

INTERNATIONAL STANDARD

**ISO
8571-4**

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 4 : File Protocol Specification

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

Partie 4 : Spécification du protocole de transfert de fichier

Reference number
ISO 8571-4:1988 (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 8571-4 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 and B form an integral part of this International Standard. Annex C is for information only.

Contents	Page
0 Introduction	1
1 Scope.....	1
2 Field of application	2
3 References	2
4 Definitions and abbreviations.....	2
Section one: General	
5 Overview of the file protocol	3
5.1 Service supported by the file protocol	3
5.2 Services assumed by the basic file protocol	3
5.3 Functions provided	3
5.4 Model used.....	3
5.5 Correspondence between service primitive names and FPDU names.....	5
5.6 Protocol functional units	5
Section two: Basic file protocol	
6 State of the association provided.....	6
6.1 Operation of the protocol.....	6
6.2 Additional state information	6
7 File protocol data units.....	6
8 File initiating entity actions	8
8.1 FTAM regime establishment (Kernel)	8
8.2 FTAM regime termination (orderly) (Kernel)	8
8.3 File selection (Kernel)	8
8.4 File deselection (Kernel)	10
8.5 File creation (Limited management).....	10
8.6 File deletion (Limited management).....	10
8.7 Read attributes (Limited management).....	11
8.8 Change attributes (Enhanced management)	11
8.9 File open (Read, Write)	11
8.10 File close (Read, Write).....	12
8.11 Begin Group (Grouping)	13
8.12 End Group (Grouping).....	13
8.13 Recovery (Recovery)	13
8.14 Locate (Access)	14

8.15	Erase (Access).....	14
9	File responding entity actions	15
9.1	FTAM regime establishment (Kernel)	15
9.2	FTAM regime termination (orderly) (Kernel)	15
9.3	File selection (Kernel)	15
9.4	File deselection (Kernel)	16
9.5	File creation (Limited management).....	16
9.6	File deletion (Limited management).....	16
9.7	Read attributes (Limited management).....	16
9.8	Change attributes (Enhanced management)	17
9.9	File Open (Read, Write)	17
9.10	File close (Read, Write).....	18
9.11	Begin Group (Grouping).....	18
9.12	End Group (Grouping).....	18
9.13	Recovery (Recovery)	18
9.14	Locate (Access)	19
9.15	Erase (Access).....	19
10	File general actions.....	20
10.1	FTAM regime termination (abrupt)	20
10.2	Protocol errors.....	20
Section three: The basic bulk data transfer protocol		
11	State of bulk data transfer activity.....	21
11.1	State of the connection endpoint.....	21
11.2	Additional state of the entities	21
12	Bulk data transfer protocol data units	22
13	Bulk data transfer initiating entity actions	22
13.1	Read (Read).....	22
13.2	Write (Write)	22
13.3	Transfer end (Read, write)	22
14	Bulk data transfer responding entity actions.....	23
14.1	Read (Read).....	23
14.2	Write (Write)	23
14.3	Transfer end (Read, write)	23
15	Bulk data transfer sending entity actions	24

15.1	Send data (Read, write)	24
15.2	End of data transfer (Read, write)	24
15.3	Checkpoint insertion (Read, write)	24
16	Bulk data transfer receiving entity actions	24
16.1	Data transfer (Read, write)	24
16.2	End of data transfer (Read, write)	24
16.3	Checkpoint reception (Read, write)	24
17	Bulk data transfer general actions	25
17.1	Discarding (Read, Write)	25
17.2	Cancel (Read, write)	25
17.3	Restarting (Restart data transfer)	26
Section four: The error recovery protocol		
18	Protocol mechanisms	27
18.1	Error detection and reporting	27
18.2	Checkpoints	27
18.3	Negotiation of restart point	27
18.4	Docket contents	27
19	Specification of the error control protocol	28
19.1	Normal operation	28
19.2	Recovery from errors	30
Section five: Abstract Syntax		
20	Abstract Syntax Definition	32
20.1	Conventions	32
20.2	Primitive Types	32
20.3	ASN.1 Module Definition	33
20.4	Abstract Syntax Definition	45
21	Application Context Name	45
Section six: Conformance		
22	Conformance	46
22.1	Statement requirements	46
22.2	Static requirements	46
22.3	Dynamic requirements	46

Annexes

A Protocol state tables 47

B Reference to FTAM PDU definitions..... 86

C ASN.1 cross reference..... 87

Figures

1 Relationship between file protocol and service definitions 1

2 File Protocol Machine model 4

3 Extended File Protocol Machine model 4

4 Procedure for class I errors 30

5 Procedure for class II errors 30

6 Procedure for class III errors 31

7 FTAM regime PDUs..... 33

8 File selection and file open regime PDUs..... 35

9 Bulk data transfer PDUs 39

10 Application wide types 41

11 File attribute types 43

12 State machine system model..... 47

Tables

1	Presentation and Session Services required by FTAM Functional Units	3
2	Protocol Data Units, FTAM regime establishment.....	6
3	Protocol data units, File regime control	7
4	Relationship of F-INITIALIZE parameters to A-ASSOCIATE parameters	9
5	Protocol Data Units	21

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

Part 4 : File Protocol Specification

0 Introduction

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

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

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

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

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

ISO 8571 consists of the following four parts:

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

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

- Annex A - Protocol State Tables;
- Annex B - Reference to FTAM PDU Definitions;

and the following annex which does not form part of the standard:

- Annex C - ASN.1 cross reference.

1 Scope

ISO 8571-4 consists of four main sections:

- a) the basic protocol (in sections two and three), which supports the internal file service;
- b) the error recovery protocol (in section four), which supports the external file service.

For each of these, ISO 8571-4 includes a formal statement of the nature of the automaton giving the necessary behaviour of each of the participating entities. It states:

- 1) the actions to be taken on receiving request and response primitives issued by a file service user;
- 2) the actions to be taken on receiving indication and confirm primitives issued by the Presentation Service provider;
- 3) the actions to be taken as a result of events within the local system.
- c) the definition (in section five) of the abstract syntax required to convey the file protocol control information.
- d) the conformance requirements to be met by implementors of this protocol (in section six).

The scope of the File Protocol is limited to the interconnection of systems; it does not specify or restrict the possible implementation of interfaces within a computer system.

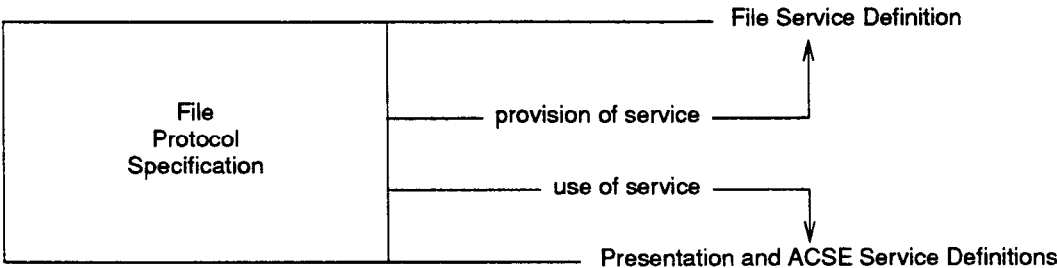


Figure 1 — Relationship between file protocol and service definitions

2 Field of application

The purpose of ISO 8571-4 within the OSI scheme is to specify the behaviour which must be exhibited by a system in order to take part in the provision of the file transfer access and management service.

The file protocol specification references three service definitions in order to express the environment within which it is applied. ISO 8571-3 defines the aims and objectives that the protocol must achieve. The Presentation Service (ISO 8822) and ACSE Service (ISO 8649-2) define the set of assumptions about the supporting facilities which the protocol may exploit (see figure 1).

3 References

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

ISO 7498-3, *Information Processing Systems - Open Systems Interconnection - Basic Reference Model*
- Part 3: Naming and Addressing

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

ISO 8571, *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.

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

ISO 8650, *Information Processing Systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element*.

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

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

ISO 8825, *Information Processing Systems - Open Systems Interconnection - Specification of basic encoding rules for Abstract Syntax Notation*

4 Definitions and abbreviations

Terms and abbreviations are defined in ISO 8571-1. The definitions of service primitive names and service state names given in ISO 8571-3 are also used in this part of ISO 8571.

Section one: General

5 Overview of the file protocol

5.1 Service supported by the file protocol

ISO 8571-4 specifies the following protocols which support the file services defined in ISO 8571-3:

- a) the basic protocol (see sections two and three), which supports the internal file service;
- b) the error recovery protocol (see section four), which supports the external file service.

5.2 Services assumed by the basic file protocol

The file protocol specified in this part of ISO 8571-4 assumes the use of the services listed in table 1. These are Presentation Services as defined in ISO 8822 and ACSE Association Control Services defined in ISO 8649. The OSI Reference Model allows the use of FTAM as one component in a distributed application; the specification of such an application before and after the FTAM regime is outside the scope of ISO 8571-4.

The assumptions made in different file protocol functional units (see 5.6) are listed separately.

5.3 Functions provided

5.3.1 Functions in the basic protocol

The basic protocol supports the internal file service (see ISO 8571-1), using the ACSE and Presentation Services. It supports the following functions:

- a) representation of the internal file service primitives as a sequence of protocol data units for transmission by the ACSE and Presentation Services;
- b) grouping, when appropriate, of the representations of logically separate service primitives as distinct data values in a single P-DATA request service primitive;
- c) ensuring the progress of the protocol.

5.3.2 Functions in the error recovery protocol

The error recovery protocol supports the external service (see ISO 8571-1), using the internal file service. It supports the following functions:

- a) management of error recovery information during the normal operation of the file service;
- b) restart of data transfer after interruption within the data transfer regime;
- c) recovery from abnormal termination of the file open or file selection regime.
- d) recovery from abnormal termination which destroys the internal file service regime.

5.4 Model used

In abstract, the operation of the protocol is modelled by the interaction of two file protocol machines (FPMs). The two FPMs communicate by means of the services available at their lower boundary, in such a way as to provide the services required at their upper boundary. These concepts are illustrated in figure 2.

The file service is defined asymmetrically with the file service user "A" being the initiator and file service user "B" being the responder.

