

INTERNATIONAL STANDARD

ISO
8571-3

First edition
1988-10-01



INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

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

Part 3 : File Service Definition

*Systèmes de traitement de l'information — Interconnexion de systèmes ouverts — Gestion,
accès et transfert de fichier —*

Partie 3 : Définition du service de transfert de fichier

Reference number
ISO 8571-3:1988 (E)



Documents
Document Sharing Hub

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8571-3 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

ISO 8571 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*

Annexes A, B, C, D and E form an integral part of this International Standard.

Contents

	Page
0 Introduction	1
1 Scope and field of application	1
2 References	1
3 Definitions	2
4 Abbreviations	2
5 Conventions	2
Section one: General	
6 Model of the file service	3
6.1 File service provider and file service users	3
6.2 File service levels	3
6.3 Regimes of the file service	4
7 Services of the file service	4
7.1 FTAM regime control	4
7.2 Filestore management	4
7.3 File selection regime control	4
7.4 File management	5
7.5 File open regime control	5
7.6 Grouping control	5
7.7 Access to file content	5
7.8 Bulk data transfer	5
7.9 Recovery	5
7.10 Checkpointing and restarting	5
8 Functional units and service classes	5
8.1 Functional units	5
8.2 Service classes	6
8.3 Application Entity roles	8
9 Levels of file service	8
10 Negotiation of service class, FTAM QoS and functional units	9
10.1 Service Class	9
10.2 FTAM Quality of Service	9
10.3 Functional units	9
Section two : Definition of file service primitives	
11 File service primitives	11

12 Sequences of primitives.....	11
12.1 Normal sequences	11
12.2 Constraints on the issue of primitives	11
12.3 Conventions	14
12.4 Confirmed Services.....	20
13 Common file service parameters	20
13.1 State result	20
13.2 Action result	20
13.3 Account	20
13.4 Charging.....	20
13.5 Attributes	20
13.6 Requested access.....	20
13.7 Access Passwords	21
13.8 Concurrency Control	21
13.9 FADU Lock.....	21
13.10 Shared ASE information.....	21
13.11 Activity Identifier	21
13.12 File Access Data Unit Identity	21
13.13 Diagnostic	22
14 FTAM regime control	23
14.1 FTAM regime establishment service.....	23
14.2 FTAM regime termination service (orderly).....	26
14.3 FTAM regime termination service (abrupt).....	26
15 File selection regime control	27
15.1 File selection service.....	27
15.2 File deselection service	28
15.3 File creation service	28
15.4 File deletion service	30
16 File management	31
16.1 Read attribute service	31
16.2 Change attribute service	31
17 File open regime control	31
17.1 File open service	32
17.2 File close service.....	33

18	Grouping control	34
18.1	Beginning of grouping service	34
18.2	End of grouping service	35
19	Recovery (Internal service only)	35
19.1	Regime recovery service	35
20	Access to file contents	36
20.1	Bulk data transfer service	36
20.2	Locate file access data unit service	36
20.3	Erase file access data unit service	37
Section three: Definition of bulk data transfer primitives		
21	Bulk data transfer service primitives	38
22	Sequences of bulk data transfer primitives	38
22.1	Normal sequences	38
22.2	Constraints on issue of primitives	38
23	Common bulk data transfer parameters	44
23.1	Bulk Data Transfer Specification	44
23.2	Checkpoint Identifier	44
24	Bulk data transfer	44
24.1	Read bulk data service	44
24.2	Write bulk data service	44
24.3	Data unit transfer service	44
24.4	End of data transfer service	45
24.5	End of transfer service	45
24.6	Cancel data transfer service	45
24.7	Sequence of primitives on write	46
24.8	Sequence of primitives on read	46
25	Checkpointing and restart (Internal BDT Service Only)	46
25.1	Checkpointing service	46
25.2	Restarting data transfer service	47

Annexes

A	Diagnostic parameter values	48
B	Relation of attributes to primitives	53
C	File transfer with commitment control	55
D	Reference to FTAM control information	58
E	State transition diagrams	59

Figures

1	Service Levels	3
2	File service regimes and related primitives	4
3	Simplified State Diagram for successful activity (see Annex E)	13
4	Confirmed service	20
5	F-U-ABORT service	27
6	F-P-ABORT service	27
7	F-P-ABORT collision	27
8	Simplified State Diagram for Bulk Data Transfer (see Annex E)	39
9	Sequence of primitives on write	46
10	Sequence of primitives on read	47
11	State Transition Diagram for Association Establishment (Initiator)	60
12	State Transition Diagram for Association Establishment (Responder)	61
13	State Transition Diagram of the File Regime Establishment Service (Initiator)	62
14	State Transition Diagram for Grouped Sequences (Initiator)	63
15	State Transition Diagram of the File Regime Establishment Service (Responder)	64
16	State Transition Diagram for Grouped Sequences (Responder)	65
17	State Transition Diagram for the Bulk Data Transfer Service (Initiator)	66
18	State Transition Diagram for the Bulk Data Transfer Service (Responder)	67

Tables

1	Services and functional units of the External File Service	7
2	Services and functional units of the Internal File Service	7
3	Functional units in the file services	9
4	Service Class Combinations	9
5	Service Class Negotiation	10
6	File service primitives	11
7	Sequence of service primitives for FTAM regime establishment — initiator	15
8	Sequence of service primitives for FTAM regime establishment — responder	15
9	Sequence of service primitives for file service regimes — initiator	16
10	Sequence of service primitives for file service regimes — responder	18
11	F-INITIALIZE parameters	24
12	F-TERMINATE parameters	26
13	F-U-ABORT parameters	26
14	F-P-ABORT parameters	26
15	F-SELECT parameters	27
16	F-DESELECT parameters	28
17	F-CREATE parameters	29
18	F-DELETE parameters	30
19	F-READ-ATTRIB parameters	31
20	F-CHANGE-ATTRIB parameters	32
21	F-OPEN parameters	32
22	F-CLOSE	34
23	F-BEGIN-GROUP parameters	35
24	F-RECOVER parameters	35
25	BDT read sub-parameters	36
26	BDT write sub-parameters	36
27	Access contexts	36
28	F-LOCATE parameters	37
29	F-ERASE parameters	37
30	Bulk data transfer service primitives	38
31	Sequence of service primitives for bulk data transfer — initiator	40
32	Sequence of service primitives for bulk data transfer — responder	42
33	F-READ parameters	44

