNLINK request | F-UNLINK indication | F-UNLINK response | F-UNLINK confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Shared ASE information | Optional | Optional(=) | Optional | Optional(=) |
| Charging | | | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional |

always deselected whatever the value of the action result parameter.

15.4.2.3 Charging

*(amend 1st paragraph)*

The charging parameter is defined in 13.4. Charges are only reported on deselection against an overriding account set up at object selection.

*(append after clause 15.4, page 30)*

## 15.5 Reference deletion

### 15.5.1 Function

The F–UNLINK service releases an existing object select regime in such a way that the reference object identified by the current pathname activity attribute ceases to exist, and is not available for selection. The object linked by the reference continues to exist, as well as any other references to the same object. The primitives may only be issued while an object is selected and the current pathname activity attribute identifies a reference object. The select regime is always released following an F–UNLINK response or confirm, for all possible values of the action result parameter.

The filestore provider performs the delete reference action after receiving the F–UNLINK indication, and before issuing the F–UNLINK response primitive, with an action result parameter of success. The delete reference action can be performed only if the initiating entity has the "delete" access control permission (see clause 13.6) for the reference object.

### 15.5.2 Types of primitives and parameters

Table 18a indicates the types of primitives and the parameters needed for deleting a reference.

15.5.2.1 Action result

The action result parameter is described in clause 13.2.

15.5.2.2 Shared ASE information

The shared ASE information parameter is described in clause 13.10.

15.5.2.3 Charging

The charging parameter is described in clause 13.4. Charging is only reported against an overriding account for this regime.

15.5.2.4 Diagnostic

The diagnostic parameter is described in clause 13.13.

*(amend title of clause 16, page 31)*

# 16 Object management

*(amend 1st paragraph, page 31)*

These services provide object management capability to the file service user. They are not concerned with the establishment of regimes. The services provide facilities for interrogation and modification, where permitted, of object attributes. Managed objects are of type file only, unless the limited filestore management functional unit is available. This is because the selection of objects other than files is permitted if and only if the limited filestore management functional unit is available.

### 16.1.1 Function

*(amend 1st paragraph)*

The F–READ–ATTRIB service interrogates the object attributes of the selected object. The request and indication convey a list of the object attribute names for which values are to be read. The response and confirm then convey the corresponding returned values. Attribute names of attributes not supported by the object type are ignored. This allows attributes to be read from objects whose type is unknown.

### 16.1.2 Types of primitives and parameters

*(amend 1st paragraph, page 31)*

Table 19 indicates the types of primitives and the parameters needed to read object attributes.

16.1.2.2 Attribute names

*(amend 1st paragraph, page 31)*

The attribute names parameter is described in clause 13.14.

### 16.2.1 Function

*(amend 1st paragraph, page 31)*

The F–CHANGE–ATTRIB service modified the attributes of the selected object. The primitives convey a list of object attribute names and values.

*(amend 2nd paragraph, page 31)*

The filestore provider performs the change attribute action and sets the two object attributes, date and time of last attribute modification and identity of last attribute modifier, after receiving the F–CHANGE–ATTRIB indication, and before issuing the F–CHANGE–ATTRIB response primitive. If any change causes an action result other than success to be generated, none of the requested changes are made; an "unsuccessful" action result parameter is returned.

### 16.2.2 Types of primitives and parameters

*(amend 1st paragraph, page 31)*

Table 20 indicates the types of primitive and the parameters needed to change object attributes.

16.2.2.2 Attributes

*(amend 1st paragraph, page 31)*

The attributes parameter, defined in 13.5, indicates on the request and indication primitives which object attributes given in the virtual filestore definition are to be changed and what the new values are to be. The parameter is a list, each element of which names an object attribute defined in ISO 8571–2 and provides a new value for it.

*(amend 2nd paragraph, page 31)*

The presence of the attributes parameter on the response and confirm is conditional on the inclusion in the attributes parameter of at least one attribute name and value. Values for attributes not specifically requested are not changed or returned. Values for attributes not in the kernel or negotiated attribute group are not changed or returned. Values for attributes not supported by the selected object type are not changed or returned. ISO 8571–2 defines the actions permitted on attributes.