The behaviour of each FPM is defined in terms of:

- a) the actions it takes:
 - 1) issue of indication or confirm service primitives to the upper service user;
 - 2) issue request or response service primitives to the lower service provider;
- b) the stimuli it receives:
 - 1) receipt of request or response service primitives from the upper service user;
 - 2) receipt of indication or confirm service primitives from the lower service provider;

Table 1 — Presentation and Session Services required by FTAM Functional Units

FTAM Functional Unit	Session Functional Unit	Presentation Functional Unit
Kernel(4)	Kernel Duplex Optionally: Resynch(1) Minor synch(2)	Kernel Duplex Optionally: Resynch(1) Minor synch(2) Context Management(3)
Recovery	Minor Synch	Minor Synch
Restart	Minor Synch Resynchronize	Minor Synch Resynchronize

NOTES

- 1 The resynchronize functional units shall be used if available.
- 2 Minor synch is required whenever the resynchronize functional unit is available. The presence of a confirmed minor synchronization point at the start of the bulk data regime is used to limit the destructive effects of resynchronization until after the regime has been fully established.
- 3 Use of the optional context management presentation functional unit implies the use of the typed data session functional unit.
- 4 The other functional units visible in the external service have the same presentation and session requirements as the kernel functional unit.

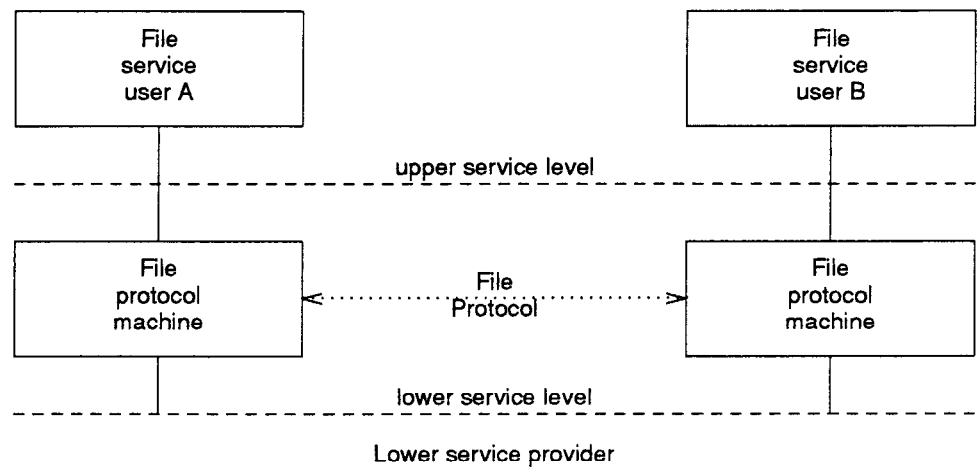


Figure 2 — File Protocol Machine model

- 3) local error indications;
- 4) management intervention.
- c) the information available:
 - 1) information associated with the lower service association endpoint; this information is lost if the lower service association ceases to exist;
 - 2) information associated with the upper service association endpoint; in the error recovery protocol, procedures are provided for recovery on loss of the lower service association, based on the information associated with the two upper service association endpoints.

- NOTES
- 1 For the basic protocol, the lower boundary service is the composite of ACSE and presentation service and the upper boundary service is the internal file service.
 - 2 For the error recovery protocol, the lower boundary service is the internal file service and the upper boundary is the external file service.
 - 3 The information describing the upper service association in the external file service is preserved across system failures, and so is available to allow meaningful recovery.

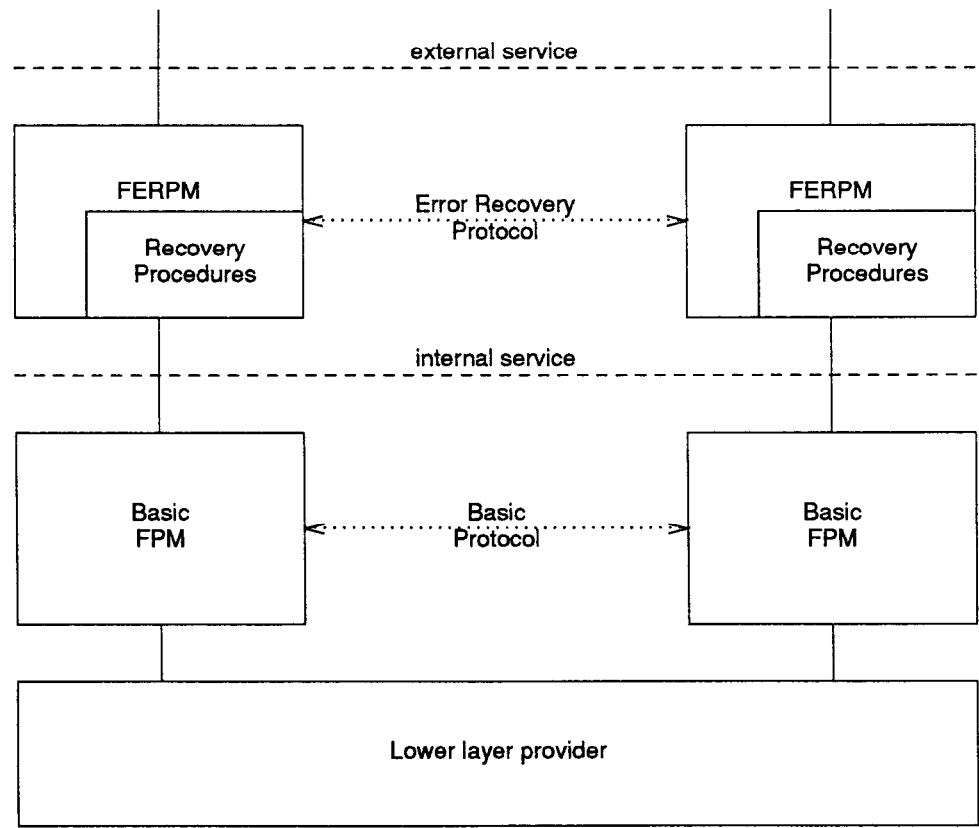


Figure 3 — Extended File Protocol Machine model

5.5 Correspondence between service primitive names and FPDU names

FPDUs which are sent as a result of a service primitive issued by a user of the supported service have the same name as that service primitive.

Examples:

Service primitive	FPDU
F-INITIALIZE request	F-INITIALIZE request FPDU
F-INITIALIZE response	F-INITIALIZE response FPDU

NOTES

- 1 Receipt of these FPDUs is signalled to the service user as F-INITIALIZE indication service primitives and F-INITIALIZE confirm service primitives respectively.
- 2 In some cases, several FPDUs may be grouped into a single PSDU. There is not therefore always a one-to-one correspondence between file service primitives and lower layer service primitives.

5.6 Protocol functional units

The functional units of the file protocol are equivalent to the functional units of the supported service:

- a) kernel functional unit;
- b) read functional unit;
- c) write functional unit;
- d) file access functional unit;
- e) limited file management functional unit;
- f) enhanced file management functional unit;
- g) grouping functional unit;
- h) FADU locking functional unit;
- i) recovery functional unit;
- j) restart functional unit.

NOTES

- 1 Each FPDU in a file protocol functional unit corresponds to the equivalent service primitive in the equivalent service functional unit.
- 2 Negotiation of a service functional unit implies the negotiation of the equivalent protocol functional unit.

Section two: Basic file protocol

6 State of the association provided

NOTE - The entities referred to in this section are basic protocol entities.

6.1 Operation of the protocol

The required behaviour of an entity conforming to the basic file protocol is expressed in the following clauses, in terms of the actions to be taken in response to each possible stimulus. Each clause is labelled with the name of the functional unit to which it applies, and the actions in a clause shall not apply if that functional unit is not successfully negotiated.

In specifying responses to service primitives issued by the IFS user, it is assumed that each type of service primitive shall only be received when the association endpoint is in a state in which the file service allows it to be received. Checking that this is the case is a matter of local interface organization.

For each PDU that may be received, the specification enumerates the states of the protocol in which the receiving entity can receive the PDU. If the action specified on receipt of a stimulus would require the issue of a service primitive which could not validly be issued in the current state, the entity concerned shall indicate a protocol error (see 10.2). If a PDU or a presentation service primitive is received under circumstances not enumerated, a protocol error has occurred. If a PDU or presentation service primitive is received that contains parameters with values inconsistent with previous values negotiated or requested during negotiation, then a protocol error has occurred.

6.2 Additional state information

The following sub-clauses define further state information associated with the basic protocol entities.

6.2.1 Expected responses list

The expected responses list records the original state and the operations remaining outstanding during a grouped exchange of PDUs and services (see 8.11, 8.12, 9.11 and 9.12). The expected responses list is an ordered list or "first in/first out queue" and holds the names of successive pending PDUs or primitives during use of the grouping mechanism. Initially the list is empty.

6.2.2 Initial state indicator

The initial state indicator records the state of the entity on invocation of the grouping mechanism. The defined values are the names of the states defined in ISO 8571-4 and "unset". Initially the value is "unset".

6.2.3 Next state indicator

The next state indicator records the next logical regime implied by receipt of grouped responses. The defined values are the state names "initialized", "selected" and "data transfer idle", and "unset". Initially the value is unset.

6.2.4 P-alter indicator

The p-alter indicator is used to record the need for additional alteration of the defined context set. The defined values are "unset" and "set". Initially, the value is "unset".

6.2.5 Bulk data transfer number

The bulk data transfer number identifies the bulk data transfers within an open regime. It is an integer in the range 0 to 999 999. Initially, the value is 0.

NOTE - The use of pieces of state information associated with an entity as an aid to specification does not imply that any distinct realization of such information is required within an entity.

7 File protocol data units

The file PDUs are complex data types defined by an abstract syntax definition (clause 20). This definition includes named parameters corresponding to those parameters in the file service carried directly by FTAM-PCI and those parameters necessary for the operation of the protocol.

For information which is always required default values are defined to avoid unnecessary overhead. The transfer syntax of these data types is negotiated and encoding performed by the presentation service provider.

The PDUs are transmitted as data values, either on a P-DATA primitive or in the user information of a ACSE primitive or in the user data of a Presentation Service primitive. The possible means of transmission of each PDU are listed in tables 2 and 3. (The means of transmission of the bulk data transfer PDUs are listed in table 5.) The PDUs marked with a * in the tables are carried on the response and confirm service primitives. All others are carried on request and indication service primitives.