34 F-WRITE parameters.....	44
35 F-DATA parameters.....	45
36 F-DATA-END parameters.....	45
37 F-TRANSFER-END parameters.....	45
38 F-CANCEL parameters.....	46
39 F-CHECK parameters.....	47
40 F-RESTART parameters.....	47
41 Error types.....	48
42 Sources and observers of errors.....	48
43 General FTAM diagnostics.....	49
44 Protocol and supporting service related diagnostics.....	49
45 Association related diagnostics.....	50
46 Selection related diagnostics.....	50
47 File management related diagnostics.....	51
48 Access related diagnostics.....	51
49 Recovery related diagnostics.....	52
50 File attributes.....	53
51 Activity attributes.....	54
52 File service primitives associated with CCR primitives.....	56
53 FTAM Primitives with Shared ASE Information parameters.....	56
54 Composite FTAM actions.....	57

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

Part 3 : File Service Definition

0 Introduction

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

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

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

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

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

It is recognised that, with respect to Communication Quality of Service, (described in 14.1.2.16), work is still in progress to provide an integrated treatment of quality of service across all of the layers of the OSI Reference Model and to ensure that the individual treatments in each layer service satisfy overall quality of service objectives in a consistent manner. As a consequence, an addendum may be added to this International Standard at a later time which reflects further quality of service developments and integration.

ISO 8571 consists of the following four parts.

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

This part of ISO 8571 contains the following annexes which form part of the standard.

- Annex A - Diagnostic parameter values
- Annex B - Relation of attributes to primitives
- Annex C - File transfer with commitment control
- Annex D - Reference to FTAM control information
- Annex E - State transition diagrams

1 Scope and field of application

This part of ISO 8571 defines in an abstract way the externally visible file transfer, access and management service within the OSI Application Layer in terms of:

- a) the primitive actions and events of the service;
- b) the parameter data associated with each primitive action and event;
- c) the relationship between, and the valid sequences of, these actions and events.

The service defined in ISO 8571-3 is that which is provided by the OSI file transfer, access and management protocol ISO 8571-4 in conjunction with the Association Control Service Elements ISO 8649 and with the Presentation service ISO 8822.

ISO 8571-3 does not specify individual implementations or products, nor does it constrain the implementation of entities and interfaces within a computer system. There is, therefore, no conformance to this part of ISO 8571.

2 References

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

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

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

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

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

ISO 8822, *Information Processing Systems - Open Systems Interconnection - Connection-oriented presentation service definition*.

ISO 8831, *Information Processing Systems -Open Systems Interconnection - Job Transfer and Manipulation Concepts and Services.*

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

ISO 9805, *Information Processing Systems -Open Systems Interconnection - Specification of protocols for Application Service Elements - Commitment, Concurrency and Recovery.*¹⁾

3 Definitions

Terms used in ISO 8571-3 are defined in ISO 8571-1.

4 Abbreviations

Abbreviations used in ISO 8571-3 are defined in ISO 8571-1.

5 Conventions

ISO 8571-3 uses the descriptive conventions in the OSI Service Conventions in ISO/TR 8509.

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

Section one: General

6 Model of the file service

6.1 File service provider and file service users

ISO 8571-3 uses the abstract model for a service defined in the OSI service conventions in ISO TR 8509 (see note 1). The model defines the interactions between the two file service users and the file service provider. Information is passed between a file service user and the file service provider by file service primitives which may carry parameters.

One of the file service users is defined as the initiator and the other is defined as the responder (see ISO 8571-1).

The responder is an application-entity handling a virtual filestore. The virtual filestore has the properties defined in ISO 8571-2, and may be realized in a real system by a real filestore or by an application process. Attributes of the virtual filestore manipulated by the service primitives defined in ISO 8571-3 are listed in annex B.

The file service defines a single activity between an initiator and a responder (see note 2).

NOTES

1 ISO/TR 8509 defines a model for the service provided by a layer of the OSI Reference Model. The file service does not correspond to such a layer (it is a division within the application layer) but the model used is identical in all other respects.

2 At any one time, an application entity may be involved in more than one instance of the file service activity and each instance is based on a separate application association.

6.2 File service levels

Two levels of file service are defined:

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 is modelled in the external file service as a series of error-free operations. Thus within the external file service there is no visibility of recoverable errors or the error recovery actions;

b) the internal file service (IFS) used by the error recovery protocol machine. This service includes primitives giving its users facilities for error recovery and control of the checkpointing mechanisms. The protocol specification which relates the external to the internal file service therefore consists of a standard set of procedures for error recovery and the protocol machine which executes these procedures is the user of the internal file service. The choice of the error recovery procedures to be used is based on the analysis of cost from the FTAM and communications quality of service requested in the External Service and local management information.