*(append after clause 16.2, page 31)*

## 16.3 Read reference attributes

### 16.3.1 Function

The F–READ–LINK–ATTRIB service interrogates the attributes of the reference identified by the current pathname activity attribute. The request and indication convey a list of the reference attribute names for which values are to be read. The response and confirm then convey the corresponding returned values.

The filestore provider performs the read reference attribute action after receiving the F–READ–LINK–ATTRIB indication, and before issuing the F–READ–LINK–ATTRIB response primitive.

### 16.3.2 Types of primitives and parameters

Table 19a indicates the types of primitives and the parameters needed for reading reference attributes.

16.3.2.1 Action result

The action result parameter is described in clause 13.2.

Table 19a – F–READ–LINK–ATTRIB parameters

| Parameter | F-READ-LINK-ATTRIB request | F-READ-LINK-ATTRIB indication | F-READ-LINK-ATTRIB response | F-READ-LINK-ATTRIB confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attribute names | Mandatory | Mandatory(=) | | |
| Attributes | | | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional |

16.3.2.2 Attribute names

The attribute names parameter is described in clause 13.14. The attribute names allowed are constrained to those defined for reference objects.

16.3.2.3 Attributes

The attributes parameter is described in clause 13.5. The presence of the attributes parameter is conditional on the inclusion in the attribute names parameter of at least one attribute with a returnable value. Values for attributes not specifically requested are not returned. Values for attributes not in the kernel attribute group or a negotiated attribute group are not returned. The value of password attributes are not returned. The responder may return "no value available" for partially supported attributes. ISO 8571–2 details the attributes which are and are not returned for reference objects.

16.3.2.4 Diagnostic

The diagnostic parameter is described in clause 13.13.

## 16.4 Change reference attributes

### 16.4.1 Function

The F–CHANGE–LINK–ATTRIB service modifies the attributes of the reference identified by the current pathname activity attribute. The primitives convey a list of reference attribute names and values.

The filestore provider performs the change reference attribute action and sets the two object attributes of the indicated reference, date and time of last attribute modification and identity of last attribute modifier, after receiving the F–CHANGE–LINK–ATTRIB indication, and before issuing the F–CHANGE–LINK–ATTRIB response primitive. If any change causes an action result other than success to be generated, none of the requested changes are made; an "unsuccessful" action result parameter is returned.

### 16.4.2 Types of primitives and parameters

Table 20a indicates the types of primitives and the parameters needed for changing reference attributes.

16.4.2.1 Action result

The action result parameter is described in clause 13.2.

16.4.2.2 Attributes

The attributes parameter is described in clause 13.5. It indicates on the request and indication primitives which reference attributes given in the virtual filestore definition are to be changed and what the new values are to be. The parameter is a list, each element of which names a reference attribute defined in ISO 8571–2 and provides a new value for it.

The presence of the attributes parameter on the response and confirm is conditional on the inclusion in the attributes parameter of at least one attribute name and value. Values for attributes not specifically requested are not changed or returned. Values for attributes not in the kernel or negotiated attribute groups are not changed or returned. ISO 8571–2

Table 20a – F–CHANGE–LINK–ATTRIB parameters

| Parameter | F-CHANGE-LINK-ATTRIB request | F-CHANGE-LINK-ATTRIB indication | F-CHANGE-LINK-ATTRIB response | F-CHANGE-LINK-ATTRIB confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Attributes | Mandatory | Mandatory(=) | Conditional | Conditional(=) |
| Diagnostic | | | Optional | Optional |

Table 20b – F–MOVE parameters

| Parameter | F-MOVE request | F-MOVE indication | F-MOVE response | F-MOVE confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Destination file-directory | Mandatory | Mandatory(=) | Optional | Optional(=) |
| Override | Mandatory | Mandatory(=) | | |
| Create password | Optional | Optional(=) | | |
| Access passwords | Conditionial | Conditional(=) | | |
| Attributes | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |

defines the actions permitted on attributes.

16.4.2.3 Diagnostic

The diagnostic parameter is described in clause 13.13.

### 16.5 Move object

### 16.5.1 Function

The F–MOVE service moves the object selected by changing its location in the filestore structure. The destination is specified by the destination file-directory parameter. This parameter changes the primary pathname object attribute of the selected object. Other object parameters may also be changed by use of the attributes parameter.