Table 2 — Protocol Data Units, FTAM regime establishment

Name	Carried by	Functional units
F-INITIALIZE request	A-ASSOCIATE	kernel
F-INITIALIZE response*	A-ASSOCIATE	kernel
F-TERMINATE request	A-RELEASE	kernel
F-TERMINATE response*	A-RELEASE	kernel
F-P-ABORT request	A-ABORT	kernel
F-U-ABORT request	A-ABORT	kernel

Table 3 — Protocol data units, File regime control

Name	Carried by	Functional units
F-SELECT request	P-DATA	kernel
F-SELECT response	P-DATA	kernel
F-DESELECT request	P-DATA	kernel
F-DESELECT response	P-DATA	kernel
F-CREATE request	P-DATA	limited file management
F-CREATE response	P-DATA	limited file management
F-DELETE request	P-DATA	limited file management
F-DELETE response	P-DATA	limited file management
F-READ-ATTRIB request	P-DATA	limited file management
F-READ-ATTRIB response	P-DATA	limited file management
F-CHANGE-ATTRIB request	P-DATA	enhanced file management
F-CHANGE-ATTRIB response	P-DATA	enhanced file management
F-OPEN request	P-DATA	read, write
F-OPEN response	P-DATA	read, write
F-CLOSE request	P-DATA	read, write
F-CLOSE response	P-DATA	read, write
F-BEGIN-GROUP request	P-DATA	grouping
F-BEGIN-GROUP response	P-DATA	grouping
F-END-GROUP request	P-DATA	grouping
F-END-GROUP response	P-DATA	grouping
F-RECOVER request	P-DATA	recovery
F-RECOVER response	P-DATA	recovery
F-LOCATE request	P-DATA	access
F-LOCATE response	P-DATA	access
F-ERASE request	P-DATA	access
F-ERASE response	P-DATA	access

A P-DATA primitive conveys a series of data values. This series of data values is referred to in ISO 8571 as a Presentation Service Data Unit (PSDU). The protocol specifies, in terms of the circumstances in which the sequence of data values in a PSDU is terminated, the exact contents of each P-DATA issued. The series of data units may not be divided into several P-DATA primitives in ways not specified by this protocol.

File PDUs are transmitted in one particular presentation context, called the FTAM-PCI context. This is defined to be the context used for the user information data value of the ACSE A-ASSOCIATE primitive, and shall be a context corresponding to the FTAM-PCI abstract syntax defined in ISO 8571-4.

NOTE - Data values in any other context are not File PDUs and this fact is used to ensure the transparency of user data.

The receiver recognizes PDUs on the basis that a File PDU shall:

- a) be transmitted in the FTAM-PCI context,
- b) consist of a single complete value from the FTAM-PCI abstract syntax.

NOTE - A data value which differs in any material way, either by omission of a mandatory parameter, or by addition of an undefined parameter, or by use of an invalid parameter value, from the defined values of the File PDU data types is not a File PDU.

An entity shall signal a protocol error (10.2) if:

- c) it receives any ACSE or presentation service primitives not specified for the functional units negotiated;
- d) it receives a sequence of data values in the FTAM PCI context which does not form a defined PDU.
- e) it receives any PDU which does not belong to the selected functional units.

8 File initiating entity actions

8.1 FTAM regime establishment (Kernel)

8.1.1 On receiving an F-INITIALIZE request service primitive from the IFS user while in an "idle" state, the entity shall:

- a) construct an F-INITIALIZE request PDU based on the parameters of the F-INITIALIZE request service primitive;
- b) set the protocol version parameter of that PDU to "version-1" and no other version; the protocol version parameter shall be set so that bits in the parameter corresponding to other versions shall not be set to either "0" or "1" as part of its value;
- c) on the basis of local knowledge set the implementation information parameter;
- d) add the PDU constructed as user information on an A-ASSOCIATE request, setting the values of the parameters of the A-ASSOCIATE request according to the information in table 4. Set the values of the presentation requirements and session requirements parameters of the A-ASSOCIATE request to propose at least the corresponding functional units as described in table 1 derived from the parameters received. Derive a list of abstract syntaxes from
 - 1) those needed to support the document types and abstract syntaxes named in the contents type list parameter with no syntaxes duplicated;
 - 2) the FTAM PCI context and any EXTERNAL syntaxes referenced from it;
 - 3) any other abstract syntaxes implied by non-FTAM parts of the application context;

and set the value of the application context parameter on A-ASSOCIATE to request this list of abstract syntaxes.

- e) enter the "initialize pending" state.

NOTES

- 1 If the resynchronize functional unit is not successfully negotiated (see clause 17) then in certain situations of flow control blockage the cancel operation may be indefinitely delayed.
- 2 The implementation information parameter contains information specific to individual implementations. It may be set by omission. Its contents are not used for negotiation.

8.1.2 On receiving an A-ASSOCIATE confirm with an F-INITIALIZE response PDU as user information while in the state "initialize pending", the entity shall:

- a) if
 - 1) the result parameter of the A-ASSOCIATE indicates failure,
 or
 - 2) the negotiable parameters have values inconsistent with the requested values.

then issue an F-INITIALIZE confirm service primitive to the IFS user, with parameters derived from the A-ASSOCIATE confirm, and thereafter cease to exist;

Consistency is achieved if:

- The returned value of the requested attribute groups parameter refers to a subset of the requested value;
- The returned value of the requested functional units refers to a subset of the requested value;

— The presentation functional units negotiated support the FTAM functional units negotiated;

— The Service Class is one and only one of the Service Classes indicated on the F-INITIALIZE request, or, in the case where the transfer and management service class was requested, then the transfer service class or the management service class is also consistent;

- b) if the result parameter of the A-ASSOCIATE primitive indicates success and the negotiable parameter values are consistent as described in (a), then issue an F-INITIALIZE confirm service primitive to the IFS user, with parameters derived from the data values received and from the parameters received on the A-ASSOCIATE confirm;
- c) enter the state "initialized".

NOTE - Once initialized it is possible for several presentation contexts supporting the FTAM PCI abstract syntax to exist in the defined context list. The entity shall subsequently interpret only the first presentation context supporting FTAM PCI encountered as FTAM PCI; any other instances do not affect the state of the FPM.

8.1.3 On receiving an A-ASSOCIATE confirm, without user information and with a result parameter indicating failure, while in the state "initialize pending", the entity shall:

- a) issue an F-INITIALIZE confirm service primitive to the IFS user with a state and action result parameters indicating failure;
- b) thereafter cease to exist.

8.2 FTAM regime termination (orderly) (Kernel)

8.2.1 On receiving an F-TERMINATE request service primitive from the IFS user while in an "initialized" state, the entity shall:

- a) send an F-TERMINATE request PDU as user information on an A-RELEASE request primitive;
- b) enter the state "terminate pending".

8.2.2 On receiving an F-TERMINATE response PDU while in the state "terminate pending", the entity shall:

- a) issue an F-TERMINATE confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) thereafter cease to exist.

8.3 File selection (Kernel)

8.3.1 On receiving an F-SELECT request service primitive from the IFS user while in the state "initialized", the entity shall:

- a) add an F-SELECT request PDU as a data value to the current PSDU;
- b) terminate the PSDU.
- c) enter the state "select pending".

8.3.2 On receiving an F-SELECT request service primitive from the IFS user while in the state "grouping" with the expected response list empty and the initial state indicator having value "initialized", the entity shall:

- a) add an F-SELECT request PDU as a data value to the current PSDU;
- b) add the value "F-SELECT response" to the end of the expected responses list.

Table 4 — Relationship of F-INITIALIZE parameters to A-ASSOCIATE parameters

A-ASSOCIATE	F-INITIALIZE			
	Request	Indication	Response	Confirm
Called Application Entity Title (8)	Called Application Entity Title	Called Application Entity Title	—	—
Calling Application Entity Title (8)	Calling Application Entity Title	Calling Application Entity Title	—	—
Responding Application Entity Title (8)	—	—	Responding Application Entity Title	Responding Application Entity Title
Application Context Name	Application Context Name(1)	—	Application Context Name	—
Called Presentation address	Called Presentation address	Called Presentation address	—	—
Calling Presentation address	Calling Presentation address	Calling Presentation address	—	—
Responding Presentation address	—	—	Responding Presentation address	Responding Presentation address
Presentation Context Definition List(2)	Derived from Contents Type List(3)	Contents Type List	—	—
Presentation Context Result List	—	—	Derived from Contents Type list(4)	Contents Type List
QoS	CQoS(5)	CQoS	CQoS	CQoS
Presentation Requirements	Derived from Presentation Context Management(6)	—	Derived from Presentation Context Management(6)	—
Session Requirements	Derived from Functional Units(7)	—	Derived from Functional Units(7)	—
Initial Synch Point Serial Number	Set to "1" by FPM	—	—	—
Initial Assignment of Tokens	Set to "Initiator" by FPM	—	—	—
Result	—	—	State Result	State Result

— Request and response columns show how A-ASSOCIATE parameters are set from F-INITIALIZE parameters.

— Indication and confirm columns show how F-INITIALIZE parameters are set from A-ASSOCIATE parameters.

— Dash indicates A-ASSOCIATE parameter not used or has no relationship to an F-INITIALIZE parameter.

NOTES

1 Any Application Context Name must support at least the function implied by the application context identified by "{iso standard 8571 application-context(1) iso-ftam(1)}", (see clause 21); if no value is provided by the user then that object identifier shall be assigned by the FPM, otherwise the Application Context Name shall be derived from the value provided by the IFS user.

2 The Presentation Contexts Definition List must contain at least those presentation contexts needed to support the current application context as implied by FTAM (see clause 21) and the Contents Type List.

3 The Presentation Context Definition list contains "ISO 8571-FTAM PCI" and the abstract syntax names of the Contents Type List.

4 The Contents Type List in these cases may affect the Available Contents Type List activity attribute.

5 CQoS is used as QoS is noted in ACSE and Presentation.

6 Presentation Requirements is set to "Context Management" if the Presentation Context Management parameter of F-INITIALIZE equals True; it is empty otherwise.

7 Session Requirements is set to the union of the session requirements necessary to support the selected protocol functional units.

8 Application Entity Title is used here and elsewhere in ISO 8571 for the collection of naming parameters defined in ISO 8649. The relation between these naming concepts is defined in ISO 7498-3.

9 The A-ASSOCIATE parameters Single Presentation Context, Default Presentation Context Name, Default Presentation Context Result and Session Connection Identifier are not related directly to any F-INITIALIZE parameters.

8.3.3 On receiving an F-SELECT response PDU while in the state "select pending" and with the negotiable parameters having values returned consistent with the requested values, where the requested access must refer to a subset of the actions requested, the entity shall:

- a) issue an F-SELECT confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) if the primitive issued did not carry a state result parameter indicating failure, enter the state "selected";
- c) if the primitive issued carried a state result parameter indicating failure, then enter the state "initialized".

8.3.4 On receiving an F-SELECT response PDU while in state "grouping" with the first element of the expected responses list having value "F-SELECT response" and with returned parameters consistent as described in 8.3.3, the entity shall:

- a) issue an F-SELECT confirm service primitive to the IFS user;
- b) if the PDU did not carry a state result indicating failure set the value of the next state indicator to "selected";
- c) delete the first element of the expected responses list.