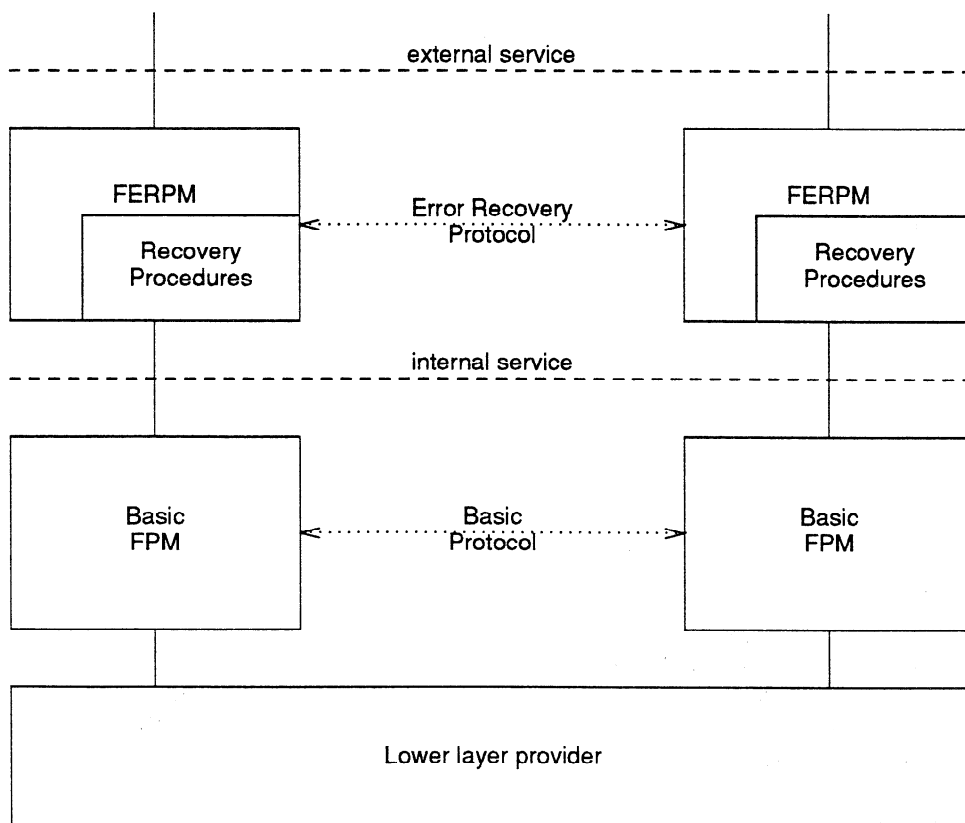


Figure 1 — Service Levels

The relationship between the internal and external file service services is shown diagrammatically in figure 1.

6.3 Regimes of the file service

Four types of file service regime are defined:

- the FTAM regime which exists while the application association is used for the FTAM protocol.
- the file selection regime during which a particular file is associated with the FTAM regime;
- the file open regime during which a particular set of processing mode, presentation contexts and concurrency controls is in operation;
- the data transfer regime during which a particular bulk data transfer specification and direction of transfer are in force.

There is at most one instance of each type of regime at any one time.

The file service provides for:

- a sequence of file selection regimes in an FTAM regime;
- a sequence of file open regimes in a file selection regime;
- 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.

Termination of a regime implies termination of all regimes nested within that regime. The nesting of regimes is shown in figure 2.

7 Services of the file service

This clause provides a short description of the services of the file service. The services and the primitives by which

they are invoked are defined in sections two and three. For each service, the user of the service (the application entity that invokes the sequence of primitives) is stated. The external and internal file service levels are defined in 6.2.

7.1 FTAM regime control

Three services are associated with FTAM regime control:

- the FTAM regime establishment service (see 14.1) is used by the initiator to create and bind an FTAM regime to the application association linking the two file service users;
- the FTAM regime termination service (orderly) (see 14.2) is used by the initiator to dissolve the FTAM regime and unbind it from the application association between the file service users and the file service provider;
- the FTAM regime termination service (abrupt) (see 14.3) is used by either of the service users or the service provider to dissolve the FTAM regime and its binding to the application association unconditionally.

7.2 Filestore management

ISO 8571-3 does not define any filestore management operations.

NOTE - Such operations may be included in future addenda to ISO 8571-3.

7.3 File selection regime control

Four services are associated with file selection regime control:

- the file selection service (see 15.1) is used by the initiator to select an existing file and to bind the specified file to the FTAM regime;
- the file deselection service (see 15.2) is used by the initiator to release the binding between the FTAM regime and the specified file;

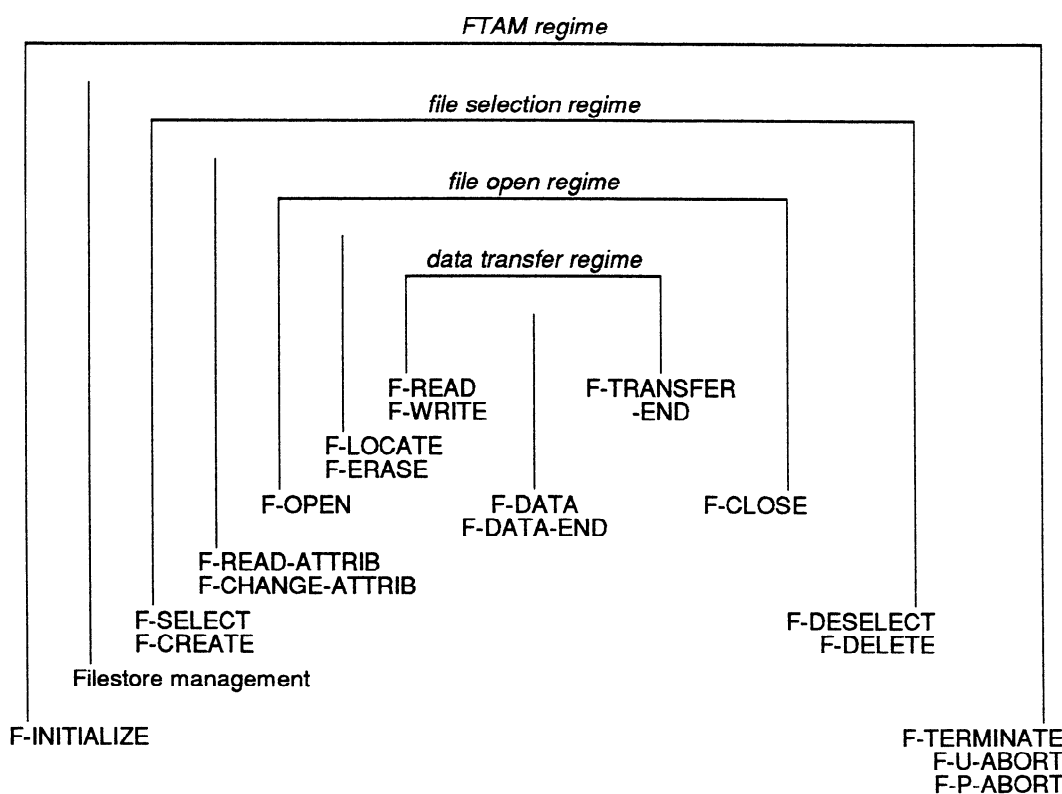


Figure 2 — File service regimes and related primitives

c) the file creation service (see 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;

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

7.4 File management

Two services are associated with file management:

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

b) the change attributes service (see 16.2) is used by the initiator to modify the file attributes of the selected file.