Moving an object consists of entering a parenthood relationship for the given object into a named file-directory, removing the parenthood relationship for that object from the old parent file-directory for that object, and updating the primary pathname attribute of the object to reflect the new primary pathname. If the object selected is a reference, then only the reference object is moved, not the referenced object.

Prior to moving the object, the filestore user identified in the F–INITIALIZE service must have change-attribute permission to the object via the corresponding pathname. Also, the filestore user must have permission to create objects in this filestore, and insert permission in the destination file-directory.

### 16.5.2 Types of primitives and parameters

Table 20b indicates the types of primitives and the parameters needed for moving an object within the filestore.

16.5.2.1 Action result

The action result parameter is defined in clause 13.2.

16.5.2.2 Destination file-directory

The destination file-directory parameter is defined in clause 13.17. It identifies the file-directory that should now maintain the parenthood relationship to the selected object

16.5.2.3 Override

The override parameter is defined in clause 13.19 and only applies to the create part of the action. Only the following override options are supported for the F–MOVE service:

a) fail the action if the object already exists;

b) delete any existing object that may exist with the same pathname and create a new object using the initial attributes provided on the filestore service primitive.

16.5.2.4 Create password

The create password parameter is described in clause 13.16.

16.5.2.5 Access passwords

The access passwords parameter is described in clause 13.7. It is used to establish delete access to any object which may be overridden by this service. This parameter is conditional on the security attribute group being successfully negotiated during establishment of the FTAM regime.

16.5.2.6 Attributes

The attributes parameter is described in clause 13.5. This parameter is used to change any additional attributes (other than primary pathname) on the object being moved.

16.5.2.7 Diagnostic

Table 20c – F–COPY parameters

| Parameter | F-COPY request | F-COPY indication | F-COPY response | F-COPY confirm |
|---|---|---|---|---|
| Action result | | | Mandatory | Mandatory |
| Destination | | | | |
| file-directory | Mandatory | Mandatory(=) | Optional | Optional(=) |
| Override | Mandatory | Mandatory(=) | | |
| Create password | Optional | Optional(=) | | |
| Access passwords | Conditional | Conditional(=) | | |
| Attributes | Optional | Optional(=) | Optional | Optional(=) |
| Diagnostic | | | Optional | Optional |

The diagnostic parameter is described in clause 13.13.

## 16.6 Copy object

### 16.6.1 Function

The F–COPY service copies the currently selected object by creating a new object with duplicate contents in the specified destination location in the filestore structure. The destination is specified by the destination file-directory parameter. The new object's parameters may also be set by use of the attributes parameter.

If the object selected is a reference, then only the reference is duplicated, not the referenced object.

If the object selected is a directory, then the copy will only succeed if the directory has no children.

Prior to copying the object, the filestore user identified in the F–INITIALIZE service must have read-attributes and read permission to the object via the corresponding pathname. Also, the filestore user must have permission to create objects in this filestore, and insert permission in the destination file-directory

### 16.6.2 Types of primitives and parameters

Table 20c indicates the types of primitives and the parameters needed for duplicating an object within the filestore.

#### 16.6.2.1 Action result

The action result parameter is defined in clause 13.2.

#### 16.6.2.2 Destination file-directory

The destination file-directory parameter is defined in clause 13.17. It identifies the file-directory in which the duplicate of the selected object should be placed.

#### 16.6.2.3 Override

The override parameter is defined in clause 13.19 and only applies to the create part of the action. Only the following override options are supported for the F–COPY service:

a) fail the action if the object already exists;

b) delete any existing object that may exist with the same pathname and create a new object using the initial attributes provided on the filestore service primitive.

#### 16.6.2.4 Create password

The create password parameter is described in clause 13.16.

#### 16.6.2.5 Access passwords

The access passwords parameter is described in clause 13.7. It is used to establish delete access to any object which may be overridden by this service. This parameter is conditional on the security attribute group being successfully negotiated during the establishment of the FTAM regime.

#### 16.6.2.6 Attributes

The attributes parameter is described in clause 13.5. This parameter is used to set any additional attributes (other than primary pathname) on the object being copied.

#### 16.6.2.7 Diagnostic

The diagnostic parameter is described in clause 13.13.