8.4 File deselection (Kernel)

8.4.1 On receiving an F-DESELECT request service primitive from the IFS user while in the state "selected", the entity shall:

- a) add an F-DESELECT request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "deselect pending".

8.4.2 On receiving an F-DESELECT request service primitive from the IFS user while in the state "grouping" with the last element of the expected response list having one of the values "F-CLOSE response", "F-CREATE response", "F-SELECT response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response", the entity shall:

- a) add an F-DESELECT request PDU as a data value to the current PSDU;
- b) add the value "F-DESELECT response" to the end of the expected responses list.

8.4.3 On receiving an F-DESELECT response PDU while in the state "deselect pending", the entity shall:

- a) issue an F-DESELECT confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "initialized".

8.4.4 On receiving an F-DESELECT response PDU while in state "grouping" with the first element of the expected responses list having value "F-DESELECT response", the entity shall:

- a) issue an F-DESELECT confirm service primitive to the IFS user;
- b) set the value of the next state indicator to "initialized";
- c) delete the first element of the expected responses list.

8.5 File creation (Limited management)

8.5.1 On receiving an F-CREATE request service primitive from the IFS user while in the state "initialized", the entity shall:

- a) add an F-CREATE request PDU as a data value to the current PSDU;
- b) terminate the current PSDU and enter the state "create pending".

8.5.2 On receiving an F-CREATE request service primitive from the IFS user while in the state "grouping" with the expected response list empty and the initial state indicator having value "initialized", the entity shall:

- a) add an F-CREATE request PDU as a data value to the current PSDU;
- b) add the value "F-CREATE response" to the end of the expected responses list.

8.5.3 On receiving an F-CREATE response PDU while in the state "create pending" and with the following negotiable parameters having values returned consistent with the requested values:

- a) initial attributes are within the requested attribute groups successfully negotiated at filestore initialization and
- b) requested access must refer to a subset of the actions requested,

the entity shall:

- c) issue an F-CREATE confirm service primitive to the IFS user, with parameters derived from the data values received;
- d) if the primitive issued did not carry a state result parameter indicating failure, enter the state "selected";
- e) if the primitive issued carried a state result parameter indicating failure, enter the state "initialized".

8.5.4 On receiving an F-CREATE response PDU while in state "grouping" with the first element of the expected responses list having value "F-CREATE response" and with returned parameters consistent as described in 8.5.3, the entity shall:

- a) issue an F-CREATE confirm service primitive to the IFS user;
- b) if the PDU did not carry a state result indicating failure set the value of the next state indicator to "selected".
- c) delete the first element of the expected responses list.

8.6 File deletion (Limited management)

8.6.1 On receiving an F-DELETE request service primitive from the IFS user while in state "selected", the entity shall:

- a) add an F-DELETE request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "delete pending".

8.6.2 On receiving an F-DELETE request service primitive from the IFS user while in the state "grouping" with the last element of the expected response list having one of the

values "F-CLOSE response", "F-CREATE response", "F-SELECT response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response", the entity shall:

- a) add an F-DELETE request PDU as a data value to the current PSDU;
- b) add the value "F-DELETE response" to the end of the expected responses list.

8.6.3 On receiving an F-DELETE response PDU while in the state "delete pending", the entity shall:

- a) issue an F-DELETE confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "initialized".

8.6.4 On receiving an F-DELETE response PDU while in state "grouping" with the first element of the expected responses list having value "F-DELETE response", the entity shall:

- a) issue an F-DELETE confirm service primitive to the IFS user;
- b) set the value of the next state indicator to "initialized";
- c) delete the first element of the expected responses list.

8.7 Read attributes (Limited management)

8.7.1 On receiving an F-READ-ATTRIB request service primitive from the IFS user while in the state "selected", the entity shall:

- a) add an F-READ-ATTRIB request PDU to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "read attribute pending".

8.7.2 On receiving an F-READ-ATTRIB request service primitive from the IFS user while in the state "grouping" with the last element of the expected response list having one of the values "F-CLOSE response", "F-CREATE response" or "F-SELECT response" or is empty the entity shall:

- a) add an F-READ-ATTRIB request PDU as a data value to the current PSDU;
- b) add the value "F-READ-ATTRIB response" to the end of the expected responses list.

8.7.3 On receiving an F-READ-ATTRIB response PDU while in the state "read attribute pending", having successfully negotiated the limited file management functional unit at filestore initialization and with the following negotiable parameters having values returned consistent with the requested values, the entity shall:

- a) issue an F-READ-ATTRIB confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "selected".

The attributes returned are within the requested attribute groups successfully negotiated at filestore initialization and were named in the attribute names parameter on the request PDU sent.

8.7.4 On receiving an F-READ-ATTRIB response PDU while in state "grouping", with the first element of the expected responses list having value "F-READ-ATTRIB response" and with returned parameters consistent as

described in 8.7.3, the entity shall:

- a) issue an F-READ-ATTRIB confirm service primitive to the IFS user with parameters derived from the data values received;
- b) delete the first element of the expected responses list.

8.8 Change attributes (Enhanced management)

8.8.1 On receiving an F-CHANGE-ATTRIB request service primitive from the IFS user while in the state "selected", the entity shall:

- a) add an F-CHANGE-ATTRIB request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "change attribute pending".

8.8.2 On receiving an F-CHANGE-ATTRIB request service primitive from the IFS user while in the state "grouping" with enhanced file management functional unit selected, and with the last element of the expected response list having one of the values "F-READ-ATTRIB response", "F-CLOSE response", "F-CREATE response" or "F-SELECT response" or is empty the entity shall:

- a) add an F-CHANGE-ATTRIB request PDU as a data value to the current PSDU;
- b) add the value "F-CHANGE-ATTRIB response" to the end of the expected responses list.

8.8.3 On receiving an F-CHANGE-ATTRIB response PDU while in the state "change attribute pending" having successfully negotiated the enhanced file management functional unit and suitable access at filestore initialization, and with the negotiable parameters having values returned consistent with the requested values the entity shall:

- a) issue an F-CHANGE-ATTRIB confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "selected".

8.8.4 On receiving an F-CHANGE-ATTRIB response PDU while in state "grouping" with the first element of the expected responses list having value "F-CHANGE-ATTRIB response" and with returned parameters consistent as described in 8.7.3, the entity shall:

- a) issue an F-CHANGE-ATTRIB confirm service primitive to the IFS user with parameters derived from the data values received;
- b) delete the first element of the expected responses list.

8.9 File open (Read, Write)

8.9.1 On receiving an F-OPEN request service primitive from the IFS user while in state "selected", the entity shall:

- a) add an F-OPEN request PDU as a data value to the current PSDU;
- b) if
 - 1) the presentation context management functional unit has been successfully negotiated
 and

- 2) the initiating entity wishes to have any abstract syntaxes supported by presentation contexts in the defined context set deleted on the basis of local system

or network management considerations for the successful establishment of the open regime (see 9.9.4)

then

the associated abstract syntax names should be sent in the remove contexts parameter of the F-OPEN request PDU;

c) if

1) the presentation context management functional unit has been successfully negotiated and

2) the initiating entity wishes to have presentation contexts for any abstract syntaxes added to defined context set on the basis of local system or network management considerations for the establishment of the open regime (see 9.9.4)

then

the associated abstract syntax names should be sent in the define contexts parameter of the F-OPEN request PDU;

d) terminate the current PSDU,

e) enter the state "open pending".

8.9.2 On receiving an F-OPEN request service primitive from the IFS user while in the state "grouping" with the last element of the expected responses list having one of the values: "F-SELECT response", "F-CREATE response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response" the entity shall:

a) add an F-OPEN request PDU as a data value to the current PSDU;

b) add the value "F-OPEN response" to the end of the expected responses list.

8.9.3 On receiving an F-OPEN response PDU while in the state "open pending" and with the negotiable parameters having values returned consistent with the requested values in accordance with the rules for negotiation given in ISO 8571-3, the entity shall:

a) if the PDU issued carried a state result parameter indicating failure, issue an F-OPEN confirm primitive to the IFS user and enter the state "selected";

b) if the value of the presentation action parameter is true, set the p-alter indicator;

c) if the p-alter indicator is not set issue an F-OPEN confirm primitive to the IFS user, with parameters derived from the data values received, and enter the state "data transfer idle";

d) if the p-alter indicator is set preserve the PDU for further processing and enter the state "p-alter context pending".

The processing mode must imply only those actions permissible by the successfully negotiated functional units and requested access at filestore initialization.

The concurrency control must be at least as restrictive as that successfully negotiated at file selection (see ISO 8571-3).

8.9.4 On receiving an F-OPEN response PDU while in state "grouping" with the first element of the expected responses list having value "F-OPEN response" and with returned parameters consistent as described in 8.9.3, the entity shall:

a) if the PDU did not carry a state result indicating failure and if the value of the presentation action parameter is true, set the p-alter indicator;

b) if the p-alter indicator is not set issue an F-OPEN confirm primitive to the IFS user, with parameters derived from the data values received;

c) if the PDU did not carry a state result indicating failure set the value of the next state indicator to "data transfer idle";

d) if the p-alter indicator is set then preserve the PDU for further processing;

e) delete the first element of the expected responses list.

8.9.5 On receiving a P-ALTER-CONTEXT indication primitive from the presentation service provider while in the state "p-alter context pending" and the p-alter indicator is set, the entity shall:

a) a presentation context definition addition result list indicating which of the proposed presentation contexts can be supported;

b) construct a presentation context deletion result list indicating which of the presentation contexts proposed for deletion may be deleted;

c) issue a P-ALTER-CONTEXT response using both lists constructed in a) and b);

d) unset the p-alter indicator;

e) if a F-OPEN response PDU was being preserve then issue an F-OPEN confirm service primitive to the IFS user.

8.9.6 If an F-OPEN response PDU is being preserved in the state "p-alter context pending", and the p-alter indicator is unset, the entity shall:

a) if the open regime has been established, but presentation contexts supporting the requested abstract syntaxes have not all been successfully negotiated set the action result parameter to indicate a recoverable error, and optionally add a diagnostic describing the context negotiation result based on local management considerations;

b) issue an F-OPEN confirm service primitive to the IFS user, with parameters derived from the data values received;

c) if the primitive issued did not carry a state result parameter indicating failure, enter the state "data transfer idle";

d) if the primitive issued carried a state result parameter indicating failure, enter the state "selected".