7.5 File open regime control

Two services are associated with file open regime control:

a) the file open service (see 17.1) is used by the initiator to establish the processing mode, presentation contexts and concurrency controls for data transfer or access;

b) the file close service (see 17.2) is used by the initiator to release the context established by the file open service.

7.6 Grouping control

Two services are associated with grouping control:

a) the beginning of grouping service (see 18.1) is used by the initiator to indicate the start of a set of primitives which are to be processed and responded to as a group;

b) the end of grouping service (see 18.2) is used by the initiator to indicate the end of a set of grouped primitives which are to be processed and responded to as a group.

7.7 Access to file content

The transfer of the FADUs of a file is performed by a bulk data transfer procedure, which forms a self contained unit. The services which make up this procedural unit are described in 7.9 and 7.10. There are two additional services associated with file access:

a) the locate file access data unit service (see 20.2) is used by the initiator to specify the identity of a file access data unit which is to be located by the responder;

b) the erase file access data unit service (see 20.3) is used by the initiator to remove a file access data unit from the file.

7.8 Bulk data transfer

Bulk data transfer refers to the transfer, optionally with checkpointing, of single file access data units (see 20.1). There are six additional services associated with different stages in bulk data transfer:

a) the read bulk data service (see 24.1) is used by the initiator to initiate a bulk data transfer from the responder

(in the role of sender), to the initiator (in the role of receiver);

b) the write bulk data service (see 24.2) is used by the initiator to initiate a bulk data transfer from the initiator (in the role of sender) to the responder (in the role of receiver);

c) the data unit transfer service (see 24.3) is used by the sender to transmit bulk data;

d) the end of data transfer service (see 24.4) is used by the sender to indicate completion of the data transfer;

e) the end of transfer service (see 24.5) is used by the initiator to confirm that the data transfer is complete;

f) the cancel data transfer service (see 24.6) is used by either the sender or the receiver to cancel a data transfer activity.

7.9 Recovery

One service is associated with recovery. This service is visible only in the internal file service.

The regime recovery service (see 19.1) is used by the initiator to recreate the open regime after a failure within the open regime. Errors occurring outside the open regime are not recoverable by this service.

NOTE - Concurrency controls remain in force during a recovery attempt, or if so specified by another ASE. Otherwise, concurrency controls are released on a permanent error (see 13.2).

7.10 Checkpointing and restarting

Two services are associated with checkpointing and restarting. These services are visible only in the internal file service:

a) the checkpointing service (see 25.1) is used by the sender of data to establish marks in the flow of data for the purpose of subsequent recovery or restart;

b) the restarting data transfer service (see 25.2) is used by the sender or the receiver of data to interrupt a transfer in progress and negotiate a point at which it is to be restarted.

8 Functional units and service classes

Functional units and file service classes are logical groupings of related services defined in ISO 8571-3 for the purpose of:

a) negotiation of the file service user's requirements during FTAM regime establishment;

b) reference by other International Standards.

NOTE - The constraint set applicable to the file restricts the functionality given in 8.1 and 8.2.

8.1 Functional units

The services associated with each functional unit are specified in tables 1 and 2.

8.1.1 Kernel functional unit

The kernel functional unit supports the basic file services for the establishment and release of the FTAM and file selection regimes.

8.1.2 Read functional unit

The read functional unit supports establishment and release of the open regime and the transfer of data from the responder to the initiator.

8.1.3 Write functional unit

The write functional unit supports establishment and release of the open regime and the transfer of data from the initiator to the responder.

8.1.4 File access functional unit

The file access functional unit allows an FADU in the file access structure to be located for file access and allows an FADU to be manipulated within the file access structure.

8.1.5 Limited file management functional unit

The limited file management functional unit supports file management for the creation and deletion of files and for the interrogation of file attributes.

8.1.6 Enhanced file management functional unit

The enhanced file management functional unit extends the capabilities of the limited file management functional unit to include the modification of file attributes.

8.1.7 Grouping functional unit

The grouping functional unit allows several regimes to be established in one exchange, by combining several independent primitives into a group for procedural purposes.

8.1.8 FADU Locking functional unit

The FADU locking functional unit allows the invocation of concurrency control locks on a per FADU basis in addition to a file basis.

8.1.9 Recovery functional unit

The recovery functional unit allows the initiator to recreate an open regime which has been destroyed by some failure. The recovery may be immediate or deferred on the existing, or different a, association.

8.1.10 Restart data transfer functional unit

The restart data transfer functional unit allows a data transfer to be interrupted and restarted immediately at a negotiated point within the current transfer.

8.1.11 Service classes and functional units

Table 1 shows which of the functional units are mandatory and which optional in each of the service classes: transfer, access, management, transfer and management, and unconstrained. Service classes are defined in 8.2.

The recovery and restart functional units are never explicitly visible in the external file service. Where they are shown as optional in table 2 it is in the internal file service that they are optional. The external file service and the internal file service are defined in 6.2.

8.2 Service classes

Five file service classes are defined in terms of combinations of functional units:

- a) file transfer class (see 8.2.1);

- b) file access class (see 8.2.2);
- c) file management class (see 8.2.3);
- d) file transfer and management class (see 8.2.4);
- e) unconstrained class (see 8.2.5).

8.2.1 File transfer class

The file transfer 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) optionally, the limited file management functional unit;
- e) optionally, but only if the limited file management functional unit is present, the enhanced file management functional unit;
- f) optionally, in the internal file service, the recovery functional unit.
- g) optionally, in the internal file service, the restart data transfer functional unit.

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 is a sequence of:

- 1) a single grouped sequence to establish a file open regime. This sequence consists of:
 - an F-BEGIN-GROUP primitive;
 - an F-SELECT or F-CREATE primitive;
 - optionally, an F-READ-ATTRIB primitive;
 - optionally, an F-CHANGE-ATTRIB primitive;
 - an F-OPEN primitive;
 - an F-END-GROUP primitive.
- 2) 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.
- 3) a single grouped sequence to release the file open and select regimes. This sequence consists of:
 - an F-BEGIN-GROUP primitive;
 - an F-CLOSE primitive;
 - optionally, an F-READ-ATTRIB primitive;
 - optionally, an F-CHANGE-ATTRIB primitive;
 - an F-DESELECT or F-DELETE primitive;
 - an F-END-GROUP primitive.