# Annex A
(normative)

# Diagnostic parameter values

## A.6 Applicability of the diagnostic parameter

*(amend 1st paragraph, page 48)*

The diagnostic parameter occurs on the following services:

F-INITIALIZE
F-U-ABORT
F-P-ABORT
F-CHANGE-PREFIX
F-LIST
F-GROUP-SELECT
F-GROUP-DELETE
F-GROUP-MOVE
F-GROUP-COPY
F-GROUP-LIST
F-GROUP-CHANGE-ATTRIB
F-SELECT
F-SELECT-ANOTHER
F-DESELECT
F-CREATE
F-CREATE-DIRECTORY
F-LINK
F-DELETE
F-UNLINK
F-READ-ATTRIB
F-CHANGE-ATTRIB
F-READ-LINK-ATTRIB
F-CHANGE-LINK-ATTRIB
F-MOVE
F-COPY
F-OPEN
F-CLOSE
F-RECOVER
F-LOCATE
F-ERASE
F-DATA-END
F-TRANSFER-END
F-CANCEL

*(insert after Table 45, page 50)*

**Table 45a – Filestore management related diagnostics**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 012 | 7000 | 5 | 5 | Access control inconsistent |
| 012 | 7001 | 5 | 5 | Access control not available |
| 012 | 7002 | 5 | 5 | Path access control inconsistent |
| 012 | 7003 | 5 | 5 | Path access control not available |
| 2 | 7004 | 5 | 5 | Invalid access passwords |
| 2 | 7005 | 5 | 5 | Invalid path access passwords |
| 0 | 7006 | 4,5 | 4,5 | Object name truncated |
| 0 | 7007 | 4,5 | 4,5 | Path member name(s) truncated |
| 2 | 7008 | 5 | 5 | Target file-directory does not exist |
| 2 | 7009 | 5 | 5 | Target object is not a file-directory |
| 2 | 7010 | 5 | 5 | No access to specified file-directory |
| 2 | 7011 | 5 | 5 | No access via specified path |
| 12 | 7013 | 5 | 5 | Ambiguous object specification |
| 0 | 7014 | 5 | 5 | No extension attribute sets are supported |
| 0 | 7015 | 5 | 5 | Extension attribute set(s) not supported |
| 12 | 7016 | 5 | 5 | Attribute value assertion too compex |
| 2 | 7017 | 5 | 5 | Attribute value assertion does not match any objects |
| 2 | 7018 | 5 | 5 | Inconsistent attribute value assertions |
| 2 | 7019 | 5 | 5 | Invalid attribute value assertion |
| 2 | 7020 | 5 | 5 | Extension attribute set member not recognized |

**Table 45b – Generalized selection related diagnostics**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 012 | 8000 | 5 | 5 | Access control not available |
| 012 | 8001 | 5 | 5 | Access control inconsistent |
| 2 | 8002 | 5 | 5 | Invalid access passwords |
| 2 | 8003 | 5 | 5 | Invalid path access passwords |
| 012 | 8004 | 5 | 5 | Path access control not available |
| 012 | 8005 | 5 | 5 | Path access control inconsistent |
| 12 | 8006 | 5 | 5 | Selection attributes not matched |
| 2 | 8007 | 5 | 5 | Attribute value assertion does not match any objects |
| 2 | 8008 | 5 | 5 | Inconsistent attribute value assertions |
| 2 | 8009 | 5 | 5 | Invalid attribute value assertion(s) |
| 12 | 8010 | 5 | 5 | Attribute value assertion too compex |
| 12 | 8011 | 5 | 5 | Maximum set size exceeded |
| 02 | 8012 | 5 | 5 | No extension attribute sets supported |
| 0 | 8013 | 5 | 5 | Extension attribute set(s) not supported |
| 2 | 8014 | 5 | 5 | Extension attribute set member not recognized |
| 2 | 8015 | 5 | 5 | Invalid account |