NOTE - If the resultant facilities are insufficient, the IFS user should terminate the regime.

8.10 File close (Read, Write)

8.10.1 On receiving an F-CLOSE request service primitive from the IFS user while in a "data transfer idle" state, the entity shall:

a) add an F-CLOSE request PDU to the current PSDU;

b) terminate the current PSDU,

c) enter the state "close pending".

NOTE - F-CLOSE request carries an action result that is used as a signalling mechanism between FERPM peers (see 19.1.3). No action result has any effect on the FPM.

8.10.2 On receiving an F-CLOSE request service primitive from the IFS user while in the state "grouping" with the expected response list empty and the initial state indicator having value "data transfer idle", the entity shall:

- a) add an F-CLOSE request PDU as a data value to the current PSDU;
- b) add the value "F-CLOSE response" to the end of the expected responses list.

8.10.3 On receiving an F-CLOSE response PDU while in the state "close pending", the entity shall:

- a) issue an F-CLOSE confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "selected".

8.10.4 On receiving an F-CLOSE response PDU while in state "grouping" with the first element of the expected responses list having value "F-CLOSE response", the entity shall:

- a) issue an F-CLOSE confirm service primitive to the IFS user with parameters derived from the data values received;
- b) set the value of the next state indicator to "selected";
- c) delete the first element of the expected responses list.

8.11 Begin Group (Grouping)

8.11.1 On receiving an F-BEGIN-GROUP request service primitive from the IFS user while in either the state "initialized" or the state "data transfer idle", the entity shall:

- a) add an F-BEGIN-GROUP request PDU as a data value to the current PSDU;
- b) set the value of the initial state indicator to the current state;
- c) enter the state "grouping".

8.11.2 On receiving an F-BEGIN-GROUP response PDU while in the state "group pending", the entity shall:

- a) issue an F-BEGIN-GROUP confirm service primitive to the IFS user;
- b) enter the state "grouping".

8.12 End Group (Grouping)

8.12.1 On receiving an F-END-GROUP request service primitive from the IFS user while in the state "grouping", the entity shall:

- a) add an F-END-GROUP request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "group pending".

8.12.2 On receiving an F-END-GROUP response PDU while in state "grouping", the entity shall:

- a) issue an F-END-GROUP confirm service primitive to the IFS user;
- b) if the requested threshold has been attained by counting consecutive primitives with either state result parameters indicating success or no state result parameter and

1) the p-alter indicator is set then enter the state "p-alter context pending"

2) the p-alter indicator is unset then enter the state named by the next state indicator.

c) If the requested threshold has not been attained then enter the state named by the initial state indicator.

d) Clear the expected responses list, unset the next state indicator and unset the initial state indicator.

NOTE - If Class II errors are signalled, the open regime has been established, but not all the necessary abstract syntaxes have corresponding supporting presentation contexts as a result of the preceding P-ALTER-CONTEXT exchange. If the resultant facilities are insufficient, the IFS user may release the regime.

8.13 Recovery (Recovery)

8.13.1 On receiving an F-RECOVER request service primitive from the IFS user while in state "initialized", the entity shall:

a) add an F-RECOVER request PDU as a data value to the current PSDU;

b) if

1) the presentation context management functional unit has been successfully negotiated
and

2) the initiating entity wishes to have presentation contexts for any abstract syntaxes added to defined context set on the basis of local system or network management considerations for the successful establishment of the open regime (see 9.9.4)
then

the associated abstract syntax names should be sent in the define-contexts parameter of the F-RECOVER request PDU;

c) if

1) the presentation context management functional unit has been successfully negotiated
and

2) the initiating entity wishes to have any abstract syntaxes supported by presentation contexts in the defined context set deleted on the basis of local system or network management considerations for the establishment of the open regime (see 9.9.4)
then

the associated abstract syntax names should be sent in the remove contexts parameter of the F-RECOVER request PDU;

d) terminate the current PSDU,

e) enter the state "recover pending".

8.13.2 On receiving an F-RECOVER response PDU while in the state "recover pending" and with the negotiable parameters having values returned consistent with the requested values in accordance with the rules for negotiation given in ISO 8571-3, the entity shall:

a) if the PDU issued carried a state result parameter indicating failure, issue an F-RECOVER confirm primitive to the IFS user and enter the state "initialized";

b) if the value of the presentation action parameter is true, set the p-alter indicator;

c) if the p-alter indicator is not set issue an F-RECOVER confirm primitive to the IFS user, with parameters derived from the data values received, and enter the state "data transfer idle recover";

d) if the p-alter indicator is set preserve the PDU for further processing and enter the state "p-alter context pending".

The processing mode must imply only those actions permissible by the successfully negotiated functional units and requested access at filestore initialization.

The concurrency control must be at least as restrictive as that successfully negotiated at file selection (see ISO 8571-3).

8.13.3 On receiving a P-ALTER-CONTEXT indication primitive from the presentation service provider while in the state "p-alter context pending" and the p-alter indicator is set, the entity shall:

- a) construct a presentation context definition result list indicating which of the proposed presentation contexts can be supported;
- b) construct a presentation context deletion result list indicating which of the presentation contexts proposed for deletion may be deleted;
- c) issue a P-ALTER-CONTEXT response using both lists constructed in a) and b);
- d) unset the p-alter indicator.

8.13.4 If an F-RECOVER response PDU is being preserved in the state "p-alter context pending", and the p-alter indicator is unset, the entity shall:

- a) if the open regime has been established, but not all the necessary abstract syntaxes have corresponding supporting presentation contexts as a result of the preceding P-ALTER-CONTEXT exchange then set the action result parameter to indicate a recoverable error, and optionally add a diagnostic describing the context negotiation result based on local management considerations;
- b) issue an F-RECOVER confirm service primitive to the IFS user, with parameters derived from the data values received;

c) if the primitive issued did not carry a state result parameter indicating failure, enter the state "data transfer idle recover";

d) if the primitive issued carried a state result parameter indicating failure, enter the state "initialized".

NOTE - If the resultant facilities are insufficient, the IFS user should terminate the regime.

8.14 Locate (Access)

8.14.1 On receiving an F-LOCATE request service primitive from the IFS user while in a "data transfer idle" state, the entity shall:

- a) add an F-LOCATE request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "locate pending".

8.14.2 On receiving an F-LOCATE response PDU while in the state "locate pending", the entity shall:

- a) issue an F-LOCATE confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "data transfer idle".

8.15 Erase (Access)

8.15.1 On receiving an F-ERASE request service primitive from the IFS user while in a "data transfer idle" state, the entity shall:

- a) add an F-ERASE request PDU to the current PSDU;
- b) terminate the current PSDU
- c) enter the state "erase pending".

8.15.2 On receiving an F-ERASE response PDU while in the state "erase pending", the entity shall:

- a) issue an F-ERASE confirm service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "data transfer idle".

9 File responding entity actions

9.1 FTAM regime establishment (Kernel)

9.1.1 On receiving an F-INITIALIZE request PDU as user information on an A-ASSOCIATE indication while in an "idle" state, the entity shall:

- a) if the entity is incapable of establishing the FTAM regime, construct an F-INITIALIZE response PDU with a state result parameter indicating protocol error, and add the PDU constructed as user information on an A-ASSOCIATE response with result parameter set to "rejected by responder (permanent)", issue the A-ASSOCIATE response and thereafter cease to exist;
- b) if the PDU contains a protocol version parameter indicating that the initiating FPM is incompatible with version 1 by not having the "version-1" bit set to "1" then construct an F-INITIALIZE response PDU with state result set to "failure", the action result set to "permanent error" and the protocol version parameter set to indicate that it supports version 1, and add the PDU constructed as user information on an A-ASSOCIATE response with result parameter set to "rejected by responder (permanent)", issue the A-ASSOCIATE response and thereafter cease to exist;
- c) issue an F-INITIALIZE indication primitive to the IFS user, with parameters derived from the data values received and from the parameters received on the A-ASSOCIATE indication, such that
 - 1) if the contents type list parameter is not present the list of context definitions to be performed is empty;
 - 2) if the elements of the presentation context definition list of the P-CONNECT indication cannot support items of the contents type list of the F-INITIALIZE request PDU then delete from the contents type list
 - those abstract syntaxes that cannot be supported and
 - those document types which reference abstract syntaxes that cannot be supported;
- d) enter the state "f-initialize pending".

9.1.2 On receiving an F-INITIALIZE response service primitive from the IFS user while in an "f-initialize pending" state, the entity shall:

- a) construct an F-INITIALIZE response PDU;
- b) set the protocol version parameter of that PDU to "version-1"; other version bits in the parameter corresponding to other versions shall not be set;
- c) on the basis of local knowledge set the implementation information parameter;
- d) send the PDU constructed as user information on an A-ASSOCIATE response. If the F-INITIALIZE response carries a state result parameter indicating failure, then set the result parameter on the A-ASSOCIATE response to indicate failure, otherwise indicate success;
- e) if the F-INITIALIZE response carried a state result parameter indicating failure, thereafter cease to exist.

NOTES

1 Once initialized it is possible for several presentation contexts supporting the FTAM PCI abstract syntax to exist in the defined context list. The entity shall subsequently interpret only the first presentation context supporting the FTAM PCI abstract syntax

encountered as FTAM PCI; any data in other presentation contexts does not affect the state of the FPM.

2 The implementation information parameter contains information specific to individual implementations. It may be omitted. Its contents shall not be used for negotiation.

9.2 FTAM regime termination (orderly) (Kernel)

9.2.1 On receiving an F-TERMINATE request PDU while in a "initialized" state, the entity shall:

- a) issue an F-TERMINATE indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-terminate pending".

9.2.2 On receiving an F-TERMINATE response service primitive from the IFS user while in the state "f-terminate pending", the entity shall:

- a) construct an F-TERMINATE response PDU;
- b) send the PDU constructed as user information on an A-RELEASE response primitive;
- c) thereafter cease to exist.

9.3 File selection (Kernel)

9.3.1 On receiving an F-SELECT request PDU while in a "initialized" state, the entity shall:

- a) issue an F-SELECT indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-select pending".

9.3.2 On receiving an F-SELECT request PDU while in the state "grouping" with the expected response list empty and the initial state indicator having value "initialized", the entity shall:

- a) issue an F-SELECT indication to the IFS user;
- b) add the value "F-SELECT response" to the end of the expected responses list.

9.3.3 On receiving an F-SELECT response service primitive from the IFS user while in an "f-select pending" state, the entity shall:

- a) add an F-SELECT response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) If the PDU contained a state result parameter indicating failure, enter the state "initialized"; otherwise, enter the state "selected".