The threshold parameter is set equal to the number of primitives between F-BEGIN-GROUP and F-END-GROUP. The threshold parameter is defined in clause 18.

NOTE - Each of the primitives shown as optional in these sequences may only be present if the corresponding functional unit was negotiated during the FTAM regime establishment.

8.2.2 File access class

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 its acceptance by the responder is always mandatory.

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

Functional Unit	Services	Service classes					Clause
		T	A	M	TM	U	
U1 Kernel	FTAM regime establishment						14.1
	FTAM regime termination (orderly)						14.2
	FTAM regime termination (abrupt)	M	M	M	M	M	14.3
	File selection						15.1
	File deselection						15.2
U2 Read	Read bulk data						24.1
	Data unit transfer	*	M		*	O	24.3
	End of data transfer						24.4
	End of transfer						24.5
	Cancel data transfer						24.6
	File open						17.1
	File close						17.2
U3 Write	Write bulk data						24.2
	Data unit transfer	*	M		*	O	24.3
	End of data transfer						24.4
	End of transfer						24.5
	Cancel data transfer						24.6
	File open						17.1
	File close						17.2
U4 File access	Locate FADU		M			O	20.2
	Erase FADU (requires U2 or U3)						20.3
U5 Limited file management	File creation						15.3
	File deletion	O	O	M	M	O	15.4
	Read attributes						16.1
U6 Enhanced file management	Change attributes (requires U5)	O	O	O	O	O	16.2
U7 Grouping	Beginning of grouping	M	O	M	M	O	18.1
	End of grouping						18.2
U8 FADU locking	FADU Locking (requires U2 or U3) and U4		O			O	

Table 2 — Services and functional units of the Internal File Service

Functional Unit	Services	Service classes					Clause
		T	A	M	TM	U	
U9 Recovery	Regime Recovery						19.1
	Checkpointing	O	O		O	O	25.1
	Cancel data transfer (for recoverable errors)						24.6
U10 Restart data transfer	Restarting data transfer						25.2
	Checkpointing	O	O		O	O	25.1
	Cancel data transfer						24.6
	(for recoverable errors)						

Key to tables 1 and 2

Service class abbreviations

T = File transfer class
 A = File access class
 M = File management class
 TM = File transfer and management class
 U = Unconstrained class

Abbreviations within service class

M = Mandatory
 O = Optional
 * = At least one of U2 or U3
 blank = not permitted

- 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;
- g) optionally, the FADU locking functional unit;
- h) optionally, in the internal file service, the recovery functional unit.
- i) optionally, in the internal file service, the restart data transfer functional unit.

NOTES

- 1 The threshold constraints applicable to the file transfer class do not apply to the file access class.
- 2 The constraints on grouping and sequence of events defined in 8.2.1 do not apply to the file access service class.

8.2.3 File management class

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.

In the file management 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 is a sequence of:

- an F-BEGIN-GROUP primitive;
- an F-SELECT or F-CREATE primitive;
- optionally, an F-READ-ATTRIB primitive;
- optionally, an F-CHANGE-ATTRIB primitive;
- an F-DESELECT or F-DELETE primitive;
- an F-END-GROUP primitive.

The threshold parameter, defined in clause 18, is set equal to the number of primitives between F-BEGIN-GROUP and F-END-GROUP.

8.2.4 File transfer and management class

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;
- f) optionally, in the internal file service, the recovery functional unit;
- g) optionally, in the internal file service, the restart data transfer functional unit.

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

a transfer comprising:

- 1) a single grouped sequence to establish a file open regime. This sequence consists of:
 - an F-BEGIN-GROUP primitive;
 - an F-SELECT or F-CREATE primitive;
 - optionally, an F-READ-ATTRIB primitive;

- optionally, an F-CHANGE-ATTRIB primitive;
- an F-OPEN primitive;
- an F-END-GROUP primitive.

2) 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 read or a valid write action, as defined in the constraint set, but not both.

3) a single grouped sequence to release the file open and select regimes. This sequence consists of:

- an F-BEGIN-GROUP primitive;
- an F-CLOSE primitive;
- optionally, an F-READ-ATTRIB primitive;
- optionally, an F-CHANGE-ATTRIB primitive;
- an F-DESELECT or F-DELETE primitive;
- an F-END-GROUP primitive.

or a single grouped sequence to effect file management consisting of:

- an F-BEGIN-GROUP primitive;
- an F-SELECT or F-CREATE primitive;
- optionally, an F-READ-ATTRIB primitive;
- optionally, an F-CHANGE-ATTRIB primitive;
- an F-DESELECT or F-DELETE primitive;
- an F-END-GROUP primitive.

The threshold parameter, defined in clause 18, is set equal to the number of primitives between F-BEGIN and F-END-GROUP.

NOTE - The primitives in the functional units may only be present in these sequences if the corresponding functional units were negotiated during the FTAM regime establishment.

8.2.5 Unconstrained class

The unconstrained class consists of :

- a) the kernel functional unit;
- b) optionally, any other functional units.

NOTES

- 1 The threshold parameter constraints do not apply to the unconstrained class.
- 2 The unconstrained service class is provided for definition of non standard applications. No restrictions are imposed by ISO 8571 on this class.

8.3 Application Entity roles

An application entity keeps the same role (either initiator or responder) for the duration of the FTAM regime. This specific role applies to each of the available functional units.

9 Levels of file service

Two levels of file service are defined in 6.2. They are:

- a) the external file service (EFS), in which the user states its FTAM quality of service requirements, but has no awareness of error recovery.
- b) the internal file service (IFS) used by the error recovery protocol machine. This service includes primitives giving its users facilities for error recovery and control of the checkpointing mechanisms.

The functional units providing visible services in the external file service and in the internal file service are defined in table 3.

The external file service is supported by the error recovery protocol, which may be null, and indirectly by the basic protocol specified in ISO 8571-4. The internal file service is supported by the basic protocol.

Selection of a named functional unit in the external service implies selection of the same named functional unit in the internal file service. In the internal file service the recovery and restart functional units are optional. The primitives of the restart and recovery functional units will not be visible to the external file service user.