**Table 45c – Filestore structure management related diagnostics**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 12 | 9000 | 5 | 5 | Target file-directory not found |
| 2 | 9001 | 5 | 5 | Initial attributes not possible |
| 2 | 9002 | 4 | 2 | Bad attribute name |
| 12 | 9003 | 5 | 5 | Object already exists |
| 12 | 9004 | 5 | 5 | Object cannot be created |
| 12 | 9005 | 5 | 5 | Object cannot be deleted |
| 012 | 9006 | 5 | 5 | Access control not available |
| 012 | 9007 | 5 | 5 | Access control inconsistent |
| 012 | 9008 | 5 | 5 | Path access control not available |
| 012 | 9009 | 5 | 5 | Path access control inconsistent |
| 2 | 9010 | 5 | 5 | No access via specified path |
| 0 | 9011 | 5 | 5 | Initial attributes altered |
| 0 | 9012 | 5 | 5 | Override deleted and recreated object with new attributes |
| 12 | 9013 | 5 | 5 | Create override - not possible |
| 2 | 9014 | 5 | 1 | Invalid create password |
| 2 | 9015 | 5 | 1 | Invalid delete password on override |
| 2 | 9016 | 5 | 1 | Bad attribute value |
| 02 | 9018 | 5 | 5 | No extension attributes supported |
| 0 | 9019 | 5 | 5 | Extension attribute set not supported |
| 2 | 9020 | 5 | 5 | Extension attribute set member not recognized |
| 2 | 9021 | 5 | 5 | No pathnames remaining in generalized select group |
| 0 | 9022 | 5 | 5 | Attribute name(s) inconsistent with object type |

*(amend Table 46, page 50)*

**Table 46 – Selection related diagnostics**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 12 | 3000 | 5 | 5 | Named object not found |
| 12 | 3001 | 5 | 5 | Selection attributes not matched |
| 2 | 3002 | 5 | 5 | Initial attributes not possible |
| 2 | 3003 | 4 | 2 | Bad attribute name |
| 12 | 3004 | 5 | 5 | Non-existent object |
| 12 | 3005 | 5 | 5 | Object already exists |
| 12 | 3006 | 5 | 5 | Object cannot be created |
| 12 | 3007 | 5 | 5 | Object cannot be deleted |
| 02 | 3008 | 5 | 5 | Concurrency control not available |
| 02 | 3009 | 5 | 5 | Concurrency control not supported |
| 02 | 3010 | 5 | 5 | Concurrency control not possible |
| 01 | 3011 | 5 | 5 | More restrictive lock |
| 12 | 3012 | 5 | 5 | Object busy |
| 12 | 3013 | 5 | 5 | Object not available |
| 012 | 3015 | 5 | 5 | Access control not available |
| 012 | 3016 | 5 | 5 | Access control inconsistent |
| 0 | 3017 | 4,5 | 4,5 | Object name truncated |
| 0 | 3018 | 5 | 5 | Initial attributes altered |

**Table 46 – Selection related diagnostics (continued)**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 12 | 3019 | 5 | 1 | Bad account |
| 0 | 3020 | 5 | 5 | Override selected existing file |
| 0 | 3021 | 5 | 5 | Override deleted and recreated file with old attributes |
| 0 | 3022 | 5 | 5 | Override deleted and recreated object with new attributes |
| 12 | 3023 | 5 | 5 | Create override - not possible |
| 12 | 3024 | 5 | 5 | Ambiguous object specification |
| 2 | 3025 | 5 | 1 | Invalid create password |
| 2 | 3026 | 5 | 1 | Invalid delete password on override |
| 2 | 3027 | 5 | 1 | Bad attribute value |
| 2 | 3028 | 5 | 1 | Requested access violates permitted actions |
| 2 | 3029 | 5 | 1 | Functional unit not available for requested access |
| 01 | 3030 | 5 | 5 | Object created but not selected |
| 012 | 3031 | 5 | 5 | Path access control not available |
| 012 | 3032 | 5 | 5 | Path access control not supported |
| 012 | 3033 | 5 | 5 | Path access control inconsistent |
| 2 | 3034 | 5 | 5 | Invalid access passwords |
| 2 | 3035 | 5 | 1 | Invalid path access passwords |
| 2 | 3036 | 5 | 5 | No access via specified path |
| 0 | 3037 | 5 | 5 | Concurrency control not supported on object type |
| 0 | 3039 | 5 | 5 | Last pathname in generalized select group |
| 1 | 3040 | 5 | 5 | No more pathnames in generalized select group |
| 0 | 3041 | 4,5 | 4,5 | Path member name(s) truncated |
| 02 | 3042 | 5 | 5 | No extension attribute sets supported |
| 0 | 3043 | 5 | 5 | Extension attribute set(s) not supported |
| 2 | 3044 | 5 | 5 | Extension attribute set member not recognized |
| 2 | 3045 | 5 | 5 | Initial attributes inconsistent with object type |
| 12 | 3047 | 5 | 5 | Ambiguous referent object specification |
| 2 | 3048 | 5 | 5 | Reference created but failed to select referent |
| 2 | 3049 | 5 | 5 | Target referent is a reference object |
| 2 | 3050 | 5 | 5 | File-directory not empty |
| 2 | 3051 | 5 | 5 | Referenced object not deleted |
| 2 | 3052 | 5 | 5 | Reference object not deleted |
| 2 | 3053 | 5 | 5 | Object cannot be unlinked |

*(amend Table 47, page 51)*

**Table 47 – File management related diagnostics**

| Type | Identifier | Observer | Source | Reason |
|---|---|---|---|---|
| 02 | 4009 | 5 | 5 | No extension attributes supported |
| 0 | 4010 | 5 | 5 | Extension attribute set not supported |
| 2 | 4011 | 5 | 5 | Extension attribute set member not recognized |
| 0 | 4012 | 5 | 5 | Attribute name(s) inconsistent with object type |
| 2 | 4013 | 5 | 5 | Object is not a reference |
| 0 | 4014 | 4,5 | 4,5 | Object name truncated |
| 0 | 4015 | 4,5 | 4,5 | Referenced object name truncated |
| 0 | 4016 | 4,5 | 4,5 | Path member name(s) truncated |
| 12 | 4017 | 5 | 5 | Target file-directory not found |
| 2 | 4018 | 5 | 5 | Initial attributes not possible |
| 12 | 4019 | 5 | 5 | Object already exists |
| 12 | 4020 | 5 | 5 | Object cannot be created |
| 12 | 4021 | 5 | 5 | Object cannot be deleted |
| 012 | 4022 | 5 | 5 | Access control not available |
| 012 | 4023 | 5 | 5 | Access control inconsistent |
| 012 | 4024 | 5 | 5 | Path access control not available |
| 012 | 4025 | 5 | 5 | Path access control inconsistent |
| 2 | 4026 | 5 | 5 | No access via specified path |
| 0 | 4027 | 5 | 5 | Initial attributes altered |
| 0 | 4028 | 5 | 5 | Override deleted and recreated object with new attributes |
| 12 | 4029 | 5 | 5 | Create override - not possible |
| 2 | 4030 | 5 | 1 | Invalid create password |
| 2 | 4031 | 5 | 1 | Invalid delete password on override |
| 2 | 4032 | 5 | 1 | Bad attribute value |

# Annex B
(normative)

## Relation of attributes to primitives

*(amend 1st paragraph, page 53)*

Tables 50 through 51 indicate which of the attributes defined in the virtual filestore are affected by the various primitives defined in the file service.

*(insert after 1st paragraph, page 53)*

Table 50 shows the affected object attributes of a given specified object. Table 50a shows the affected attributes of the parent file-directory containing the specified object. Table 50b shows the affected attributes of a linking reference through which the object may have been specified.

Notes on Tables 50 to 50b:

1) Any password values should not be returned.

2) affects attribute of reference object when target object is accessed via reference.

3) Destination file-directory's attribute.

*(amend 2nd paragraph, page 53)*

Primitives which do not occur in the tables 50 through 51 do not affect the virtual filestore attributes directly. However, the deselect, delete, unlink, and group delete primitives may restore a previous value of the current account attribute.

*(amend list item (e), 2nd paragraph, page 53)*

e) set; the attribute value is set on the basis of a value transferred on a parameter of the request and indication primitives, or to a local default if there is no value transferred;

*(amend Table 50, page 53)*

**Table 50 – Object attributes**

| Attribute Name | F-CHANGE-PREFIX | F-LIST | F-GROUP-SELECT |
|---|---|---|---|
| access control | compare | compare & return [1] | compare |
| availability | | compare & return | compare |
| contents type | | compare & return | compare |
| date & time of creation | | compare & return | compare |
| date & time of last attribute modification | | compare & return | compare |
| date & time of last modification | | compare & return | compare |
| date & time of last read access | | compare & return | compare |
| future object size | | compare & return | compare |
| identity of creator | | compare & return | compare |
| identity of last attribute modifier | | compare & return | compare |
| identity of last modifier | | compare & return | compare |
| identity of last reader | | compare & return | compare |
| legal qualifications | | compare & return | compare |
| linked object type | | compare & return | compare |
| linked object | | compare & return | compare |
| object name | | compare & return | compare |
| object size | | compare & return | compare |
| object type | | compare & return | compare |
| path access control | | compare & return [1] | compare |
| permitted actions | | compare & return | compare |
| primary pathname | | compare & return | compare |
| private use | | compare & return | compare |
| storage account | | compare & return | compare |
| child object | | | |

Table 50 – Object attributes (continued)

| Attribute Name | F-GROUP-COPY | F-GROUP-MOVE | F-GROUP-LIST | F-GROUP-CHANGE-ATTRIB | F-SELECT |
|---|---|---|---|---|---|
| access control | compare & set | compare & change | compare & return [1 | compare & change | compare |
| availability | set | change | return | change | |
| contents type | | | return | | |
| date & time of creation | implicit | | return | | |
| date & time of last attribute modification | implicit | implicit | return | implicit | |
| date & time of last modification | implicit | | return | | |
| date & time of last read access | implicit | | return | | |
| future object size | set | change | return | change | |
| identity of creator | set | | return | | |
| identity of last attribute modifier | implicit | implicit | return | implicit | |
| identity of last modifier | implicit | | return | | |
| identity of last reader | implicit | | return | | |
| legal qualifications | set | change | return | change | |
| linked object type | implicit | | return | | |
| linked object | implicit | | return | | |
| object name | compare & set | compare &change | compare &return | compare & change | compare & return |
| object size | implicit | | return | | |
| object type | implicit | | return | | |
| path access control | set | change | return [1] | change | |
| permitted actions | compare | compare | compare & return | compare | compare |
| primary pathname | implicit | implicit | return | change | |
| private use | set | change | return | change | |
| storage account | set | change | return | change | |
| child object | | | | | |

**Table 50 – Object attributes (continued)**

| Attribute Name | F-CREATE | F-CREATE-DIRECTORY | F-LINK | F-DELETE |
|---|---|---|---|---|
| access control | set & return [1] | set & return [1] | compare | compare |
| availability | set & return | set & return | | |
| contents type | set & return | | | |
| date & time of creation | implicit | implicit | | |
| date & time of last attribute modification | implicit | implicit | | |
| date & time of last modification | implicit | implicit | | |
| date & time of last read access | implicit | implicit | | |
| future object size | set & return | set & return | | |
| identity of creator | set & return | set & return | | |
| identity of last attribute modifier | implicit | implicit | | |
| identity of last modifier | implicit | implicit | | |
| identity of last reader | implicit | implicit | | |
| legal qualifications | set & return | set & return | | |
| linked object type | | | | |
| linked object | | | | |
| object name | set & return | set & return | | |
| object size | implicit | implicit | | |
| object type | implicit | implicit | return | |
| path access control | | set & return [1] | | compare |
| permitted actions | set & return | set & return | compare | |
| primary pathname | implicit | implicit | return | |
| private use | set & return | | | |
| storage account | set & return | set & return | | |
| child object | | | | |

**Table 50 – Object attributes (continued)**

| Attribute Name | F-UNLINK | F-READ-ATTRIB | F-READ-LINK-ATTRIB | F-CHANGE-ATTRIB |
|---|---|---|---|---|
| access control | | compare & return [1] | | compare, change, & return [1] |
| availability | | return | | change & return |
| contents type | | return | | |
| date & time of creation | | return | | |
| date & time of last attribute modification | | return | | implicit |
| date & time of last modification | | return | | |
| date & time of last read access | | return | | |
| future object size | | return | | change & return |
| identity of creator | | return | | |
| identity of last attribute modifier | | return | | implicit |
| identity of last modifier | | return | | |
| identity of last reader | | return | | |
| legal qualifications | | return | | change & return |
| linked object type | | | | |
| linked object | | | | |
| object name | | return [2] | | change & return [2] |
| object size | | return | | |
| object type | | return | | |
| path access control | | return [1] | | change & return [1] |
| permitted actions | | return | | |
| primary pathname | | return | | change & return |
| private use | | return | | change & return |
| storage account | | return | | change & return |
| child object | | return | | |

Table 50 – Object attributes (continued)

| Attribute Name | F-CHANGE-LINK-ATTRIB | F-COPY | F-MOVE | F-CLOSE & F-ABORT |
|---|---|---|---|---|
| access control | | compare & set | compare & change | |
| availability | | set | change | |
| contents type | | implicit | | |
| date & time of creation | | implicit | | |
| date & time of last attribute modification | | implicit | | |
| date & time of last modification | | implicit | | implicit |
| date & time of last read access | | implicit | | implicit |
| future object size | | set | change | |
| identity of creator | | implicit | | |
| identity of last attribute modifier | | implicit | | |
| identity of last modifier | | implicit | | implicit |
| identity of last reader | | implicit | | implicit |
| legal qualifications | | set | change | |
| linked object type | | | | |
| linked object | | | | |
| object name | | set | change | |
| object size | | set | | implicit |
| object type | | set | | |
| path access control | | set | change | |
| permitted actions | | set | | |
| primary pathname | | implicit | implicit | |
| private use | | set | change | |
| storage account | | set | change | |
| child object | | | | |

**Table 50a – Parent file-directory attributes**

| Attribute Name | F-CHANGE-PREFIX | F-LIST | F-GROUP-SELECT |
|---|---|---|---|
| access control | compare | compare | compare |
| date & time of creation | | | |
| date & time of last attribute modification | | | |
| date & time of last modification | | | |
| date & time of last read access | | implicit | implicit |
| future object size | | | |
| identity of creator | | | |
| identity of last attribute modifier | | | |
| identity of last modifier | | | |
| identity of last reader | | implicit | implicit |
| legal qualifications | | | |
| object name | | | |
| object size | | | |
| object type | | | |
| path access control | compare | compare | compare |
| permitted actions | | | |
| primary pathname | | | |
| private use | | | |
| storage account | | | |

**Table 50a – Parent file-directory attributes (continued)**

| Attribute Name | F-GROUP-COPY | F-GROUP-MOVE | F-GROUP-LIST | F-GROUP-CHANGE-ATTRIB | F-SELECT |
|---|---|---|---|---|---|
| access control | compare [3] | compare [3] | | | |
| date & time of creation | | | | | |
| date & time of last attribute modification | | | | | |
| date & time of last modification | implicit [3] | implicit [3] | | | |
| date & time of last read access | | | | | |
| future object size | | | | | |
| identity of creator | | | | | |
| identity of last attribute modifier | | | | | |
| identity of last modifier | implicit [3] | implicit, implicit [3] | | | |
| identity of last reader | | | | | |
| legal qualifications | | | | | |
| object name | | | | | |
| object size | implicit [3] | implicit, implicit [3] | | | |
| object type | | | | | |
| path access control | compare, compare [3] | compare, compare [3] | | compare, | |
| permitted actions | | | | | |
| primary pathname | | | | | |
| private use | | | | | |
| storage account | | | | | |

**Table 50a – Parent file-directory attributes (continued)**

| Attribute Name | F-CREATE | F-CREATE-DIRECTORY | F-LINK | F-DELETE |
|---|---|---|---|---|
| access control | compare | compare | compare | compare |
| date & time of creation | | | | |
| date & time of last attribute modification | | | | |
| date & time of last modification | implicit | implicit | implicit | implicit |
| date & time of last read access | | | | |
| future object size | | | | |
| identity of creator | | | | |
| identity of last attribute modifier | | | | |
| identity of last modifier | implicit | implicit | implicit | implicit |
| identity of last reader | | | | |
| legal qualifications | | | | |
| object name | | | | |
| object size | implicit | implicit | implicit | implicit |
| object type | | | | |
| path access control | compare | compare | compare [3] | compare |
| permitted actions | | | | |
| primary pathname | | | | |
| private use | | | | |
| storage account | | | | |