9.3.4 On receiving an F-SELECT response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-SELECT response", the entity shall:

- a) add an F-SELECT response PDU as a data value to the current PSDU;
- b) if the response did not carry a state result indicating failure set the value of the next state indicator to "selected";
- c) delete the first element of the expected responses list.

9.4 File deselection (Kernel)

9.4.1 On receiving an F-DESELECT request PDU while in a "selected" state, the entity shall:

- a) issue an F-DESELECT indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-deselect pending".

9.4.2 On receiving an F-DESELECT request PDU while in the state "grouping" with the first element of the expected response list having one of the values "F-CLOSE response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response", the entity shall:

- a) issue an F-DESELECT indication to the IFS user;
- b) add the value "F-DESELECT response" to the end of the expected responses list.

9.4.3 On receiving an F-DESELECT response service primitive from the IFS user while in the state "f-deselect pending", the entity shall:

- a) add an F-DESELECT response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "initialized".

9.4.4 On receiving an F-DESELECT response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-DESELECT response", the entity shall:

- a) add an F-DESELECT response PDU to the current PSDU;
- b) set the value of the next state indicator to "initialized";
- c) delete the first element of the expected responses list.

9.5 File creation (Limited management)

9.5.1 On receiving an F-CREATE request PDU while in a "initialized" state, the entity shall:

- a) issue an F-CREATE indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-create pending".

9.5.2 On receiving an F-CREATE request PDU while in the state "grouping" with the expected response list empty and the initial state indicator having value "initialized", the entity shall:

- a) issue an F-CREATE indication service primitive to the IFS user;
- b) add the value "F-CREATE response" to the end of the expected responses list.

9.5.3 On receiving an F-CREATE response service primitive from the IFS user while in the state "f-create pending", the entity shall:

- a) add an F-CREATE response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) If the PDU contained a state result parameter indicating failure, enter the state "initialized"; otherwise, enter the state "selected".

9.5.4 On receiving an F-CREATE response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-CREATE response", the entity shall:

- a) add an F-CREATE response PDU to the current PSDU;
- b) if the PDU did not carry a state result indicating failure set the value of the next state indicator to "selected";
- c) delete the first element of the expected responses list.

9.6 File deletion (Limited management)

9.6.1 On receiving an F-DELETE request PDU while in a "selected" state, the entity shall:

- a) issue an F-DELETE indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-delete pending".

9.6.2 On receiving an F-DELETE request PDU while in the state "grouping" with the first element of the expected response list having one of the values "F-CLOSE response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response", the entity shall:

- a) issue an F-DELETE indication service primitive to the IFS user;
- b) add the value "F-DELETE response" to the end of the expected responses list.

9.6.3 On receiving an F-DELETE response service primitive from the IFS user while in state "f-delete pending", the entity shall:

- a) add an F-DELETE response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "initialized".

9.6.4 On receiving an F-DELETE response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-DELETE response", the entity shall:

- a) add an F-DELETE response PDU to the current PSDU;
- b) set the value of the next state indicator to "initialized";
- c) delete the first element of the expected responses list.

9.7 Read attributes (Limited management)

9.7.1 On receiving an F-READ-ATTRIB request PDU while in a "selected" state, the entity shall:

- a) issue an F-READ-ATTRIB indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-read attribute pending".

9.7.2 On receiving an F-READ-ATTRIB request PDU while in the state "grouping" with the first element of the expected response list having one of the values "F-SELECT-response", "F-CREATE-response" or "F-CLOSE response", the entity shall:

- a) issue an F-READ-ATTRIB indication to the IFS user;

- b) add the value "F-READ-ATTRIB response" to the end of the expected responses list.

9.7.3 On receiving an F-READ-ATTRIB response service primitive from the IFS user while in state "f-read attribute pending", the entity shall:

- a) add an F-READ-ATTRIB response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "selected".

9.7.4 On receiving an F-READ-ATTRIB response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-READ-ATTRIB response", the entity shall:

- a) add an F-READ-ATTRIB response PDU as a data value to the current PSDU;
- b) delete the first element of the expected responses list.

9.8 Change attributes (Enhanced management)

9.8.1 On receiving an F-CHANGE-ATTRIB request PDU while in a "selected" state, the entity shall:

- a) issue an F-CHANGE-ATTRIB indication service primitive to the IFS user, with parameters derived from the data values received; enter the state "f-change attribute pending".

9.8.2 On receiving an F-CHANGE-ATTRIB request PDU while in the state "grouping" with the first element of the expected response list having one of the values "F-SELECT response", "F-CREATE response", "F-READ-ATTRIB response" or "F-CLOSE response", the entity shall:

- a) issue an F-CHANGE-ATTRIB indication service primitive;
- b) add the value "F-CHANGE-ATTRIB response" to the end of the expected responses list.

9.8.3 On receiving an F-CHANGE-ATTRIB response service primitive from the IFS user while in the state "f-change attribute pending", the entity shall:

- a) add an F-CHANGE-ATTRIB response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "selected".

9.8.4 On receiving an F-CHANGE-ATTRIB response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-CHANGE-ATTRIB response", the entity shall:

- a) add an F-CHANGE-ATTRIB response PDU to the current PSDU;
- b) delete the first element of the expected responses list.

9.9 File Open (Read, Write)

9.9.1 On receiving an F-OPEN request PDU while in a "selected" state, the entity shall:

- a) issue an F-OPEN indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-open pending".

9.9.2 On receiving an F-OPEN request PDU while in the state "grouping" with the last element of the expected responses list having one of the values: "F-SELECT response", "F-CREATE response", "F-READ-ATTRIB response" or "F-CHANGE-ATTRIB response" the entity shall:

- a) issue an F-OPEN indication to the IFS user;
- b) add the value "F-OPEN response" to the end of the expected responses list.

9.9.3 On receiving an F-OPEN response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-OPEN response", the entity shall:

- a) add an F-OPEN response PDU as a data value to the current PSDU;
- b) if the PDU did not carry a state result indicating failure set the value of the next state indicator to "data transfer idle";
- c) delete the first element of the expected responses list.

9.9.4 On receiving an F-OPEN response service primitive from the IFS user while in state "f-open pending", the entity shall:

- a) if
 - 1) the presentation context management functional unit has been successfully negotiated
 - and
 - 2) any presentation contexts in the defined context list are to be deleted from the defined context list (see notes below),
 then
 the associated abstract syntax names shall be included in the presentation context deletion list parameter of the P-ALTER-CONTEXT request (see (f) below), provided they are not required in the new open regime;
- b) if
 - 1) the presentation context management functional unit has been successfully negotiated
 - and
 - 2) a presentation context is to be added to the defined context list on the basis of the contents type name parameter for the successful establishment of the open regime
 then
 the associated abstract syntax names and presentation context identifications shall be included in the presentation context addition list parameter of the P-ALTER-CONTEXT request (see (f) below);
- c) add an F-OPEN response PDU as a data value to the current PSDU; if any necessary actions were identified in (a) or (b) to alter the presentation context, include a presentation action parameter, with value "true";
- d) terminate the current PSDU;
- e) if the state result parameter on the F-OPEN response primitive received indicates failure of the operation, enter state "selected" and do not execute (f);
- f) if the need to alter context was identified in (a) or (b), issue a P-ALTER-CONTEXT request presentation service primitive after any current PSDU has been terminated, and enter the state "presentation action pending"; otherwise enter the state "data transfer idle".

NOTES

- 1 If the contents type name is not present then the list of context definitions to be performed is empty.
- 2 The remove contexts parameter received on the F-OPEN request may be taken into consideration in the construction of the presentation context deletion list. The only relationship between the remove contexts parameter and the presentation context deletion list is implementation dependent.
- 3 None of the actions relating to the P-ALTER-CONTEXT request shall be selected if the result parameter on the primitive received indicated failure.

9.9.5 On receiving a P-ALTER-CONTEXT confirm primitive from the presentation service provider while in the state "presentation action pending", the entity shall enter the state "data transfer idle".

NOTE - If the P-ALTER-CONTEXT negotiation could not be carried out precisely as requested, the responder waits for the initiator to decide whether the resultant open regime can be used or not.

9.10 File close (Read, Write)

9.10.1 On receiving an F-CLOSE request PDU while in a "data transfer idle" state, the entity shall:

- a) issue an F-CLOSE indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-close pending".

NOTE - F-CLOSE request carries an action result that is used as a signalling mechanism between FERPM peers (see 19.1.3). No action result has any effect on the FPM.

9.10.2 On receiving an F-CLOSE request PDU while in the state "grouping" with the expected response list empty and the initial state indicator having value "data transfer idle", the entity shall:

- a) issue an F-CLOSE indication service primitive to the IFS user;
- b) add the value "F-CLOSE response" to the end of the expected responses list.

9.10.3 On receiving an F-CLOSE response service primitive from the IFS user while in an "f-close pending" state, the entity shall:

- a) add an F-CLOSE response PDU to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "selected".

9.10.4 On receiving an F-CLOSE response service primitive from the IFS user while in state "grouping" with the first element of the expected responses list having value "F-CLOSE response", the entity shall:

- a) add an F-CLOSE response PDU to the current PDU.
- b) set the value of the next state indicator to "selected";
- c) delete the first element of the expected responses list.

9.11 Begin Group (Grouping)

9.11.1 On receiving an F-BEGIN-GROUP request PDU while in either the state "initialized" or the state "data transfer idle", the entity shall:

- a) issue an F-BEGIN-GROUP indication to the IFS user;
- b) set the value of the initial state indicator to the current state;

- c) enter the state "grouping".

9.11.2 On receiving an F-BEGIN-GROUP response service primitive from the IFS user while in the state "group pending", the entity shall:

- a) add an F-BEGIN-GROUP response PDU to the current PSDU;
- b) enter the state "grouping".

9.12 End Group (Grouping)

9.12.1 On receiving an F-END-GROUP request PDU while in the state "grouping", the entity shall:

- a) issue an F-END-GROUP indication to the IFS user;
- b) enter the state "group pending".

9.12.2 On receiving an F-END-GROUP response service primitive from the IFS user while in state "grouping", the entity shall:

- a) add an F-END-GROUP response PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) if the requested threshold has been attained by counting consecutive primitives with either state result parameters indicating success or no state result parameter and
 - 1) the p-alter indicator is set then issue a P-ALTER-CONTEXT request with the contexts to be defined or deleted as determined in 9.9.4, then enter the state "presentation alter context pending".
 - 2) the p-alter indicator is unset then enter the state named by the next state indicator;
- d) if the requested threshold has not been attained by counting consecutive primitives with either state result parameters indicating success or no state result parameter, enter the state named by the initial state indicator; and clear the expected responses list, unset the next state indicator and unset the initial state indicator.