Table 3 — Functional units in the file services

External file service	Internal file service
Kernel	Kernel
Read	Read
Write	Write
File access	File access
Limited file management	Limited file management
Enhanced file management	Enhanced file management
Grouping	Grouping
FADU locking	FADU locking
	Recovery
	Restart data transfer

10 Negotiation of service class, FTAM QoS and functional units

Service class and FTAM quality of service are negotiated independently on the F-INITIALIZE exchange. Each of these negotiations contributes to the final selection of functional units available on the association.

The kernel functional unit is always available and is not included in the negotiations.

The availability of the functional units, read, write, file access, limited file management, enhanced file management, grouping and FADU locking is controlled by the service class negotiated (see Tables 1 and 2).

The proposal of either, or both, of the restart or recovery functional units is as a result of a local decision based on the underlying service and the FTAM quality of service negotiated between the two file service users.

10.1 Service Class

The initiator states in the service class parameter of the F-INITIALIZE request primitive the service classes required. The valid combinations are defined in table 4. Each of the combinations listed may have included with it the unconstrained class.

Table 4 — Service Class Combinations

Abbreviation	Service Class Capability
T	Transfer class
M	Management class
A	Access class
T,A	Transfer class, Access class
T,M,TM	Transfer class, Management class, Transfer and Management class
A,T,M,TM	Access class, Transfer class, Management class or Transfer and Management class

During negotiation the service provider removes any service classes it is unable to support and signals the remaining set to the responder on the F-INITIALIZE indication.

The responder removes from the service class list all classes it is unable to support. It then selects from the remaining classes the highest service class it is able to support. It returns this on the service class parameter of the F-INITIALIZE response primitive. Support of both the transfer class and the management class requires support of the transfer and management class.

The order of service classes is defined as, highest to lowest; access, transfer and management, transfer, management, unconstrained.

The agreed service class is signalled to the initiator on the F-INITIALIZE confirm primitive. This negotiation of service classes always results in a single service class being agreed. Table 5 details the outcome of the negotiation.

The functional unit constraints of each service class are defined in tables 1 and 2. The functional unit negotiation is defined in 10.3.

10.2 FTAM Quality of Service

The initiator states on the F-INITIALIZE request primitive the FTAM quality of service indicating the error classes the application is susceptible to.

The service provider adds to the functional units included in the functional units parameter neither, one or both of the recovery and restart functional units, according to the file service user's requested FTAM quality of service, local knowledge and which of the error recovery and restart functional units it is able to support. The service provider indicates to the responder, on the FTAM quality of service parameter of the F-INITIALIZE indication primitive, the value requested by the initiator and possibly reduced by the service provider.

The responding file protocol machine uses the initiator's FTAM quality of service, local knowledge and capability to determine whether it is able to support the requested FTAM quality of service. If the responder is able to provide the requested FTAM quality of service it returns the parameter unchanged. The recovery and/or restart functional units are removed from the set of functional units parameter if the responding file protocol machine is unable to support them.

If the responder is unable to support the requested FTAM quality of service, it returns on the F-INITIALIZE response primitive the value it is able to support. The service provider may reduce the value of the FTAM quality of service if the resulting functional units do not provide the capability required. The final value of the FTAM quality of service is signalled to the file service user on the F-INITIALIZE confirm primitive. The negotiation of this parameter does not, in itself, prevent the establishment of an FTAM regime. If the quality of service is below that acceptable to the file service user the regime may be terminated by the issue of an F-TERMINATE request primitive by the file service user.

10.3 Functional units

The initiator states on the F-INITIALIZE request primitive its requirement in terms of functional units. The complete list proposed comprises all the mandatory functional units, except the kernel, in the service classes proposed in the service class parameter plus additional optional functional

units. The recovery and restart functional units are only inserted by the service provider.

The service provider removes from this set any functional units it is unable to support and signals the remaining set to the responder on the F-INITIALIZE indication primitive. The responder removes from this set any functional units it is

unable to support and signals the remaining set to the service provider on the F-INITIALIZE response primitive.

The service provider signals the same set to the initiator on the F-INITIALIZE confirm primitive. This negotiated set of functional units is then available for use in the established FTAM regime.

Table 5 — Service Class Negotiation

Initiator capability	Responder capability					
	T	M	A	T,A	T,M,TM	A,T,M,TM
T	T	—	—	T	T	T
M	—	M	—	—	M	M
A	—	—	A	A	—	A
T,A	T	—	A	A	T	A
T,M,TM	T	M	—	T	TM	TM
A,T,M,TM	T	M	A	A	TM	A

Key to Table 5

In table 5 the symbols have the meaning defined in table 4. Additionally the — symbol has the following meaning:

- if both initiator and responder capabilities included the unconstrained service class then that class is the result of the negotiation.
- if either or both lack the unconstrained class capability then the responder rejects the FTAM regime establishment attempt with a state result parameter indicating failure and, optionally, an appropriate diagnostic value.

Section two : Definition of file service primitives

11 File service primitives

Each of the services constituting the file service is achieved by invoking a sequence of file service primitives.

Table 6 lists for each service;

- a) the primitives associated with the service;
- b) the parameters associated with the primitives;
- c) the file service user that is permitted to issue the request primitive;

In table 6, parameters and primitives which are only visible in the internal service are enclosed in square brackets.

The semantics of the primitives and their parameters are defined in clauses 13 to 20.

The primitives of the bulk data transfer service are listed in clause 21 and defined in clauses 23 to 25.

12 Sequences of primitives

This clause defines the constraints on the permissible sequences for the EFS in which the primitives defined in clauses 14 to 20 can occur. The individual primitives in a service may occur only in the sequences given as part of the primitive definitions. The F-RECOVER, F-RESTART and F-CHECK service primitives are not issued in the external file service.

12.1 Normal sequences

The normal progress of use of the file service is illustrated by the state transition diagram shown in figure 3, applicable separately to each application entity. Full state transition diagrams are included in annex E.

12.2 Constraints on the issue of primitives

The primitives may be issued in any sequence consistent with the constraints given in tables 7 to 10. The sequences

Table 6 — File service primitives

Primitive	Confirmed	Request by	Parameters
F-INITIALIZE	Yes	Initiator	State Result Action Result Called application title Calling application title Responding application title Called presentation address Calling presentation address Responding presentation address Presentation context management Application context name Service class Functional units Attribute groups Shared ASE information FTAM QoS Communication QoS Contents type list Initiator Identity Account Filestore password Diagnostic [Checkpoint window]
F-TERMINATE	Yes	Initiator	Shared ASE information Charging
F-U-ABORT	No	Either	Action Result Diagnostic
F-P-ABORT	No	Provider	Action result Diagnostic
F-SELECT	Yes	Initiator	State Result Action Result Attributes Requested Access Access passwords Concurrency control Shared ASE information Account Diagnostic

Table 6 (continued) — File service primitives

Primitive	Confirmed	Request by	Parameters
F-DESELECT	Yes	Initiator	Action Result Charging Shared ASE information Diagnostic
F-CREATE	Yes	Initiator	State Result Action Result Override Initial Attributes Create password Requested Access Access passwords Concurrency control Shared ASE information Account Diagnostic
F-DELETE	Yes	Initiator	Action Result Shared ASE information Charging Diagnostic
F-READ-ATTRIB	Yes	Initiator	Action Result Attribute names Attributes Diagnostic
F-CHANGE-ATTRIB	Yes	Initiator	Action Result Attributes Diagnostic
F-OPEN	Yes	Initiator	State Result Action result Processing mode Contents type Concurrency control Shared ASE information Enable FADU Locking Diagnostic [Activity identifier] [Recovery mode]
F-CLOSE	Yes	Initiator	Action Result Shared ASE information Diagnostic
F-BEGIN-GROUP	Yes	Initiator	Threshold
F-END-GROUP	Yes	Initiator	-
[F-RECOVER]	Yes	Initiator	[State Result Action result Activity identifier Bulk transfer number Requested Access Access passwords Contents type Recovery point Diagnostic]
F-LOCATE	Yes	Initiator	Action Result FADU identity FADU Lock Diagnostic
F-ERASE	Yes	Initiator	Action Result FADU identity Diagnostic

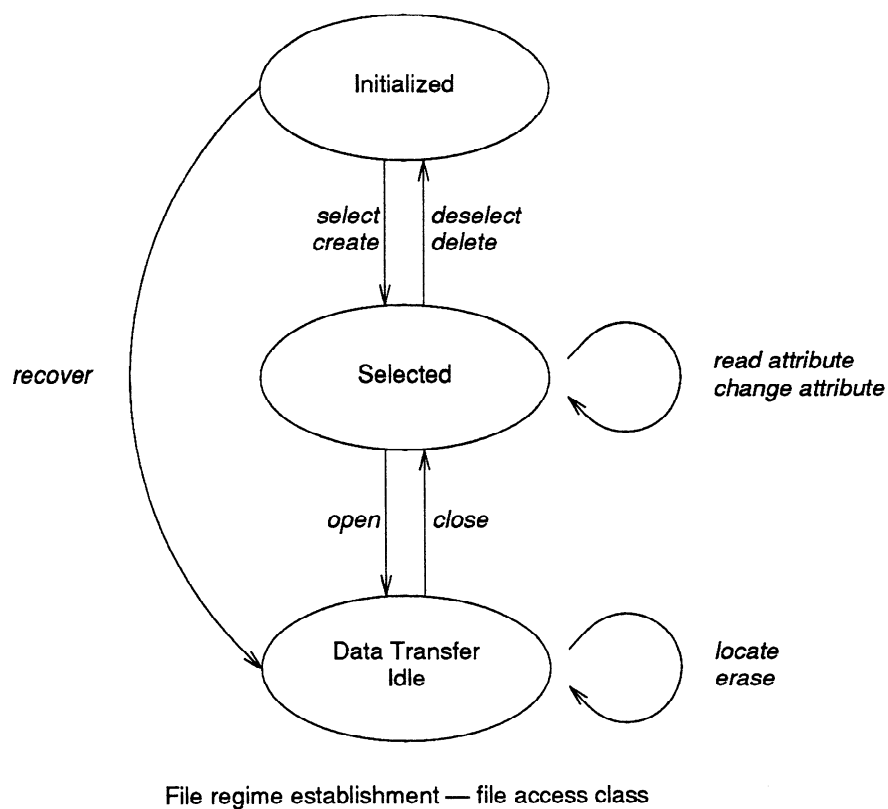
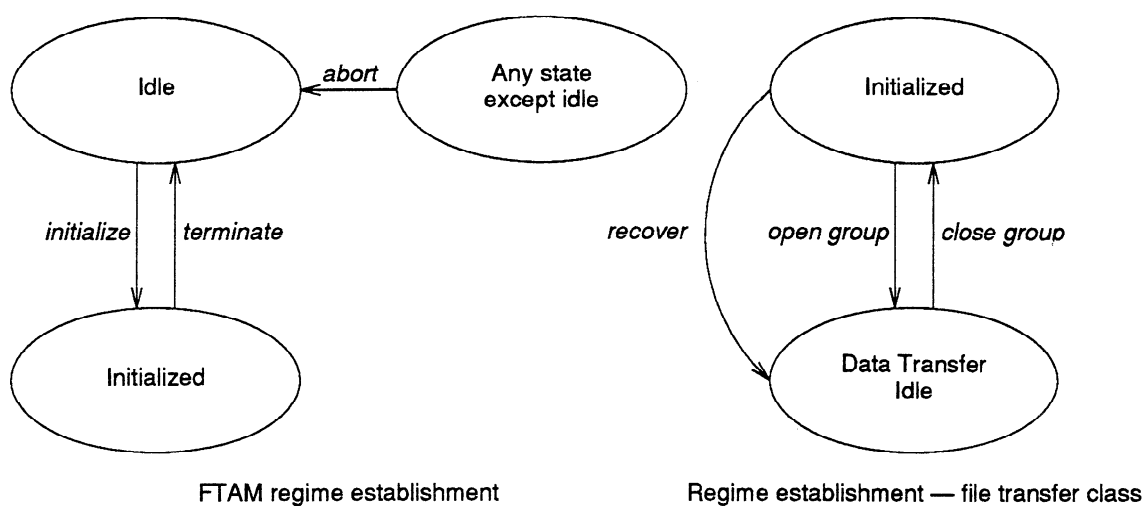


Figure 3 — Simplified State Diagram for successful activity (see Annex E)

of primitives are defined under the individual services. Individual primitive sequences may be interleaved to form the following grouped sequences, which are expressed using the notation defined in annex E.

- a) F-BEGIN-GROUP
(F-SELECT[F-CREATE]
[F-READ-ATTRIB] [F-CHANGE-ATTRIB]
F-OPEN
F-END-GROUP
- b) F-BEGIN-GROUP
F-CLOSE
[F-READ-ATTRIB] [F-CHANGE-ATTRIB]
(F-DESELECT[F-DELETE]
F-END-GROUP
- c) F-BEGIN-GROUP
(F-SELECT[F-CREATE]
[F-READ-ATTRIB] [F-CHANGE-ATTRIB]
(F-DESELECT[F-DELETE]
F-END-GROUP
- d) F-BEGIN-GROUP
(F-SELECT[F-CREATE]
[F-READ-ATTRIB] [F-CHANGE-ATTRIB]
F-END-GROUP
- e) F-BEGIN-GROUP
[F-READ-ATTRIB] [F-CHANGE-ATTRIB]
(F-DESELECT[F-DELETE]
F-END-GROUP

NOTES

1 Not all the above sequences are allowed in the file transfer, file management and file transfer and management classes (see 8.2.1, 8.2.3, 8.2.4). In the file transfer class, only sequences (a) and (b) are valid, and 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. In the file management class only sequence (c) is valid. (see 8.2.3). In the file transfer and management class only the sequences (a), (b) and (c) are valid.

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 Conventions

The following conventions apply to entries in tables 7 to 10.

12.3.1 Conventions for tables 7 and 8

In tables 7 and 8, the entry "yes" implies that the succession given can occur. The column "start of file regimes" indicates entry to tables 9 and 10. The row "end of file regimes" indicates exit from tables 9 and 10, and the row "within file regimes" indicates any other entry in tables 9 and 10.

12.3.2 Conventions for tables 9 and 10

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 file management functional unit
Emgt	enhanced file 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 file management functional units
Lmgt-Rec	limited file management and recovery functional units
Rec-Acc	recovery and file access functional units
G-Rec	grouping and recovery functional units
G-Acc	grouping and file access functional units

The row "start file regimes" indicates entry from tables 7 and 8, and the column "end file regimes" indicates return to tables 7 and 8. The row and column "bulk data transfer" indicate use of the bulk data transfer service defined in section three.

In tables 7 to 10, the value of the state result parameter is indicated by:

- (+ve) positive response or confirm; the primitive carries a state result indicating a successful transition;
- (-ve) negative response or confirm; the primitive carries a state result indicating an unsuccessful transition or regime establishment attempt.

Table 7 — Sequence of service primitives for FTAM regime establishment — initiator

Previous file service event	May next issue			
	F-INITIALIZE request	F-TERMINATE request	F-U-ABORT request	Start File Selection Regimes
Idle	Yes			
F-INITIALIZE request			Yes	
F-INITIALIZE confirm (+ve)			Yes	Yes
F-INITIALIZE confirm (-ve)	Yes			
F-TERMINATE request			Yes	
F-TERMINATE confirm	Yes			
F-U-ABORT request	Yes			
F-U-ABORT indication	Yes			
F-P-ABORT indication	Yes			
Within file selection regimes			Yes	
End file selection regimes		Yes	Yes	Yes

Table 8 — Sequence of service primitives for FTAM regime establishment — responder

Previous file service event	May next issue			
	F-INITIALIZE response	F-TERMINATE response	F-U-ABORT request	Start File Selection Regimes
Idle				
F-INITIALIZE indication	Yes		Yes	
F-INITIALIZE response (+ve)			Yes	Yes
F-INITIALIZE response (-ve)				
F-TERMINATE indication		Yes	Yes	
F-TERMINATE response				
F-U-ABORT request				
F-U-ABORT indication				
F-P-ABORT indication				
Within file selection regimes			Yes	
End file selection regimes			Yes	Yes

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

Previous file service event	May next issue					
	F-SELECT request	F-DESELECT request	F-CREATE request	F-DELETE request	F-READ-ATTRIB request	F-CHANGE-ATTRIB request
Start file selection regimes	Kernel		Lmgt			
F-SELECT request		Grouping		G-Lmgt	G-Lmgt	G-Emgt
F-SELECT confirm (+ve)		Kernel		Lmgt	Lmgt	Emgt
F-SELECT confirm (-ve)	Kernel		Lmgt			
F-DESELECT request						
F-DESELECT confirm	Kernel		Lmgt			
F-CREATE request		Grouping		G-Lmgt	G-Lmgt	G-Emgt
F-CREATE confirm (+ve)		Lmgt		Lmgt	Lmgt	Emgt
F-CREATE confirm (-ve)	Lmgt		Lmgt			
F-DELETE request						
F-DELETE confirm	Lmgt		Lmgt			
F-READ-ATTRIB request		G-Lmgt		G-Lmgt		G-Emgt
F-READ-ATTRIB confirm		Lmgt		Lmgt	Lmgt	Emgt
F-CHANGE-ATTRIB request		G-Emgt		G-Emgt		
F-CHANGE-ATTRIB confirm		Emgt		Emgt	Emgt	Emgt
F-OPEN request						
F-OPEN confirm (+ve)						
F-OPEN confirm (-ve)		Kernel		Lmgt	Lmgt	Emgt
F-CLOSE request		Grouping		G-Lmgt	G-Lmgt	G-Emgt
F-CLOSE confirm		Kernel		Lmgt	Lmgt	Emgt
F-BEGIN-GROUP request	Grouping	Grouping	G-Lmgt	G-Lmgt	G-Lmgt	G-Emgt
F-BEGIN-GROUP confirm						
F-END-GROUP request						
F-END-GROUP confirm	Grouping	Grouping	G-Lmgt	G-Lmgt	G-Lmgt	G-Emgt
F-RECOVER request						
F-RECOVER confirm (+ve)						
F-RECOVER confirm (-ve)	Recover		Lmgt			
Bulk data transfer						
F-LOCATE request						
F-LOCATE confirm						
F-ERASE request						
F-ERASE confirm						