9.13 Recovery (Recovery)

9.13.1 On receiving an F-RECOVER request PDU while in a "initialized" state, the entity shall:

- a) issue an F-RECOVER indication service primitive to the IFS user, with parameters derived from the data values received;
- b) enter the state "f-recover pending".

9.13.2 On receiving an F-RECOVER response service primitive from the IFS user while in state "f-recover pending", the entity shall:

- a) if
 - 1) the presentation context management functional unit has been successfully negotiated and
 - 2) any presentation contexts in the defined context set are to be deleted from the defined context set (see notes below),
 then
 the associated presentation context identifications shall be included in the presentation context deletion list parameter of the P-ALTER-CONTEXT request (see (f))

below), provided they are not required in the new open regime;

b) if

1) the presentation context management functional unit has been successfully negotiated and

2) a presentation context is to be added to the defined context set on the basis of the contents type name parameter, for the successful establishment of the open regime.

then

the associated presentation context identifications shall be included in the presentation context addition parameter of the P-ALTER-CONTEXT request (see (f) below);

c) add an F-RECOVER response PDU as a data value to the current PSDU; If any necessary actions were identified in (a) or (b) to alter the presentation context, include a presentation action parameter, with value "true";

d) terminate the current PSDU;

e) if the state result parameter on the F-RECOVER response primitive received indicates failure of the operation, enter state "initialized" and do not execute (f);

f) if the need to alter context was identified in (a) or (b), issue a P-ALTER-CONTEXT request presentation service primitive after any current PSDU has been terminated, and enter the state "presentation action pending", otherwise enter the state "data transfer idle recover".

NOTES

1 If the contents type name is not present then the list of context definitions to be performed is empty.

2 The remove contexts parameter received on the F-RECOVER request may be taken into consideration in the construction of the presentation context deletion list. The only relationship between the remove contexts parameter and the presentation context deletion list is implementation dependent.

3 None of these actions shall be selected if the result parameter on the primitive received indicated failure.

9.13.3 On receiving a P-ALTER-CONTEXT confirm primitive from the presentation service provider while in the state "presentation action pending", the entity shall enter the state "data transfer idle recover".

NOTE - If the P-ALTER-CONTEXT negotiation could not be carried out precisely as requested, the responder waits for the initiator to decide whether the resultant open regime can be used or not.

9.14 Locate (Access)

9.14.1 On receiving an F-LOCATE request PDU while in an "data transfer idle" state, the entity shall:

a) issue an F-LOCATE indication service primitive to the IFS user, with parameters derived from the data values received;

b) enter the state "f-locate pending".

9.14.2 On receiving an F-LOCATE response service primitive from the IFS user while in the state "f-locate pending", the entity shall:

a) add an F-LOCATE response PDU as a data value to the current PSDU;

b) terminate the current PSDU;

c) enter the state "data transfer idle".

9.15 Erase (Access)

9.15.1 On receiving an F-ERASE request PDU while in an "data transfer idle" state, the entity shall:

a) issue an F-ERASE indication service primitive to the IFS user, with parameters derived from the data values received;

b) enter the state "f-erase pending".

9.15.2 On receiving an F-ERASE response service primitive from the IFS user while in the state "f-erase pending", the entity shall:

a) add an F-ERASE response PDU as a data value to the current PSDU;

b) terminate the current PSDU;

c) enter the state "data transfer idle".

10 File general actions

10.1 FTAM regime termination (abrupt)

10.1.1 On receiving an A-ABORT indication while in any state the entity shall examine the user information and evaluate it as one of the types listed below and then take the corresponding actions listed in 10.1.1.1, 10.1.1.2 and 10.1.1.3. The user information parameter of an A-ABORT indication received by an FTAM entity may be of three forms:

- a) an F-U-ABORT PDU (see 10.1.1.1),
- b) an F-P-ABORT PDU (see 10.1.1.2) or
- c) absent (see 10.1.1.3).

10.1.1.1 If the user information is an F-U-ABORT PDU, the entity shall:

- a) issue an F-U-ABORT indication service primitive to the IFS user, with parameters derived from the data values received;
- b) thereafter cease to exist.

10.1.1.2 If the user information is an F-P-ABORT PDU, the entity shall:

- a) issue an F-P-ABORT indication service primitive to the IFS user, with parameters derived from the data values received;
- b) thereafter cease to exist.

10.1.1.3 If the user information is absent, the entity shall:

- a) issue an F-P-ABORT indication service primitive to the IFS user, with an action result indicating a transient error.
- b) thereafter cease to exist.

NOTE - This may result in a permanent error indication being converted to a temporary error indication.

10.1.2 On receiving an A-P-ABORT indication while in any state, the entity shall:

- a) issue an F-P-ABORT indication service primitive to the IFS user, with a diagnostic parameter indicating a lower layer failure.
- b) thereafter cease to exist.

10.1.3 On receiving an F-U-ABORT request service primitive from the IFS user while in any state, the entity shall:

- a) on the basis of local information regarding the ability of the underlying services to convey unlimited user information by A-ABORT:
 - 1) if the ability exists, construct an F-U-ABORT PDU with values derived from the parameters received and send the PDU as user information on an A-ABORT request service primitive.
 - 2) if the ability does not exist, issue an A-ABORT with no user information.
- b) thereafter cease to exist.

10.2 Protocol errors

On detecting a protocol error (see 6.1) while in any state, the entity shall:

- a) issue an F-P-ABORT indication to the IFS user, with an action result parameter indicating a protocol error.
- b) on the basis of local information regarding the ability of the underlying services to convey unlimited user information by A-ABORT:
 - 1) if the ability exists, construct an F-P-ABORT PDU with a result indicating a protocol error and send the PDU as user information on an A-ABORT request service primitive.
 - 2) if the ability does not exist, issue an A-ABORT request with no user information.
- c) thereafter cease to exist.

Section three: The basic bulk data transfer protocol

11 State of bulk data transfer activity

NOTE - The entities referred to in this section are basic protocol entities.

11.1 State of the connection endpoint

For each PDU that may be received, the specification enumerates the states of the protocol in which the receiving entity can receive the PDU. If the action specified on receipt of a stimulus would require the issue of a service primitive which could not validly be issued in the current state, the entity concerned shall indicate a protocol error (see 10.2).

11.2 Additional state of the entities

The following sub-clauses define the items of state information associated with the basic protocol entities for the purposes of bulk data transfer.

11.2.1 Read/write indicator

The read/write indicator records whether the current bulk data transfer is to or from the initiator. The value is set upon the beginning of the data transfer regime. The defined values are "reading", "writing" and "unset". The initial value is unset.

11.2.2 Discard Indicator

The discard indicator is used to signal that data received during cancellation or before recovery is invalid and should be thrown away. If the recovery or restart functional units are selected and presentation resynchronization functional unit has been successfully negotiated, it is used in conjunction with session resynchronization to produce a recovery without user visibility of the error; otherwise it is used during the cancel phase. The defined values are "unset" and "set". Initially, the value is "unset".

11.2.3 Checkpoint identifier expected

The checkpoint identifier expected reflects the sequence of checkpoints within bulk data, and is reset by the start of bulk data transfer and by error recovery mechanisms. The value is incremented when a checkpoint is made. The value is an integer in the range 1 to 999998. Initially, the value is determined by the state of the association.

The checkpoint identifier expected applies only to FTAM regimes for which the use of the restart and/or recovery functional units have been successfully negotiated.

11.2.4 Next synchronization point number

The next synchronization point number reflects the sequence of events in the supporting synchronization services. The number is the serial number of the next session synchronization point to be issued by the session

Table 5 — Protocol Data Units

Name	Carried by	Functional units
F-READ request	P-DATA	read
F-WRITE request	P-DATA	write
data value (see note 1)	P-DATA	read, write
F-DATA-END request	P-DATA	read, write
F-TRANSFER-END request	P-DATA	read, write
F-TRANSFER-END response	P-DATA	read, write
F-CANCEL request	P-RESYNCHRONIZE (abandon) request user data or P-DATA	read, write
F-CANCEL response	P-RESYNCHRONIZE (abandon) request user data or P-DATA	read, write
F-CHECK request	(see note 2)	recovery, restart
F-CHECK response	(see note 2)	recovery, restart
F-RESTART request	P-RESYNCHRONIZE (restart) request user data	restart
F-RESTART response	P-RESYNCHRONIZE (restart) response user data	restart

NOTES

1 The data value corresponds to an F-DATA request service primitive. There is no F-DATA request PDU as such.

2 The F-CHECK request and response primitives are mapped directly onto the P-SYNC-MINOR request and response primitives. There is no additional syntax, and therefore no PDU definition.

service provider. The value is an integer in the range 0 to 999998. Initially on a newly created session connection, the value is 1.

The next synchronization point number applies only to FTAM regimes for which the use of the presentation minor synchronization functional unit has been successfully negotiated.

11.2.5 Synchronization Offset

The synchronization offset is a constant established when a read or write bulk data transfer is initiated or recovered, which gives the difference between the checkpoint identifier expected and the next resynchronization point number.

The synchronization offset applies only to FTAM regimes for which the use of the presentation minor synchronization functional unit has been successfully negotiated.

NOTE - Except during the issue of a checkpoint, or performance of the restart procedure the value of the synchronization offset is equal to the difference between the expected checkpoint number and the next synchronization point number.

11.2.6 Outstanding checkpoint counter

The outstanding checkpoint counter records the number of checkpoints which are unacknowledged.

12 Bulk data transfer protocol data units

The PDUs are transmitted as user data on a Presentation service primitive. The possible means of transmission of each PDU are listed in table 5.

Any valid bulk data transfer sequence of data values and PDUs carried on P-DATA may be concatenated into a single PSDU, provided no checkpoints need to be inserted in that sequence. The receiver of a PSDU unpacks it into individual data values and/or PDUs, and processes each according to the procedures specified in clauses 13 to 17.

13 Bulk data transfer initiating entity actions

13.1 Read (Read)

13.1.1 On receiving an F-READ request service primitive from the IFS user the entity shall:

- a) send an F-READ request PDU as a P-DATA request;
- b) if the presentation minor synchronization functional unit has been successfully negotiated, issue P-TOKEN-GIVE request for the synchronize minor token;
- c) increment the bulk data transfer number;
- d) set the checkpoint identifier expected so that
 - 1) if the open regime was established by a regime recovery service exchange then the value is the value negotiated plus one;
 - 2) if the open regime was not established by a regime recovery service exchange then the value is one;
- e) set the outstanding checkpoint counter to 0;
- f) if the presentation minor synchronization functional unit has been successfully negotiated, calculate the synchronization offset; this is equal to the next synchronization point number, minus the checkpoint identifier expected, plus one;

g) set the read/write indicator to "reading";

h) if the presentation minor synchronization functional unit has been successfully negotiated, enter the state "first minor synch indication pending"; otherwise enter the state "read data transfer" in the role of receiver.

13.1.2 If a P-SYNC-MINOR indication is received from the presentation service provider while in the state "first minor synch indication pending", the entity shall:

- a) issue a P-SYNC-MINOR response to the presentation service provider;
- b) increment the next synchronization point serial number;
- c) enter the state "read data transfer" in the role of receiver.

13.2 Write (Write)

13.2.1 On receiving an F-WRITE request service primitive from the IFS user while in the state "data transfer idle", the entity shall:

- a) send an F-WRITE request PDU as a P-DATA request;
- b) increment the bulk data transfer number;
- c) set the checkpoint identifier expected so that
 - 1) if the open regime was established by a regime recovery service exchange then the value is the value negotiated plus one;
 - 2) if the open regime was not established by a regime recovery service exchange then the value is one;
- d) if the presentation minor synchronization functional unit has been successfully negotiated, calculate the synchronization offset; this is equal to the next synchronization point number, minus the checkpoint identifier expected, plus one;
- e) set the outstanding checkpoint counter to 0;
- f) if the presentation minor synchronization functional unit has been successfully negotiated, issue a P-SYNC-MINOR (explicit) request primitive to the presentation service provider;
- g) if the presentation minor synchronization functional unit has been successfully negotiated, increment the next synchronization point number;
- h) set the read/write indicator to "writing";
- i) if the presentation minor synchronization functional unit has been successfully negotiated, enter the state "first minor synch indication pending"; otherwise enter the state "write data transfer" in the role of sender.

13.3 Transfer end (Read, write)

13.3.1 On receiving an F-TRANSFER-END request service primitive from the IFS user while in the state "read ending" or state "write ending", the entity shall:

- a) add an F-TRANSFER-END request PDU as a data value to the current PSDU;
- b) terminate the current PSDU;
- c) enter the state "transfer ending read" or "transfer ending write", depending on the value of the read/write indicator.

13.3.2 On receiving an F-TRANSFER-END response PDU while in the state "transfer ending write" the entity shall:

- a) issue an F-TRANSFER-END confirm service primitive to the IFS user;
- b) enter the state "data transfer idle".

13.3.3 On receiving an F-TRANSFER-END response PDU while in the state "transfer ending read" the entity shall:

- a) if the presentation minor synchronization functional unit has not been successfully negotiated, then issue an F-TRANSFER-END confirm service primitive to the IFS user;
- b) if the presentation minor synchronization functional unit has been successfully negotiated, enter the state "p-token pending"; if it has not, enter the state "data transfer idle".

13.3.4 On receiving a P-TOKEN-GIVE indication for the sync-minor token while in the state "p-token pending", the entity shall issue an F-TRANSFER-END confirm service primitive to the IFS user and enter the state "data transfer idle".

14 Bulk data transfer responding entity actions

14.1 Read (Read)

14.1.1 On receiving an F-READ request PDU while in an "data transfer idle" state the entity shall:

- a) issue an F-READ indication service primitive to the IFS user, with parameters derived from the data items received;
- b) increment the bulk data transfer number;
- c) set the checkpoint identifier expected so that
 - 1) if the open regime was established by a regime recovery service exchange then the value is the value negotiated plus one;
 - 2) if the open regime was not established by a regime recovery service exchange then the value is one;
- d) set the read/write indicator to "reading";
- e) set the outstanding checkpoint counter to 0;
- f) if the presentation minor synchronization functional unit has been successfully negotiated, calculate the synchronization offset; this is equal to the next synchronization point number, minus the checkpoint identifier expected, plus one;
- g) if the presentation minor synchronization functional unit has been successfully negotiated, enter the state "synch minor token pending"; if it has not, enter the state "read data transfer" in the role of sender.

14.1.2 On receiving a P-TOKEN-GIVE indication for the synchronize minor token while in the state "token pending", the entity shall:

- a) issue a P-SYNC-MINOR request to the presentation service provider;
- b) increment the next synchronization point number;
- c) enter the state "first synch minor confirm pending" in the role of sender.

14.2 Write (Write)

14.2.1 On receiving an F-WRITE request PDU while in an "data transfer idle" state, the entity shall:

- a) issue an F-WRITE indication service primitive to the IFS user, with parameters derived from the data items received;
- b) increment the bulk data transfer number;
- c) set the checkpoint identifier expected so that
 - 1) if the open regime was established by a regime recovery service exchange then the value is the value negotiated plus one;
 - 2) if the open regime was not established by a regime recovery service exchange then the value is one;
- d) set the read/write indicator to "writing"
- e) set the outstanding checkpoint counter to 0;
- f) if the presentation minor synchronization functional unit has been successfully negotiated, calculate the synchronization offset; this is equal to the next synchronization point number, minus the checkpoint identifier expected, plus one;
- g) if the presentation minor synchronization functional unit has been successfully negotiated, enter the state "first synch minor indication pending"; if it has not, enter the state "write data transfer" in the role of receiver.

14.2.2 On receiving a P-SYNC-MINOR indication while in the state "write minor synch pending", the entity shall:

- a) issue a P-SYNC-MINOR response;
- b) increment the next synchronization point number;
- c) enter the state "write data transfer" in the role of receiver.

14.3 Transfer end (Read, write)

14.3.1 On receiving an F-TRANSFER-END request PDU while in the "read ending" or "write ending" state, the entity shall:

- a) issue an F-TRANSFER-END indication service primitive to the IFS user, with parameters derived from the data items received;
- b) enter the state "transfer ending read" or "transfer ending write", depending on the value of the read/write indicator.

14.3.2 On receiving an F-TRANSFER-END response service primitive from the IFS user, the entity shall:

- a) send an F-TRANSFER-END response PDU as a P-DATA request;
- b) if the read/write indicator is set to read and the presentation minor synchronize functional unit has been successfully negotiated, issue a P-TOKEN-GIVE request service primitive to the presentation service provider;
- c) unset the read/write indicator;
- d) enter the state "data transfer idle".

14.3.3 On receiving an F-TRANSFER-END request PDU while in the state "cancel pending", the entity shall do nothing.

15 Bulk data transfer sending entity actions

15.1 Send data (Read, write)

15.1.1 On receiving an F-DATA request primitive from the IFS user while in the state "read data transfer", the state "write data transfer" or the state "first synch minor confirm pending", the sending entity shall

- a) add the data element given on the F-DATA request to the current PSDU;
- b) check that any data item sent is consistent with any context within the defined context set, and indicate a protocol error if it is not.

Any sequence of data elements given on the F-DATA request primitives, i.e. one or several data elements may be concatenated into a single PSDU, provided no checkpoints need to be inserted in that sequence. In any specific instance of the FTAM protocol the sequence of data elements is determined by local system concerns.

15.1.2 On receiving a P-SYNCH-MINOR confirm while in the state "first synch minor confirm pending", the sending entity shall

- a) if an F-CANCEL request service primitive is being preserved, enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator, and then immediately perform the actions in 17.2.1;
- b) if an F-RESTART request service primitive is being preserved, enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator, and then immediately perform the actions in 17.3.1;
- c) enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator.

15.2 End of data transfer (Read, write)

15.2.1 On receiving an F-DATA-END request service primitive from the IFS user while in the state "read data transfer", the state "write data transfer" or the state "first synch minor confirm pending", the entity shall:

- a) add an F-DATA-END request PDU as a data value to the current PSDU;
- b) if it is the responding entity then terminate the current PSDU;
- c) if the state was "first synch minor confirm pending", enter the state "first synch minor confirm pending at data end";
- d) enter the state "read ending" or "write ending" depending on the value of the read/write indicator.

15.2.2 On receiving a P-SYNCH-MINOR confirm while in the state "first synch minor confirm pending at data end", the sending entity shall enter the state "read ending" or "write ending" depending on the value of the read/write indicator.

15.3 Checkpoint insertion (Read, write)

15.3.1 On receiving an F-CHECK request service primitive from the IFS user while in the state "read data transfer", the state "write data transfer" or the state "first synch minor confirm pending", the entity shall:

- a) terminate the current PSDU;
- b) confirm that the checkpoint identifier given is equal to the checkpoint identifier expected;
- c) increment the outstanding checkpoint counter;
- d) verify that the checkpoint window has not been exceeded;
- e) issue a P-SYNC-MINOR request with the type parameter set to 'optional'. Confirm that the synchronization point serial number received from the Presentation service provider is equal to the next synchronization point serial number;
- f) increment the expected checkpoint identifier and the next synchronization point serial number; if the next synchronization point serial number exceeds 999 999 then a protocol error has occurred;
- g) if any of the checks in (b), (c), (d), (e) or (f) above fail, then indicate a type II error locally.

15.3.2 On receiving P-SYNC-MINOR confirm while in one of the "read data transfer", "write data transfer", "read ending", "write ending" or "write transfer ending" states the entity shall:

- a) issue an F-CHECK confirm service primitive to the IFS user, with parameters derived from the information received. The checkpoint identifier is calculated by subtracting the synchronization offset from the synchronization point serial number.
- b) decrement the outstanding checkpoint counter by the difference between the the checkpoint identifier expected minus the synch point serial number received minus the offset.

16 Bulk data transfer receiving entity actions

16.1 Data transfer (Read, write)

16.1.1 On receiving a data element in a context other than FTAM PCI context while in a "read data transfer" or a "write data transfer" state the entity shall issue an F-DATA indication service primitive to the IFS user.

NOTE - This includes receipt of any data element of type File-Contents-Data-Element (see ISO 8571-2 for the definition of an abstract syntax of the access structure of an hierarchical file).

16.2 End of data transfer (Read, write)

16.2.1 On receiving an F-DATA-END request PDU while in the state "read data transfer" or the state "write data transfer" the entity shall:

- a) issue an F-DATA-END indication service primitive to the IFS user;
- b) enter the state "read ending" or "writing ending" depending on the value of the "read/write indicator";

16.2.2 On receiving an F-DATA-END request PDU while in the state "cancel pending" the entity shall do nothing.

16.3 Checkpoint reception (Read, write)

16.3.1 On receiving a P-SYNC-MINOR indication while in a "read data transfer" or a "write data transfer" state with checkpoints outstanding the entity shall: