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

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Reference number
ISO 8571-4 : 1988 (E)

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

© International Organization for Standardization, 1988 ●

Printed in Switzerland

Contents

Section one: General

Section two: Basic file protocol

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Section three: The basic bulk data transfer protocol

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Section four: The error recovery protocol

Section five: Abstract Syntax

Section six: Conformance

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Annexes

Figures

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Tables

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

This page intentionally left blank

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

INTERNATIONAL STANDARD ISO 8571-4 : 1988 (E)

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

Part 4 : File Protocol Specification

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

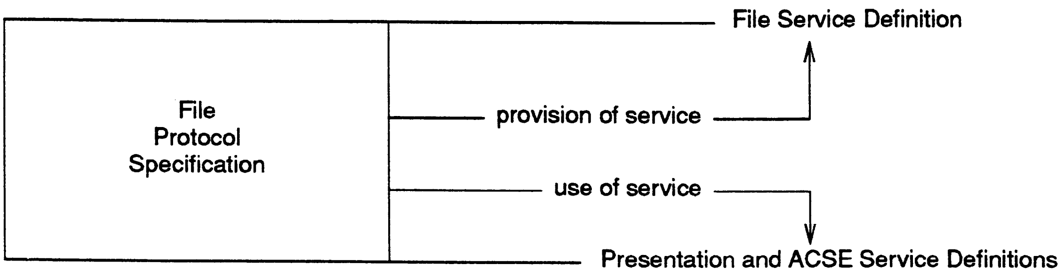
File Service Definition
File Protocol Specification
provision of service
use of service
Presentation and ACSE Service Definitions

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

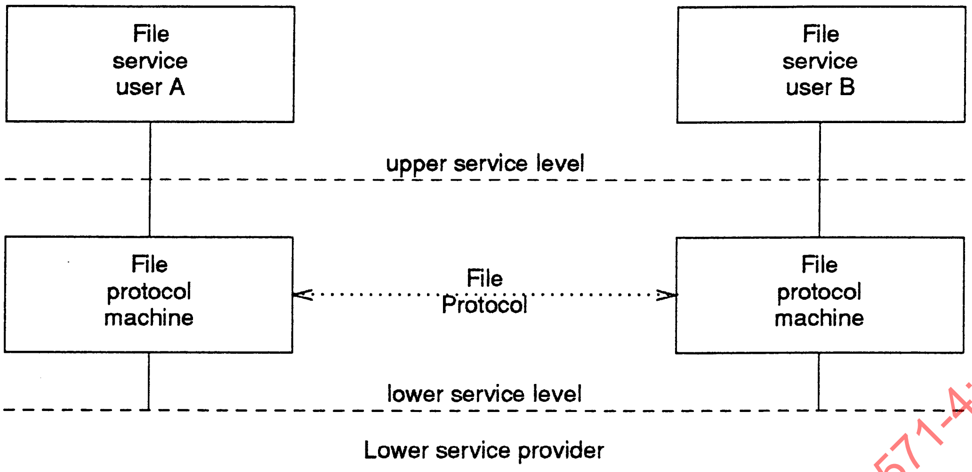

Figure 2 — File Protocol Machine model

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

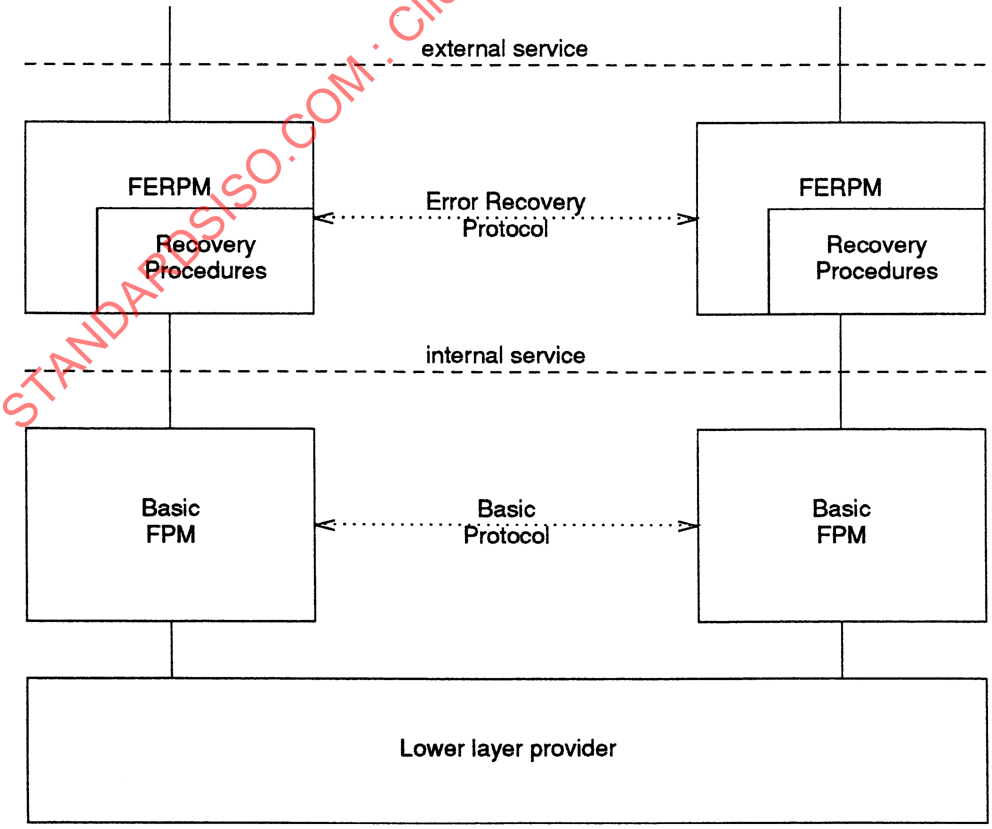

Figure 3 — Extended File Protocol Machine model

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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".

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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 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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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 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;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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".

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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)

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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".

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

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:

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

a) verify that the checkpoint window has not been exceeded;

b) increment the outstanding checkpoint counter;

c) check that the synchronization point serial number received equals the next synchronization point serial number;

d) check that the expected checkpoint identifier equals the synchronization point serial number received minus the synchronization offset;

e) If any of the checks in (a), (b) or (c) fails, indicate a protocol error;

f) issue an F-CHECK indication service primitive to the IFS user, with checkpoint id parameters equal to the synchronization point serial number received minus the synchronization offset;

g) increment the checkpoint identifier expected and the next synchronization point serial number.

16.3.2 On receiving an F-CHECK response service primitive from the IFS user, while in a "read data transfer", "write data transfer", "read ending" or "write ending" state, or, if the entity is the responder also in the a "transfer ending write" state the entity shall;

a) issue a P-SYNC-MINOR response with parameter equal to the checkpoint identifier from the service primitive plus the synchronization offset.

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.

17 Bulk data transfer general actions

17.1 Discarding (Read, Write)

17.1.1 If an F-CANCEL request PDU is received or transmitted, either on a P-RESYNCHRONIZE primitive or a P-DATA primitive, or if an F-RESTART request PDU is transmitted on a P-RESYNCHRONIZE primitive, the entity shall set the discard indicator.

17.1.2 On receiving a P-DATA indication which is not in the FTAM-PCI-context that is being used for communication between the FPMs while in a "read data transfer" or a "write data transfer" state with the discard indicator set, the entity shall do nothing.

17.2 Cancel (Read, write)

17.2.1 On receiving an F-CANCEL request service primitive from the IFS user while in the "read data transfer" or the "write data transfer state", the entity shall

a) if the presentation resynchronization functional unit has been successfully negotiated, then issue a P-RESYNCHRONIZE (abandon) request with an F-CANCEL request PDU as user data and set the discard indicator; the P-RESYNCHRONIZE (abandon) request shall convey an F-CANCEL request PDU as user data. Otherwise issue an F-CANCEL request PDU using P-DATA;

b) if the presentation minor synchronization functional unit has been successfully negotiated, and the issuer of the F-CANCEL is the responder, return the sync-minor token to the initiator;

c) set the outstanding checkpoint counter to zero;

d) enter the state cancel response pending.

17.2.2 On receiving an F-CANCEL request service primitive while in the "first synch minor confirm pending" state, the entity shall preserve the service primitive for further processing.

17.2.3 On receiving a P-RESYNCHRONIZE (abandon) indication with user data containing an F-CANCEL request PDU while in the "read data transfer" or the "write data transfer" state, the entity shall:

a) discard any user data which it has not yet delivered, and continue to receive PDUs;

b) issue an F-CANCEL indication service primitive to the IFS user;

c) set the outstanding checkpoint counter to zero;

d) enter the state "f-cancel response pending".

17.2.4 On receiving an F-CANCEL request PDU on a P-DATA indication while in the "read data transfer" or the "write data transfer" state, the entity shall

a) discard any user data which it has not yet delivered, and continue to receive PDUs;

b) issue an F-CANCEL indication service primitive to the IFS user;

c) set the outstanding checkpoint counter to zero;

d) enter the state "f-cancel response pending".

17.2.5 On receiving an F-CANCEL response service primitive from the IFS user, while in the state "f-cancel response pending", the entity shall:

a) if presentation resynchronization functional unit has been successfully negotiated, issue a P-RESYNCHRONIZE (abandon) response with user data containing an F-CANCEL response PDU. Otherwise issue an F-CANCEL response as user data on a P-DATA request;

b) set the checkpoint identifier expected and the expected synchronization number to their values at the beginning of transfer;

c) unset the discard indicator;

d) enter the state "data transfer idle".

17.2.6 On issuing an F-CANCEL confirm service primitive to the IFS user, or on sending an F-CANCEL response PDU, the entity shall unset the discard indicator.

17.2.7 On receiving a P-RESYNCHRONIZE (abandon) confirm with user data containing an F-CANCEL response PDU, while in the "cancel response pending" state, the entity shall

a) issue an F-CANCEL confirm service primitive to the IFS user, with parameters derived from the information received;

b) enter the state "data transfer idle".

17.2.8 On receiving an F-CANCEL response PDU on a P-DATA indication while in the "cancel response pending" state, the entity shall;

a) issue an F-CANCEL confirm service primitive to the IFS user, with parameters derived from the information received;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

b) enter the state "data transfer idle".

17.2.9 Collision of cancel requests carried on P-RESYNCHRONIZE primitives may occur. On receiving a P-RESYNCHRONIZE (abandon) indication with user data containing an F-CANCEL request PDU while in the "cancel response pending" state, the entity shall;

a) issue an F-CANCEL confirm service primitive to the IFS user;

b) issue a P-RESYNCHRONIZE (abandon) response with user data containing an F-CANCEL response PDU and diagnostic from the F-CANCEL request previously issued;

c) enter the state "data transfer idle".

17.2.10 If the presentation resynchronization functional unit has not been successfully negotiated, then collision of cancel requests carried on P-DATA primitives may occur. If an F-CANCEL request PDU is received on a P-DATA indication while in the state "cancel response pending", the entity shall;

a) issue an F-CANCEL confirm service primitive to the IFS user.

b) enter the state "data transfer idle".

17.3 Restarting (Restart data transfer)

17.3.1 On receiving an F-RESTART request service primitive from the IFS user, the entity shall;

a) issue a P-RESYNCHRONIZE (restart) request and set the discard indicator. The user data shall contain an F-RESTART request PDU including a checkpoint identifier equal to the checkpoint id in the F-RESTART primitive, and the parameters shall indicate return of the sync-minor token to the sender. The synchronization point number proposed to the Presentation Service provider shall be equal to the next resynchronization point serial number minus one;

b) set the outstanding checkpoint counter to zero;

c) enter the state "restart response pending".

17.3.2 On receiving a P-RESYNCHRONIZE (restart) indication with user data containing an F-RESTART request PDU while in a "read data transfer" or "write data transfer" state, the entity shall;

a) discard any user data which it has not yet delivered, and continue to receive PDUs;

b) issue an F-RESTART indication service primitive to the IFS user. The checkpoint id shall be equal to the value received in the PDU;

c) set the outstanding checkpoint counter to zero;

d) enter the state "f-restart response pending".

17.3.3 On receiving an F-RESTART response service primitive from the IFS user, the entity shall;

a) issue a P-RESYNCHRONIZE (restart) response with user data containing an F-RESTART response PDU; include a checkpoint id parameter equal to the checkpoint id in the F-RESTART primitive;

b) set the checkpoint identifier expected equal to the checkpoint id received plus one;

c) set the next synchronization point number equal to the resynchronization point number negotiated with the Presentation Service provider;

d) set the synchronization offset to the next synchronization point number minus the checkpoint identifier expected;

e) enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator.

17.3.4 On receiving a P-RESYNCHRONIZE (restart) confirm, with user data containing an F-RESTART response PDU, the entity shall;

a) unset the discard indicator;

b) issue an F-RESTART confirm service primitive to the IFS user, with parameters derived from the information received in the PDU;

c) set the checkpoint identifier expected equal to the checkpoint id received plus one;

d) set the next synchronization point number equal to the resynchronization point number negotiated with the Presentation Service provider;

e) set the synchronization offset to the next synchronization point number minus the checkpoint identifier expected;

f) enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator.

17.3.5 On receiving a P-RESYNCHRONIZE (restart) indication with user data containing an F-RESTART request PDU while in "restart response pending" state, the entity shall;

a) unset the discard indicator;

b) issue an F-RESTART confirm service primitive to the IFS user. If the entity is the sender, the checkpoint id shall be equal to the value received on the PDU. If the entity is the receiver, the value shall be equal to the value previously transmitted in the F-RESTART request PDU;

c) issue a P-RESYNCHRONIZE (restart) response to the presentation service provider, with user data containing an F-RESTART response PDU with checkpoint id equal to the value determined in (a).

d) set the checkpoint identifier expected equal to the checkpoint id received plus one;

e) set the next synchronization point number equal to the resynchronization point number negotiated with the Presentation Service provider;

f) set the synchronization offset to the next synchronization point number minus the checkpoint identifier expected;

g) enter the state "read data transfer" or "write data transfer" depending on the value of the read/write indicator.

17.3.6 On receiving an F-RESTART request service primitive while in the "first synch minor confirm pending" state, the entity shall preserve the service primitive for further processing.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Section four: The error recovery protocol

18 Protocol mechanisms

NOTE - The entities referred to in section four are error recovery protocol entities.

18.1 Error detection and reporting

18.1.1 Classes of error

Errors are categorized as follows:

Class I errors	damage	the data transfer regime only.
Class II errors	damage	the select or open regimes.
Class III errors	lose	the association

18.1.2 Error detection mechanisms

Errors are detected either

a) by one of the entities which are operating the error recovery procedure, either as a result of events local to that entity or of consistency checks present in the data transferred; or

b) by the provider of the supporting service. Provider detected errors will be signalled to both the entities using the service by the F-P-ABORT primitive, but the two users will not necessarily be informed at the same time.

NOTES

1 If one entity detects an abnormal condition but is able to apply some correction locally without need for an exchange of protocol, that error is not considered visible within the OSI Environment.

2 Users of the file service may also detect errors and act upon them as appropriate but their resolution is not handled by the error recovery entities.

18.1.3 Error reporting

If the error is detected by the service provider, both service users will be informed and no additional error reporting will be necessary. However, if the error is detected by one of the error recovery entities, the class of error must be reported to the other. This is achieved as follows for transient errors:

a) for class I errors, the class of error is indicated by attempting to negotiate a restart point (see 19.2.1); if the presentation resynchronize functional unit is not available, the class I error is signalled as if it were a class II error;

b) for class II errors, the class of error is indicated by an action result parameter indicating failure, or a suitable optional diagnostic parameter in an F-CANCEL request protocol data unit;

c) for class III errors, the class of error is indicated by a local signal that conveys a transient abort condition to the FPM.

Permanent errors detected by a FERPM are signalled to the IFS user by issuing an F-P-ABORT indication Permanent errors detected by a FERPM are signalled to the peer FERPM by issuing a local signal that conveys a permanent abort condition to the FPM.

18.2 Checkpoints

Recovery makes use of the identification of points in the file to form checkpoints. These are points inserted into the data flow by the sender at appropriate points during the data transfer. The algorithm used by the sender to position checkpoints is a local matter, but on restart or recovery it implies the same point in the data flow as was originally marked by that checkpoint identifier.

Checkpoint acknowledgement is used to limit the number of points at which recovery may be requested, and so to limit commitment of resources. Acknowledgement of a checkpoint also acts as an acknowledgement of any unacknowledged checkpoints received before the checkpoint in question. Each entity states, in the F-INITIALIZE protocol data unit it sends, the maximum number of checkpoints it is prepared to have outstanding when acting as sender; the protocol will not progress if the other entity does not acknowledge checkpoints when that number are outstanding.

18.3 Negotiation of restart point

The responsibility for initiating a recovery action lies with the entity with the most complete information on the stage reached before the error. Thus

a) recovery from class I errors is requested by the receiver;

b) recovery from class II or III errors is requested by the initiator of the activity which failed.

For a class I error, either party may propose a restart; if the receiver makes the proposal, it includes a restart point in the request, and if the sender makes the proposal, the receiver includes a restart point in its response. If resynchronization is not available the error will have been signalled as a class II error, and the following procedure applies.

For a class II or III error, the protocol entity supporting the initiator makes a proposal when re-establishing the file regimes with an F-RECOVER primitive, and the protocol entity supporting the filestore provider may make an overriding proposal on the response if it was the receiver at the time of the error.

NOTE - A restart may imply some real time delay (due, for example, to media repositioning) in some OSIE/RSE mappings.

The time delay before recovery is attempted may be based on the suggested delay value from the diagnostic associated with the failure, if one is present.

18.4 Docket contents

Recovery from class II or III errors necessitates the preservation of a certain collection of information relating to the transfer. The information is associated with the external file service connection when restart or recovery functional units are selected, not the internal file service connection. This requires the management of the following information, referred to here as a docket. A docket makes the following information available;

a) an identifier for the activity; this identifier is not reused if there is a possibility of a resumption;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

b) for the responder, the identity and location of initiator, or for the initiator, the identity and location of responder;

c) a record of the type of the activity (including an indication of read or write if applicable);

d) the bulk data transfer number of the transfer in progress or completed;

e) an indication of the activity state, which may be "starting", "in progress", "data transfer finished" or "finished";

f) a list of checkpoint identifiers and the position in the data flow they correspond to;

g) a count of checkpoints which have not yet been acknowledged;

h) the presentation context for the file content, recorded at file open;

i) the access context for the file content, recorded at start of bulk data transfer;

j) a description of locking mechanisms currently in place for the data;

k) the current location in the file structure.

Use of the term docket does not imply that the information must, in practice, be packaged in a particular way, but is merely a descriptive convenience. However, the reliability depends on the preservation of this body of information, which must exhibit at least the reliability expected of the recovery procedure. An implementation cannot be expected to retain dockets indefinitely; it is a matter of local choice when this procedure is abandoned and dockets deleted. However, for as long as the docket is maintained, concurrency control locking any files recorded in it should be maintained.

19 Specification of the error control protocol

19.1 Normal operation

19.1.1 Issue of service primitives

For every request or response primitive received from the user of the external file service, an error recovery entity shall issue a primitive of the same type, including parameters identical to those received, to the internal file service provider. It may add parameters peculiar to the error control protocol.

For every indication or confirm primitive received from the internal file service provider which

a) is not concerned with error recovery (F-INITIALIZE, F-TERMINATE, F-SELECT, F-CREATE, F-DELETE, F-OPEN, F-LOCATE, F-ERASE, F-READ, F-WRITE or instances of F-DESELECT, F-CLOSE carrying an action result indicating transient error); or

b) carries an action result parameter indicating a permanent error;

an error recovery entity, which is not engaged in a recovery procedure, shall issue a primitive of the same type, including parameters identical to those received, to the external file service user. F-INITIALIZE and F-OPEN may carry parameters relevant to the error control protocol which are not made available to the external file service user.

Any primitive received from the FPM with an action result indicating transient error starts a recovery procedure in the FERPM.

An error recovery entity is defined to be in a recovery procedure from the moment an error is detected until it has negotiated a recovery point and re-sent or received any previously handled data following the recovery point.

19.1.2 FTAM Quality of Service Negotiation

19.1.2.1 Initiating FERPM roles

19.1.2.1.1 Requests

Upon receipt of an F-INITIALIZE request service primitive from the EFS user the entity shall

a) examine the FTAM quality of service parameter and evaluate it on the basis of its value, local knowledge and its capabilities;

b) determine which of the restart and recovery functional units are required to provide the FTAM quality of service that is closest to the requested FTAM quality of service; and

c) add the resulting functional units to the functional unit parameter.

19.1.2.1.2 Responses

Upon receipt of an F-INITIALIZE confirm from the FPM the entity shall

a) note the returned FTAM quality of service;

b) note the functional units negotiated;

c) delete the values "restart" and "recovery" from the functional units parameter if they are present; and

d) issue an F-INITIALIZE confirm service primitive to the EFS user with the FTAM quality of service parameter as modified, the functional units parameter as modified and all other parameters unchanged.

19.1.2.2 Responding FERPM roles

19.1.2.2.1 Indications

Upon receipt of an F-INITIALIZE indication service primitive from the FPM the entity shall

a) examine the FTAM quality of service parameter and evaluate and possible modify it on the basis of its value, local knowledge and its capabilities;

b) determine and remove those requested functional units which cannot be offered;

c) record if the restart or recovery functional units are both requested and available;

d) delete the restart and/or recovery functional units, if present, from the functional unit parameter; and

e) issue an F-INITIALIZE indication, without the restart or recovery functional units, to the EFS user.

19.1.2.2.2 Responses

Upon receipt of an F-INITIALIZE response from the EFS user the entity shall

a) note the returned FTAM quality of service;

b) add to the functional units parameter the restart or recovery functional units if they are recorded as required and available;

c) issue an F-INITIALIZE response service primitive to the IFS provider with the functional units parameter as modified and all other parameters unchanged.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

NOTES

1 There is no formal correspondence between the FQoS negotiated and the functional units finally negotiated. The functional units will depend in part on the perceived reliability of both local systems and known properties of the communications mechanisms in use.

2 A minimal FQoS requiring no error control at all does not imply that there is no FERPM, but only implies that the FERPM provides no error control.

19.1.3 Checkpoint insertion

The sending error recovery entity should insert F-CHECK requests into the sequence of F-DATA primitives it is called upon to transfer.

NOTE - The points at which these primitives are inserted will depend on local management strategies, based on the way the information is received and on the requested quality of service.

19.1.4 Checkpoint acknowledgement

The receiving FERPM shall issue F-CHECK response primitives before the number of outstanding checkpoints exceeds the sending FERPM's checkpoint window.

The sending FERPM shall suspend the issuing of F-DATA request and F-CHECK request primitives to the FPM when the number of outstanding checkpoints equals the sending FERPM's checkpoint window.

19.1.5 Docket management

In the course of a successful transfer, the initiator or responder shall perform the actions described in this clause.

19.1.5.1 After receiving an F-INITIALIZE request from the EFS user and before issuing an F-INITIALIZE request to the IFS provider the initiating FERPM shall retain for inclusion in any docket subsequently created the values of the parameters requested.

19.1.5.2 After receiving an F-INITIALIZE indication from the IFS provider and before issuing an F-INITIALIZE indication to the EFS user the responding FERPM shall retain for inclusion in any docket subsequently created the values of the parameters requested.

19.1.5.3 After receiving an F-INITIALIZE response from the EFS user and before issuing an F-INITIALIZE request to the IFS provider the responding FERPM shall retain for inclusion in any docket subsequently created the values of the parameters returned.

19.1.5.4 After receiving an F-INITIALIZE confirm from the IFS provider and before issuing an F-INITIALIZE confirm to the EFS user the initiating FERPM and shall retain for inclusion in any docket subsequently created the values of the parameters returned.

19.1.5.5 After receiving an F-OPEN request from the external file service user, and before issuing an F-OPEN request to the internal file service provider, the initiating error recovery entity shall create a docket and record in it the following information as defined in 18.4: the activity identifier, the location of the responder, the type of activity, the access context, the presentation context, a null checkpoint list and the state "starting".

19.1.5.6 After receiving an F-OPEN indication from the internal file service provider, and before issuing an F-OPEN indication to the external file service user, the responding error recovery entity shall create a docket and record in it the following information as defined in 18.4: the activity identifier, the location of the initiator, the type of activity, the access context, the presentation context, a null checkpoint list and the state "starting".

19.1.5.7 If immediately following the F-OPEN indication to the external file service user, there is an F-OPEN response with a state result parameter indicating failure from the external file service user, the responding error recovery entity shall delete its docket before issuing the corresponding request to the internal file service provider.

19.1.5.8 On receiving an F-OPEN confirm with a state result indicating an error, the initiator shall delete its docket.

19.1.5.9 On issuing a request or receiving an indication of an F-READ or an F-WRITE, an entity shall record the state as "in progress", and increment and record the bulk data transfer number.

19.1.5.10 The error recovery entity which is sending data shall, on issuing an F-CHECK request primitive, add to the list in its docket the checkpoint identifier in that primitive.

19.1.5.11 The error recovery entity which is receiving data shall, on receiving an F-CHECK indication primitive, add to the list in its docket the checkpoint identifier in that primitive.

19.1.5.12 The error recovery entity which is receiving data shall, on securing the data preceding any F-CHECK indication primitive received, issue an F-CHECK response primitive. If one action which secures data would result in the issue of more than one F-CHECK response, only the response with higher synchronization point number needs to be issued.

19.1.5.13 The error recovery entity which is receiving data shall, on issuing an F-CHECK response primitive, delete from the list in its docket any checkpoint identifiers earlier than the one in that primitive.

19.1.5.14 The error recovery entity which is sending data shall, on receiving an F-CHECK confirm primitive, delete from the list in its docket any checkpoint identifiers earlier than the one in that primitive.

19.1.5.15 The error recovery entities shall record "data transfer finished" in their dockets in the following circumstances:

initiator sending	F-TRANSFER-END confirm
initiator receiving	F-TRANSFER-END request
responder sending	F-TRANSFER-END indication
responder receiving	F-TRANSFER-END response

19.1.5.16 The initiating error recovery entity shall, on issuing an F-CLOSE request without an action result to the internal service provider, record the state "finished" in its docket;

19.1.5.17 If the action result parameter indicates success or permanent error, the responding error recovery entity shall delete its docket on issuing an F-CLOSE response to the internal service provider. Otherwise, the responding error recovery entity shall retain its docket and initiate recovery procedures.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

19.1.5.18 If the action result parameter indicates success or permanent error, the initiating error recovery entity shall delete its docket on receiving an F-CLOSE confirm from the internal file service provider. Otherwise, the initiating error recovery entity shall retain its docket and initiate recovery procedures.

19.2 Recovery from errors

The following clauses specify the error recovery procedures. These procedures are summarized in figures at the start of each clause.

19.2.1 Class I errors

19.2.1.1 If an error recovery entity detects a class I error (see 18.1), it shall issue an F-RESTART request to the internal service provider.

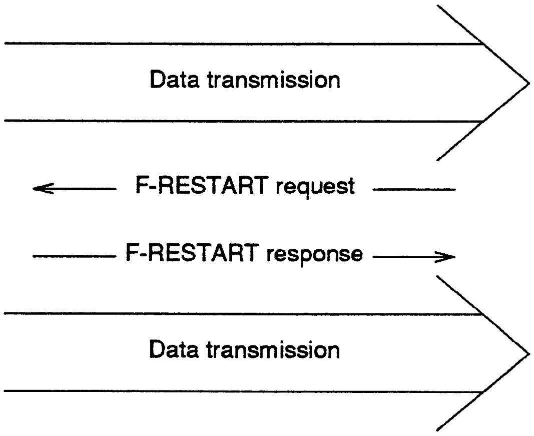

Figure 4 — Procedure for class I errors

19.2.1.2 If an error recovery entity detects a class I error (see 18.1) and the presentation resynchronize functional unit has been successfully negotiated and the restart functional unit has been successfully negotiated then it shall issue an F-RESTART request to the IFS provider. If the restart functional unit is not available, but the recover functional unit is available then it shall perform a recovery following the procedures described in 19.2.2.1. If the entity is a receiving entity, the F-RESTART request shall include a recovery point which is a checkpoint identifier such that:

a) if intermediate restart is not possible, it indicates restart of the activity from the beginning, otherwise

b) it is the most recent checkpoint from the list in the docket.

NOTE - The session synchronization point numbers are related to the checkpoint identifiers by the synchronization offset (see section three).

19.2.1.3 If the entity is a sending entity, it shall not include a recovery point parameter in the F-RESTART request.

19.2.1.4 If a sending error recovery entity receives an F-RESTART indication it shall send an F-RESTART response, and restart data transfer at the checkpoint quoted in the F-RESTART indication.

NOTE - No response should be expected in this case if an F-RESTART request had been issued.

19.2.1.5 If a sending error recovery entity receives an F-RESTART confirm it shall restart data transfer at the recovery point given in the confirm.

19.2.1.6 If a receiving error recovery entity receives an F-RESTART indication, and if it has not previously sent an F-RESTART request which remains unconfirmed, it shall issue an F-RESTART response, quoting a recovery point as in 19.2.1.2.

19.2.1.7 If a receiving error recovery entity receives an F-RESTART indication after sending an F-RESTART request and before receiving an F-RESTART confirm, it shall discard the primitive.

19.2.1.8 If the restart functional unit is not available then class I errors shall be handled as class II errors.

19.2.2 Class II errors

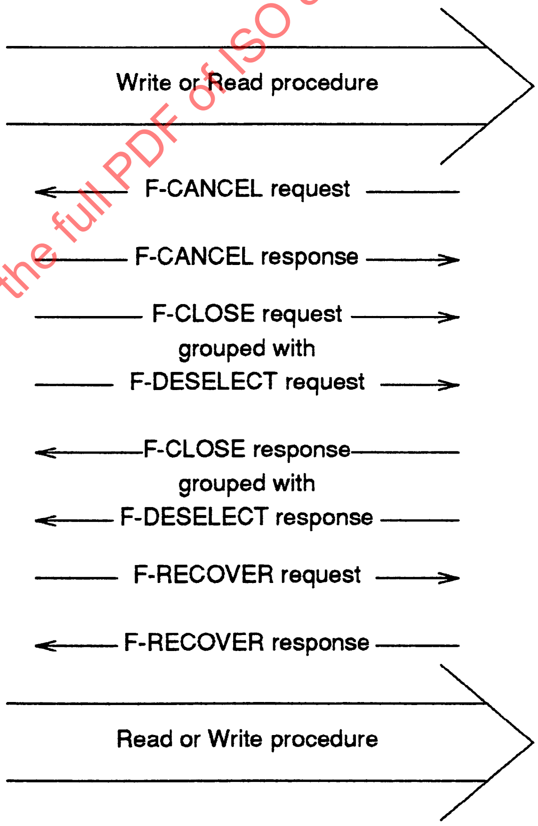

Figure 5 — Procedure for class II errors

NOTE - Figure 5 assumes that grouping of the F-CLOSE and F-DESELECT primitives is being used. This will always be the case in the file transfer service class.

19.2.2.1 If an error recovery entity detects a class II error (see 18.1) it shall issue an F-CANCEL request with an action result parameter indicating a transient (or recoverable) error with optional diagnostic.

19.2.2.2 If an error recovery entity receives an F-CANCEL indication, it shall issue an F-CANCEL response.

19.2.2.3 If an initiating error recovery entity receives an F-CANCEL indication with an action result parameter indicating a class II transient error, or an F-CANCEL confirmation after having signalled a class II transient error,

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

it should issue an F-CLOSE request and an F-DESELECT request, and on receiving the F-CLOSE confirm and F-DESELECT confirm it should perform any necessary corrective action. It should then perform an F-RECOVER recovery as specified in 19.2.3.2 and following.

19.2.3 Class III errors

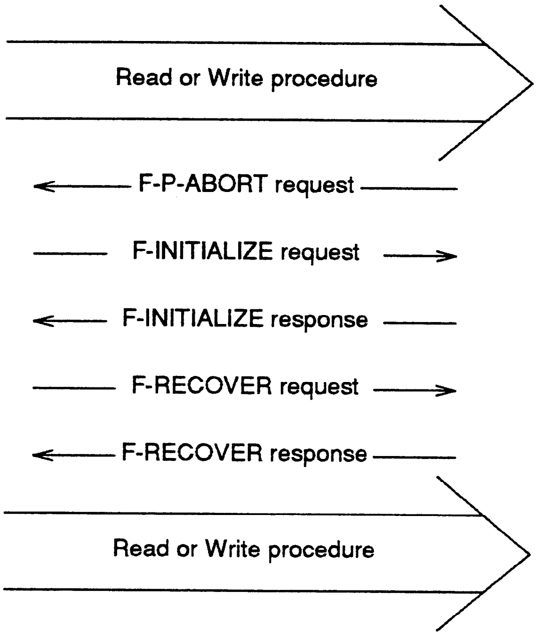

Figure 6 — Procedure for class III errors

19.2.3.1 If an initiating error recovery entity detects a class III error (by receiving an F-P-ABORT or by a local reinitialization) it shall establish an association to the location recorded in its docket by means of an exchange of F-INITIALIZE primitives using the parameter values recorded in its docket when the FTAM regime was first established. If a diagnostic was received with a recommended retry time field and the association cannot be re-established, the entity should delay for the period of time specified in the diagnostic received. If none is specified, it shall delay for a locally defined period of time and attempt the association establishment again.

NOTE - This procedure should be repeated a locally determined number of times before abandoning the activity and deleting the docket.

19.2.3.2 When the association is established, the initiating entity shall issue an F-RECOVER request primitive, quoting the activity identifier from its docket. It shall quote a recovery point and a bulk data transfer number. The checkpoint identifier quoted shall be:

a) if the entity was acting as a sending entity, the oldest checkpoint identifier from the list in its docket;

b) if the entity was acting as a receiving entity, the most recent checkpoint identifier from the list in its docket.

19.2.3.3 A responding error recovery entity which receives an F-RECOVER indication shall respond as follows:

a) if it has a docket for the same activity identifier, issued at the same location, and is prepared to continue the activity, it shall issue an F-RECOVER response. It shall include a recovery point which is:

1) if the entity was acting as a receiving entity, the most recent checkpoint identifier from the list in its docket;

2) if the entity was acting as a sending entity, equal to the value received.

It shall then enter the state "data transfer idle" and proceed as specified in 19.2.3.5.

b) If it is not prepared to continue the activity but because of local system considerations is likely to be able to continue later, it shall issue a F-RECOVER response primitive with an action result parameter indicating a transient error, and optionally with a diagnostic parameter carrying a value for recommended retry time.

c) If it does not have a docket for the activity, or if its recorded bulk data transfer numbers do not correspond it shall issue an F-RECOVER response primitive with an action result parameter indicating a permanent error, and optionally a diagnostic indicating "activity identifier unknown".

19.2.3.4 If an initiating error recovery entity receives an F-RECOVER confirm primitive with an action result parameter indicating a permanent error, it shall examine its own docket and:

a) if the docket state is "starting", restart the complete activity by issuing an F-SELECT request, then an F-OPEN request to the internal file service provider as in 19.1.3.1.

b) if the state is "in progress" or "data transfer finishing", issue an F-P-ABORT to the external file service user with a result parameter indicating a permanent error;

c) if the state is "finished", issue an F-CLOSE confirm to the external file service user, indicating a successful completion, followed by an F-P-ABORT indication to the external file service user and an F-U-ABORT request to the internal file service provider, with a diagnostic parameter indicating the loss of selection.

19.2.3.5 If the F-RECOVER confirm carries an action result indicating success, the entities then take up their respective roles as sender and receiver in the data transfer procedure, starting with the F-READ or F-WRITE primitive and followed by the F-DATA primitive after the agreed recovery point. The receiver resumes the issue of F-DATA indications to the external file service user at the first such primitive which has not previously been issued.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Section five: Abstract Syntax

20 Abstract Syntax Definition

This clause specifies the abstract syntax for the FTAM protocol information.

NOTES

1 The syntax is defined using a small collection of basic data types that are application independent.

2 The means by which FTAM protocol information is to be transferred (as P-DATA or as user data on other presentation or ACSE service primitives) is defined in sections two and three.

20.1 Conventions

The abstract syntax is defined using the notation specified in ASN.1 (Abstract Syntax Notation One) defined in ISO 8824.

For each of the FTAM service parameters which is to be transferred by an FTAM PDU, there is a PDU field (an ASN.1 NamedType) with the same name as the corresponding service parameter (see ISO 8571-3), except for the differences required by the use of ASN.1, which are that blanks are replaced by hyphens and capitals by their corresponding lower case letters, e. g. "Contents Type" becomes "contents-type".

The same naming convention is used to define the types of the file attributes (see ISO 8571-2).

20.2 Primitive Types

The ASN.1 definition uses the following data types as they are defined in ISO 8824

a) Simple types

1) BOOLEAN;

2) INTEGER;

3) BIT STRING; OCTET STRING;

4) NULL;

5) GraphicString;

6) GeneralizedTime;

7) EXTERNAL;

8) OBJECT IDENTIFIER;

9) ObjectDescriptor

b) Structured types

1) SEQUENCE;

2) SEQUENCE OF;

3) SET;

4) SET OF;

5) CHOICE;

6) Tagged

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

20.3 ASN.1 Module Definition

The comments in this ASN.1 module definition form part of the standard, and shall be implemented.

A cross-reference of the module definition is included in annex C.

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  PDU ::= CHOICE { FTAM-Regime-PDU, File-PDU, Bulk-Data-PDU }

  FTAM-Regime-PDU ::= CHOICE {
     f-initialize-request  [0] IMPLICIT F-INITIALIZE-request,
     f-initialize-response [1] IMPLICIT F-INITIALIZE-response,
     f-terminate-request   [2] IMPLICIT F-TERMINATE-request,
     f-terminate-response  [3] IMPLICIT F-TERMINATE-response,
     f-u-abort-request     [4] IMPLICIT F-U-ABORT-request,
     f-p-abort-request     [5] IMPLICIT F-P-ABORT-request }

  F-INITIALIZE-request ::= SEQUENCE {
     protocol-version                Protocol-Version DEFAULT { version-1 },
     implementation-information      Implementation-Information OPTIONAL,
     presentation-context-management [2] IMPLICIT BOOLEAN DEFAULT FALSE,
     service-class                   Service-Class DEFAULT { transfer-class },
      -- Only the valid combinations as specified in ISO 8571-3 are allowed
     functional-units                Functional-Units,
     attribute-groups                Attribute-Groups DEFAULT {},
     shared-ASE-information          Shared-ASE-Information OPTIONAL,
     ftam-quality-of-service         FTAM-Quality-Of-Service,
     contents-type-list              Contents-Type-List OPTIONAL,
     initiator-identity              User-Identity OPTIONAL,
     account                         Account OPTIONAL,
     filestore-password              Password OPTIONAL,
     checkpoint-window               [8] IMPLICIT INTEGER DEFAULT 1 }

  F-INITIALIZE-response ::= SEQUENCE {
     state-result                    State-Result  DEFAULT success,
     action-result                   Action-Result DEFAULT success,
     protocol-version                Protocol-Version DEFAULT { version-1 },
     implementation-information      Implementation-Information OPTIONAL,
     presentation-context-management [2] IMPLICIT BOOLEAN DEFAULT FALSE,
     service-class                   Service-Class DEFAULT { transfer-class },
      -- Only the valid combinations as specified in ISO 8571-3 are allowed
     functional-units                Functional-Units,
     attribute-groups                Attribute-Groups DEFAULT {},
     shared-ASE-information          Shared-ASE-Information OPTIONAL,
     ftam-quality-of-service         FTAM-Quality-Of-Service,
     contents-type-list              Contents-Type-List OPTIONAL,
     diagnostic                      Diagnostic OPTIONAL,
     checkpoint-window               [8] IMPLICIT INTEGER DEFAULT 1 }

```

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Figure 7 — FTAM regime PDUs

```
  Protocol-Version ::= [0] IMPLICIT BIT STRING { version-1 (0) }

  Implementation-Information ::= [1] IMPLICIT GraphicString
  -- this parameter is provided solely for the convenience of implementors needing
  -- to distinguish between implementations of a specific version number on different
  -- equipment, it shall not be the subject of conformance test.

  Service-Class ::= [3] IMPLICIT BIT STRING
                  { unconstrained-class (0),
                    management-class (1),
                    transfer-class (2),
                    transfer-and-management-class (3),
                    access-class (4) }

  Functional-Units ::= [4] IMPLICIT BIT STRING
                  { read (2),
                    write (3),
                    file-access (4),
                    limited-file-management (5),
                    enhanced-file-management (6),
                    grouping (7),
                    fadu-locking (8),
                    recovery (9),
                    restart-data-transfer (10) }
   -- Values 2 to 10 are chosen to align with numbering scheme used in ISO 8571-3

  Attribute-Groups ::= [5] IMPLICIT BIT STRING
                  { storage (0),
                    security (1),
                    private (2) }

  FTAM-Quality-Of-Service ::= [6] IMPLICIT INTEGER
                  { no-recovery (0),
                    class-1-recovery (1),
                    class-2-recovery (2),
                    class-3-recovery (3) }

  Contents-Type-List ::= [7] IMPLICIT SEQUENCE OF CHOICE
           { document-type-name   Document-Type-Name,
             abstract-syntax-name Abstract-Syntax-Name }

  F-TERMINATE-request ::= SEQUENCE {
      shared-ASE-information Shared-ASE-Information OPTIONAL }

  F-TERMINATE-response ::= SEQUENCE {
      shared-ASE-information Shared-ASE-Information OPTIONAL,
      charging               Charging OPTIONAL }

  F-U-ABORT-request ::= SEQUENCE {
      action-result      Action-Result DEFAULT success,
      diagnostic         Diagnostic OPTIONAL }

  F-P-ABORT-request ::= SEQUENCE {
      action-result  Action-Result DEFAULT success,
      diagnostic     Diagnostic OPTIONAL }

END
```

Figure 7 (continued) — FTAM regime PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  File-PDU ::= CHOICE {
        f-select-request         [ 6] IMPLICIT F-SELECT-request,
        f-select-response        [ 7] IMPLICIT F-SELECT-response,
        f-deselect-request       [ 8] IMPLICIT F-DESELECT-request,
        f-deselect-response      [ 9] IMPLICIT F-DESELECT-response,
        f-create-request         [10] IMPLICIT F-CREATE-request,
        f-create-response        [11] IMPLICIT F-CREATE-response,
        f-delete-request         [12] IMPLICIT F-DELETE-request,
        f-delete-response        [13] IMPLICIT F-DELETE-response,
        f-read-attrib-request    [14] IMPLICIT F-READ-ATTRIB-request,
        f-read-attrib-response   [15] IMPLICIT F-READ-ATTRIB-response,
        f-change-attrib-request  [16] IMPLICIT F-CHANGE-ATTRIB-request,
        f-change-attrib-response [17] IMPLICIT F-CHANGE-ATTRIB-response,
        f-open-request           [18] IMPLICIT F-OPEN-request,
        f-open-response          [19] IMPLICIT F-OPEN-response,
        f-close-request          [20] IMPLICIT F-CLOSE-request,
        f-close-response         [21] IMPLICIT F-CLOSE-response,
        f-begin-group-request    [22] IMPLICIT F-BEGIN-GROUP-request,
        f-begin-group-response   [23] IMPLICIT F-BEGIN-GROUP-response,
        f-end-group-request      [24] IMPLICIT F-END-GROUP-request,
        f-end-group-response     [25] IMPLICIT F-END-GROUP-response,
        f-recover-request        [26] IMPLICIT F-RECOVER-request,
        f-recover-response       [27] IMPLICIT F-RECOVER-response,
        f-locate-request         [28] IMPLICIT F-LOCATE-request,
        f-locate-response        [29] IMPLICIT F-LOCATE-response,
        f-erase-request          [30] IMPLICIT F-ERASE-request,
        f-erase-response         [31] IMPLICIT F-ERASE-response }

  F-SELECT-request ::= SEQUENCE {
     attributes             Select-Attributes,
     requested-access       Access-Request,
     access-passwords       Access-Passwords OPTIONAL,
     concurrency-control    Concurrency-Control OPTIONAL,
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     account                Account OPTIONAL }

  F-SELECT-response ::= SEQUENCE {
     state-result           State-Result DEFAULT success,
     action-result          Action-Result DEFAULT success,
     attributes             Select-Attributes,
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     diagnostic             Diagnostic OPTIONAL }

  F-DESELECT-request ::= SEQUENCE {
     shared-ASE-information Shared-ASE-Information OPTIONAL }


  F-DESELECT-response ::= SEQUENCE {
     action-result          Action-Result DEFAULT success,
     charging               Charging OPTIONAL,
          -- Present if and only if the account field was present on
          -- the PDU which established the selection regime.
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     diagnostic             Diagnostic OPTIONAL }

```

Figure 8 — File selection and file open regime PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
F-CREATE-request ::= SEQUENCE {
   override                [0] IMPLICIT INTEGER
                { create-failure (0),
                  select-old-file (1),
                  delete-and-create-with-old-attributes (2),
                  delete-and-create-with-new-attributes (3) }
                       DEFAULT create-failure,
   initial-attributes      Create-Attributes,
   create-password         Password OPTIONAL,
        -- Present is an additional password to the filestore password
        -- is required to allow the file creation.
   requested-access        Access-Request,
   access-passwords        Access-Passwords OPTIONAL,
        -- Present only if required to satisfy access control
        -- requirement on existing file.
   concurrency-control     Concurrency-Control OPTIONAL,
   shared-ASE-information Shared-ASE-Information OPTIONAL,
   account                 Account OPTIONAL }

F-CREATE-response ::= SEQUENCE {
   state-result            State-Result   DEFAULT success,
   action-result           Action-Result  DEFAULT success,
   initial-attributes      Create-Attributes,
   shared-ASE-information Shared-ASE-Information OPTIONAL,
   diagnostic              Diagnostic     OPTIONAL }

F-DELETE-request ::= SEQUENCE {
   shared-ASE-information Shared-ASE-Information OPTIONAL }

F-DELETE-response ::= SEQUENCE {
   action-result           Action-Result  DEFAULT success,
   shared-ASE-information Shared-ASE-Information OPTIONAL,
   charging                Charging OPTIONAL,
   diagnostic              Diagnostic OPTIONAL }

F-READ-ATTRIB-request ::= SEQUENCE {
   attribute-names     [0] IMPLICIT BIT STRING
        -- Kernel group
           { read-filename (0),
             read-permitted-actions (1),
             read-contents-type (2),
        -- Storage group
             read-storage-account (3),
             read-date-and-time-of-creation (4),
             read-date-and-time-of-last-modification (5),
             read-date-and-time-of-last-read-access (6),
             read-date-and-time-of-last-attribute-modification (7),
             read-identity-of-creator (8),
             read-identity-of-last-modifier (9),
             read-identity-of-last-reader (10),
             read-identity-of-last-attribute-modifier (11),
             read-file-availability (12),
             read-filesize (13),
             read-future-filesize (14),
        -- Security group
             read-access-control (15),
             read-legal-qualifications (16),
        -- Private group
             read-private-use (17) } }
```

Figure 8 (continued) — File selection and file open regime PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
F-READ-ATTRIB-response ::= SEQUENCE {
   action-result  Action-Result   DEFAULT success,
   attributes     Read-Attributes OPTIONAL,
        -- Password values within access control can not be read by means of
        -- the read attribute action. Whether other parts of the access control
        -- file attribute can be read by means of the read attribute action is
        -- decided locally by the responding entity, and it shall not be the
        -- subject of conformance test.
   diagnostic     Diagnostic      OPTIONAL }

F-CHANGE-ATTRIB-request ::= SEQUENCE {
   attributes     Change-Attributes }

F-CHANGE-ATTRIB-response ::= SEQUENCE {
   action-result  Action-Result DEFAULT success,
   attributes     Change-Attributes     OPTIONAL,
        -- Password values within access control attribute are never returned.
        -- Other attributes are returned as an implementation choice.
   diagnostic     Diagnostic     OPTIONAL }

F-OPEN-request ::= SEQUENCE {
   processing-mode        [0] IMPLICIT BIT STRING
                     { f-read (0),
                       f-insert (1),
                       f-replace (2),
                       f-extend (3),
                       f-erase (4) } DEFAULT { f-read },
   contents-type          [1] CHOICE {
                  unknown  [0] IMPLICIT NULL,
                  proposed [1] Contents-Type-Attribute },
   concurrency-control    Concurrency-Control OPTIONAL,
   shared-ASE-information Shared-ASE-Information  OPTIONAL,
   enable-fadu-locking    [2] IMPLICIT BOOLEAN DEFAULT FALSE,
   activity-identifier    Activity-Identifier OPTIONAL,
        -- Only used in the recovery functional unit.
   recovery-mode          [3] IMPLICIT INTEGER
                     { none (0),
                       at-start-of-file (1),
                       at-any-active-checkpoint (2) } DEFAULT none,
   remove-contexts        [4] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
   define-contexts        [5] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL }

F-OPEN-response ::= SEQUENCE {
   state-result           State-Result DEFAULT success,
   action-result          Action-Result DEFAULT success,
   contents-type          [1] Contents-Type-Attribute,
   concurrency-control    Concurrency-Control OPTIONAL,
   shared-ASE-information Shared-ASE-Information  OPTIONAL,
   diagnostic             Diagnostic OPTIONAL,
   recovery-mode          [3] IMPLICIT INTEGER
                     { none (0),
                       at-start-of-file (1),
                       at-any-active-checkpoint (2) } DEFAULT none,
   presentation-action    [6] IMPLICIT BOOLEAN DEFAULT FALSE }
      -- this flag is set if the responder is going to follow this response by
      -- a P-ALTER-CONTEXT exchange

F-CLOSE-request ::= SEQUENCE {
   action-result           Action-Result DEFAULT success,
   shared-ASE-information  Shared-ASE-Information OPTIONAL,
   diagnostic              Diagnostic OPTIONAL }
```

Figure 8 (continued) — File selection and file open regime PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
F-CLOSE-response ::= SEQUENCE {
   action-result           Action-Result DEFAULT success,
   shared-ASE-information  Shared-ASE-Information  OPTIONAL,
   diagnostic              Diagnostic OPTIONAL }

F-BEGIN-GROUP-request ::= SEQUENCE {
   threshold       [0] IMPLICIT INTEGER }

F-BEGIN-GROUP-response ::= SEQUENCE { }
   -- No elements defined, shall be empty

F-END-GROUP-request ::= SEQUENCE { }
   -- No elements defined, shall be empty

F-END-GROUP-response ::= SEQUENCE { }
   -- No elements defined, shall be empty

F-RECOVER-request ::= SEQUENCE {
   activity-identifier  Activity-Identifier,
   bulk-transfer-number [0] IMPLICIT INTEGER,
   requested-access     Access-Request,
   access-passwords     Access-Passwords OPTIONAL,
   recovery-point       [2] IMPLICIT INTEGER DEFAULT 0,
      -- zero indicates beginning of file
      -- point after last checkpoint indicates end of file
   remove-contexts      [3] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
   define-contexts      [4] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL }

F-RECOVER-response ::= SEQUENCE {
   state-result        State-Result  DEFAULT success,
   action-result       Action-Result DEFAULT success,
   contents-type       [1] Contents-Type-Attribute,
   recovery-point      [2] IMPLICIT INTEGER DEFAULT 0,
      -- zero indicates beginning of file
      -- point after last checkpoint indicates end of file
   diagnostic          Diagnostic OPTIONAL,
   presentation-action [6] IMPLICIT BOOLEAN DEFAULT FALSE }
      -- this flag is set if the responder is going to follow this response by
      -- a P-ALTER-CONTEXT exchange

F-LOCATE-request ::= SEQUENCE {
   file-access-data-unit-identity FADU-Identity,
   fadu-lock                      FADU-Lock OPTIONAL }

F-LOCATE-response ::= SEQUENCE {
   action-result                  Action-Result DEFAULT success,
   file-access-data-unit-identity FADU-Identity OPTIONAL,
   diagnostic                     Diagnostic OPTIONAL }

F-ERASE-request ::= SEQUENCE {
   file-access-data-unit-identity  FADU-Identity }

F-ERASE-response ::= SEQUENCE {
   action-result  Action-Result DEFAULT success,
   diagnostic     Diagnostic OPTIONAL }

END
```

Figure 8 (continued) — File selection and file open regime PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  Bulk-Data-PDU ::= CHOICE {
        f-read-request          [32] IMPLICIT F-READ-request,
        f-write-request         [33] IMPLICIT F-WRITE-request,
             -- There is no F-DATA FPDU, the contents of a file
             -- are transferred in a diferent presentation context
             -- and there is therefore no need to define the types
             -- of file contents in the FTAM PCI abstract syntax.
             -- File contents data are carried in values of the
             -- data type Data-Element as defined in ISO 8571-2
        f-data-end-request      [34] IMPLICIT F-DATA-END-request,
        f-transfer-end-request  [35] IMPLICIT F-TRANSFER-END-request,
        f-transfer-end-response [36] IMPLICIT F-TRANSFER-END-response,
        f-cancel-request        [37] IMPLICIT F-CANCEL-request,
        f-cancel-response       [38] IMPLICIT F-CANCEL-response,

             -- There is no F-CHECK PDU
        f-restart-request       [39] IMPLICIT F-RESTART-request,
        f-restart-response      [40] IMPLICIT F-RESTART-response }
  F-READ-request ::= SEQUENCE {
     file-access-data-unit-identity FADU-Identity,
     access-context                 Access-Context,
     fadu-lock                      FADU-Lock OPTIONAL }

  F-WRITE-request ::= SEQUENCE {
     file-access-data-unit-operation [0] IMPLICIT INTEGER
                       { insert (0),
                         replace (1),
                         extend (2) },
     file-access-data-unit-identity FADU-Identity,
     fadu-lock                      FADU-Lock OPTIONAL }

  F-DATA-END-request ::= SEQUENCE {
     action-result  Action-Result DEFAULT success,
     diagnostic     Diagnostic OPTIONAL }

  F-TRANSFER-END-request ::= SEQUENCE {
     shared-ASE-information  Shared-ASE-Information OPTIONAL }

  F-TRANSFER-END-response ::= SEQUENCE {
     action-result          Action-Result DEFAULT success,
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     diagnostic             Diagnostic OPTIONAL }

  F-CANCEL-request ::= SEQUENCE {
     action-result          Action-Result DEFAULT success,
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     diagnostic             Diagnostic OPTIONAL }

  F-CANCEL-response ::= SEQUENCE {
     action-result          Action-Result DEFAULT success,
     shared-ASE-information Shared-ASE-Information OPTIONAL,
     diagnostic             Diagnostic OPTIONAL }

  F-RESTART-request ::= SEQUENCE {
     checkpoint-identifier [0] IMPLICIT INTEGER }

  F-RESTART-response ::= SEQUENCE {
     checkpoint-identifier [0] IMPLICIT INTEGER }

END
```

Figure 9 — Bulk data transfer PDUs

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  Abstract-Syntax-Name ::= [APPLICATION 0] IMPLICIT OBJECT IDENTIFIER

  Access-Context ::= [APPLICATION 1] IMPLICIT SEQUENCE {
     access-context [0] IMPLICIT INTEGER
                  { hierarchical-all-data-units (0),     -- HA
                    hierarchical-no-data-units (1),      -- HN
                    flat-all-data-units (2),             -- FA
                    flat-one-level-data-units (3),       -- FL
                    flat-single-data-unit (4),           -- FS
                    unstructured-all-data-units (5),     -- UA
                    unstructured-single-data-unit (6) }, -- US
     level-number   [1] IMPLICIT INTEGER OPTIONAL }
          -- Present if and only if flat-one-level-data-units
          -- (access context FL) is selected.
     -- As defined in ISO 8571-2

  Access-Passwords ::= [APPLICATION 2] IMPLICIT SEQUENCE {
     read-password             [0] IMPLICIT Password,
     insert-password           [1] IMPLICIT Password,
     replace-password          [2] IMPLICIT Password,
     extend-password           [3] IMPLICIT Password,
     erase-password            [4] IMPLICIT Password,
     read-attribute-password   [5] IMPLICIT Password,
     change-attribute-password [6] IMPLICIT Password,
     delete-password           [7] IMPLICIT Password }

  Access-Request ::= [APPLICATION 3] IMPLICIT BIT STRING
                  { read (0),
                    insert (1),
                    replace (2),
                    extend (3),
                    erase (4),
                    read-attribute (5),
                    change-attribute (6),
                    delete-file (7) }

  Account ::= [APPLICATION 4] IMPLICIT GraphicString

  Action-Result ::= [APPLICATION 5] IMPLICIT INTEGER
             { success (0),
               transient-error (1),
               permanent-error (2) }

  Activity-Identifier ::= [APPLICATION 6] IMPLICIT INTEGER

  Application-Entity-Title ::= [APPLICATION 7] ACSE-1.AE-title
     -- As defined in ISO 8650

  Change-Attributes ::= [APPLICATION 8] IMPLICIT SEQUENCE {
  -- Kernel group
     filename            [ 0] IMPLICIT Filename-Attribute OPTIONAL,
  -- Storage group
     storage-account     [ 3] Account-Attribute OPTIONAL,
     file-availability   [12] File-Availability-Attribute OPTIONAL,
     future-filesize     [14] Filesize-Attribute OPTIONAL,
  -- Security group
     access-control      [15] Access-Control-Change-Attribute OPTIONAL,
     legal-qualification [16] Legal-Qualification-Attribute OPTIONAL,
  -- Private group
     private-use         [17] Private-Use-Attribute OPTIONAL }

```

Figure 10 — Application wide types

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
Charging ::= [APPLICATION 9] IMPLICIT SEQUENCE OF SEQUENCE {
   resource-identifier [0] IMPLICIT GraphicString,
   charging-unit       [1] IMPLICIT GraphicString,
   charging-value      [2] IMPLICIT INTEGER }

Concurrency-Control ::= [APPLICATION 10] IMPLICIT SEQUENCE
               { read             [0] IMPLICIT Lock,
                 insert           [1] IMPLICIT Lock,
                 replace          [2] IMPLICIT Lock,
                 extend           [3] IMPLICIT Lock,
                 erase            [4] IMPLICIT Lock,
                 read-attribute   [5] IMPLICIT Lock,
                 change-attribute [6] IMPLICIT Lock,
                 delete-file      [7] IMPLICIT Lock }

Lock  ::= INTEGER
       { not-required (0),
         shared (1),
         exclusive (2),
         no-access (3) }

Constraint-Set-Name ::= [APPLICATION 11] IMPLICIT OBJECT IDENTIFIER

Create-Attributes ::= [APPLICATION 12] IMPLICIT SEQUENCE {
-- Kernel group
   filename            [ 0] IMPLICIT Filename-Attribute,
   permitted-actions   [ 1] IMPLICIT Permitted-Actions-Attribute,
   contents-type       [ 2] Contents-Type-Attribute,
-- Storage group
   storage-account     [ 3] Account-Attribute OPTIONAL,
   file-availability   [12] File-Availability-Attribute OPTIONAL,
   future-filesize     [14] Filesize-Attribute OPTIONAL,
-- Security group
   access-control      [15] Access-Control-Attribute OPTIONAL,
   legal-qualification [16] Legal-Qualification-Attribute OPTIONAL,
-- Private group
   private-use         [17] Private-Use-Attribute OPTIONAL }

Diagnostic  ::= [APPLICATION 13] IMPLICIT SEQUENCE OF SEQUENCE {
   diagnostic-type  [0] IMPLICIT INTEGER
           { informative (0),
             transient (1),
             permanent (2) },
   error-identifier [1] IMPLICIT INTEGER,
        -- As defined in ISO 8571-3
   error-observer   [2] IMPLICIT Entity-Reference,
   error-source     [3] IMPLICIT Entity-Reference,
   suggested-delay  [4] IMPLICIT INTEGER OPTIONAL,
   further-details  [5] IMPLICIT GraphicString OPTIONAL }

Entity-Reference ::= INTEGER
           { no-categorization-possible (0),
             initiating-file-service-user (1),
             initiating-file-protocol-machine (2),
             service-supporting-the-file-protocol-machine (3),
             responding-file-protocol-machine (4),
             responding-file-service-user (5) }
   --
   -- NOTE
   --  1. The values 0 and 3 are only valid as values in error-source
   --
   --  2. The value 5 corresponds to the virtual filestore.

Document-Type-Name ::= [APPLICATION 14] IMPLICIT OBJECT IDENTIFIER
```

Figure 10 (continued) — Application wide types

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```

  FADU-Identity ::= [APPLICATION 15] CHOICE
     { first-last  [0] IMPLICIT INTEGER { first (0), last (1) },
       relative    [1] IMPLICIT INTEGER { previous (0), current (1), next (2) },
       begin-end   [2] IMPLICIT INTEGER { begin (0), end (1) },
       single-name [3] IMPLICIT Node-Name,
       name-list   [4] IMPLICIT SEQUENCE OF Node-Name,
       fadu-number [5] IMPLICIT INTEGER }
         -- As defined in ISO 8571-2

  Node-Name ::= EXTERNAL
    -- The type to be used for Node-Name is defined in ISO8571-FADU

  FADU-Lock ::= [APPLICATION 16] IMPLICIT INTEGER
                { off (0), on (1) }

  Password ::= [APPLICATION 17] CHOICE { GraphicString, OCTET STRING }

  Read-Attributes ::= [APPLICATION 18] IMPLICIT SEQUENCE {
  -- Kernel group
     filename                  [0] IMPLICIT Filename-Attribute OPTIONAL,
     permitted-actions         [1] IMPLICIT Permitted-Actions-Attribute OPTIONAL,
     contents-type             [2] Contents-Type-Attribute OPTIONAL,
  -- Storage group
     storage-account           [3] Account-Attribute OPTIONAL,
     date-and-time-of-creation [4] Date-and-Time-Attribute OPTIONAL,
     date-and-time-of-last-modification
                               [5] Date-and-Time-Attribute OPTIONAL,
     date-and-time-of-last-read-access
                               [6] Date-and-Time-Attribute OPTIONAL,
     date-and-time-of-last-attribute-modification
                               [7] Date-and-Time-Attribute OPTIONAL,
     identity-of-creator       [8] User-Identity-Attribute OPTIONAL,
     identity-of-last-modifier [9] User-Identity-Attribute OPTIONAL,
     identity-of-last-reader  [10] User-Identity-Attribute OPTIONAL,
     identity-of-last-attribute-modifier
                              [11] User-Identity-Attribute OPTIONAL,
     file-availability        [12] File-Availability-Attribute OPTIONAL,
     filesize                 [13] Filesize-Attribute OPTIONAL,
     future-filesize          [14] Filesize-Attribute OPTIONAL,
  -- Security group
     access-control           [15] Access-Control-Attribute OPTIONAL,
     legal-qualification      [16] Legal-Qualification-Attribute OPTIONAL,
  -- Private group
     private-use              [17] Private-Use-Attribute OPTIONAL }

  Select-Attributes ::= [APPLICATION 19] IMPLICIT SEQUENCE {
  -- Kernel group
     filename [0] IMPLICIT Filename-Attribute }

  Shared-ASE-Information ::= [APPLICATION 20] IMPLICIT EXTERNAL
     -- This field may be used to convey commitment control as
     -- as described in ISO 8571-3

  State-Result ::= [APPLICATION 21] IMPLICIT INTEGER
             { success (0),
               failure (1) }

  User-Identity ::= [APPLICATION 22] IMPLICIT GraphicString

END
```

Figure 10 (continued) — Application wide types

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  Access-Control-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
       -- Indicates partial support of this attribute.
       -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT SET OF Access-Control-Element }
     --  The semantics of this attribute is described in ISO 8571-2

  Access-Control-Change-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
       -- Indicates partial support of this attribute.
       -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT SEQUENCE {
       insert-values  [0] IMPLICIT SET OF Access-Control-Element OPTIONAL,
             -- This field is used by the change attribute actions to indicate
             -- new values to be inserted in the access control file attribute.
       delete-values  [1] IMPLICIT SET OF Access-Control-Element OPTIONAL} }
             -- This field is used by the change attribute action to indicate
             -- old values to be removed from the access control file attribute.
     --  The semantics of this attribute is described in ISO 8571-2

  Access-Control-Element ::= SEQUENCE {
     action-list        [0] IMPLICIT Access-Request,
     concurrency-access [1] IMPLICIT Concurrency-Access OPTIONAL,
     identity           [2] IMPLICIT User-Identity OPTIONAL,
     passwords          [3] IMPLICIT Access-Passwords OPTIONAL,
     location           [4] IMPLICIT Application-Entity-Title OPTIONAL }

  Concurrency-Access ::= SEQUENCE {
               read             [0] IMPLICIT Concurrency-Key,
               insert           [1] IMPLICIT Concurrency-Key,
               replace          [2] IMPLICIT Concurrency-Key,
               extend           [3] IMPLICIT Concurrency-Key,
               erase            [4] IMPLICIT Concurrency-Key,
               read-attribute   [5] IMPLICIT Concurrency-Key,
               change-attribute [6] IMPLICIT Concurrency-Key,
               delete-file      [7] IMPLICIT Concurrency-Key }

  Concurrency-Key ::= BIT STRING
                    { not-required (0),
                      shared (1),
                      exclusive (2),
                      no-access (3) }

  Account-Attribute ::=  CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute
          -- This value shall only appear in response PDUs
     actual-values      Account }

  Contents-Type-Attribute ::= CHOICE {
     document-type                      [0] IMPLICIT SEQUENCE {
          document-type-name Document-Type-Name,
          parameter                     [0] ANY OPTIONAL },
     -- The actual types to be used for values of the parameter field
     -- are defined in the named document type.
     constraint-set-and-abstract-syntax [1] IMPLICIT SEQUENCE {
          constraint-set-name  Constraint-Set-Name,
          abstract-syntax-name Abstract-Syntax-Name } }
```

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Figure 11 — File attribute types

```
  Date-and-Time-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT GeneralizedTime }

  File-Availability-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT INTEGER
             { immediate-availability (0),
               deferred-availability (1) } }

  Filename-Attribute ::= SEQUENCE OF GraphicString

  Filesize-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT INTEGER }

  Legal-Qualification-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     actual-values      [1] IMPLICIT GraphicString }

  Permitted-Actions-Attribute ::= BIT STRING
          -- Actions available
             { read (0),
               insert (1),
               replace (2),
               extend (3),
               erase (4),
               read-attribute (5),
               change-attribute (6),
               delete-file (7),
          -- FADU-Identity groups available
               traversal (8),
               reverse-traversal (9),
               random-order (10) }

  Private-Use-Attribute ::= CHOICE {
     no-value-available            [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     abstract-syntax-not-supported [1] IMPLICIT NULL,
          -- Indicates, that abstract syntax is not available
     actual-values                 [2] IMPLICIT EXTERNAL }

  User-Identity-Attribute ::= CHOICE {
     no-value-available [0] IMPLICIT NULL,
          -- Indicates partial support of this attribute.
          -- This value shall only appear in response PDUs
     actual-values      User-Identity }

END
```

Figure 11 (continued) — File attribute types

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

20.4 Abstract Syntax Definition

This International Standard assigns the ASN.1 object identifier value

{ iso standard 8571 abstract-syntax (2) ftam-pci (1) }

as an abstract syntax name for the set of presentation data values, each of which is a value of the ASN.1 type ISO8571-FTAM.PDU. The corresponding ASN.1 object descriptor value shall be

"FTAM PCI"

The ASN.1 object identifier and object descriptor values

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

and

"Basic Encoding of a single ASN.1 type"

(assigned to an information object in ISO 8825) can be used as a transfer syntax name with this abstract syntax.

21 Application Context Name

For the purpose of being able to run an application which only contains the ACSE and FTAM as ASEs, the object identifier value

{ iso standard 8571 application-context (1) iso-ftam (1) }

and the object descriptor value

"ISO FTAM"

are assigned to an information object of type

ACSE-1.Application-context-name

as defined in ISO 8650.

NOTE - This definition is subject to refinement as a result of on-going work within ISO in the area of the Application Layer structure.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Section six: Conformance

22 Conformance

A system claiming to implement the procedures specified in this International Standard shall comply with the requirements in 22.1 to 22.3.

22.1 Statement requirements

The following shall be stated by the implementor:

a) which file service classes are supported;

b) which functional units are supported;

c) which values of the recovery mode parameter are supported, in those cases where any error recovery procedures are implemented;

d) whether the system is capable of acting in the role of initiator or responder or both;

e) whether the system is capable of acting in the role of sender or receiver or both;

f) which optional groups of attributes (storage or security) are supported;

g) which level of support is associated with each individual attribute and its optional components (see ISO 8571-2);

h) the range of values supported for each attribute for which values are available;

i) which document types and/or which abstract and transfer syntaxes for the transfer of file contents are supported by the Open System on which the Application Entity resides;

j) whether the hierarchical file model (see ISO 8571-2) is supported, and, if so, which constraint sets and, where relevant, the maximum depth of hierarchy supported;

k) any circumstances under which the existence of a file, its contents, or the values of the supported attributes change between separate accesses using the FTAM protocol;

l) any circumstances under which the modifications to the file contents or the values of the file attributes by FTAM protocol exchanges will not subsequently be available for use;

m) which optional parameters are supported, and the range of values for all parameters;

n) which character sets it supports for FTAM-PCI and for FADU structuring information;

o) the range of values and their meaning and purpose, for the Private Use attribute, if it is supported;

p) which diagnostic codes are used, and in what circumstances.

22.2 Static requirements

The system shall:

a) act in the role of initiator or responder or both; (see ISO 8571-1)

b) support at least the kernel group of attributes; (see ISO 8571-2)

c) support at least the minimum range of values defined for each attribute which is claimed to be supported (see ISO 8571-2). The attributes in the Kernel group shall always be supported;

d) support the transfer syntax derived from the encoding rules specified in ISO 8825 and named

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

for the purpose of generating and interpreting FTAM protocol information.

e) support one or more of the allowable file service classes and the groups of file service functional units that are implied by those file service classes; no other groupings of functional units shall be supported;

f) if the read and/or write functional unit is supported, the mapping of F-CANCEL to P-DATA must be supported regardless of whether the mapping to P-RESYNCHRONIZE is also supported;

g) support the recovery mode parameter value "start of bulk data transfer" and the corresponding procedures if any error recovery procedures are supported.

22.3 Dynamic requirements

The system shall:

a) follow all procedures relevant to each functional unit that the system claims to implement. The procedures are specified in Annex A and sections two, three and four;

b) support the mapping onto the presentation service defined in clauses 7 and 12;

c) execute any operations (whether invoked using the FTAM protocol or otherwise), which change the contents or values of any attributes of a file, in a manner consistent with virtual filestore operations (as defined in ISO 8571-2), and shall update the values of any supported attributes accordingly;

d) for the purpose of testing, be capable of operating without the use of the Private Use and Legal Qualifications attributes.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

Annex A
Protocol state tables

(This annex forms part of the standard.)

A.1 Introduction

In the event of a discrepancy becoming apparent in the protocol described in the body of this standard and the protocol described in this Annex, this Annex is to take precedence.

These tables describe the operation of the basic file protocol machine (FPM) and error recovery protocol machine (FERPM).

The basic protocol tables are divided into three groups, covering:

a) the FTAM regime management protocol machine;

b) the file regime management protocol machine;

c) the bulk data transfer protocol machine.

The initial and final states of the file regime management protocol machine are also states of FTAM regime management protocol machine. The initial and final states of the bulk data transfer protocol machine are also states of the file regime management protocol machine.

To provide formally complete and consistent description of the FTAM protocol sequences of events are considered indivisible in the model. That indivisibility ensures both that the states used in the description are well defined and that they are sufficient to describe the protocol.

The reception of a service primitive and the generation of dependent actions are considered to be an indivisible action. The reception of an FPDU and the generation of dependent actions are considered to be an indivisible action.

The indivisibility of actions may, in some implementations, cause certain events from file service users to be invalid at some service interface.

The conventions adopted in this Annex are described below.

A.1.1 System model

There are four types of receivers and sources of incoming and outgoing events (see figure 12):

a) The internal file service user, which is the file error recovery protocol machine (FERPM);

b) The external file service user - initiator or responder;

c) The local system environment;

d) The underlying presentation and ACSE service providers.

All primitives to/from the external file service user are passed through as primitives to/from the internal file service user to the basic FPM.

If the FERPM is not null - that is, either the RESTART or the RECOVER functional unit is available - parameters needed for error control and recovery may be added to these primitives.

Signals to/from the local environment group together events signalling errors and events signalling interactions amongst the FERPM, its docket and its local file system. These local signals are

e) L-ERRABT — local signal indicating protocol or other local errors leading to F-P-ABORT, with a permanent error value in the action result parameter;

f) L-PABORT — local signal indicating that a F-P-ABORT request PDU with a transient error value in the action result should be issued;

g) L-ERROR1, L-ERROR2 and L-ERROR3 — local signals indicating class I, class II and class III errors respectively;

h) L-SUSPND — a local signal sent by the sender in the RESTART state to the local file system to suspend the issuing of F-DATA request and F-DATA-END request primitives;

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

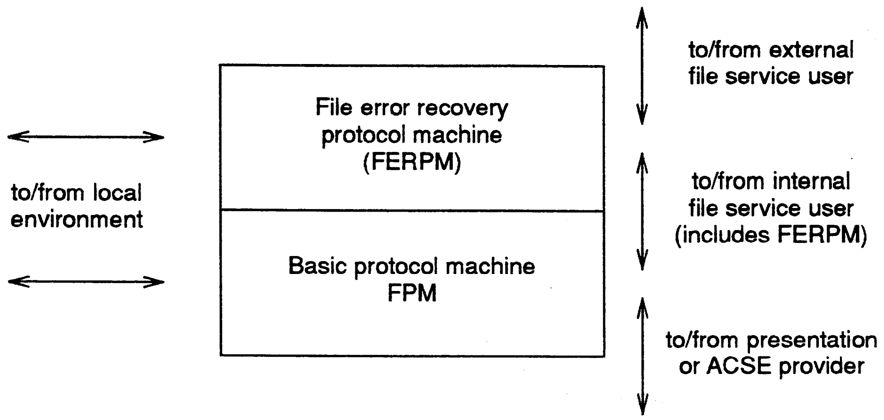

Figure 12 — State machine system model

i) L-RESUME — a local signal to the local file system to resume the issuing of F-DATA request and F-DATA-END request primitives when a sender has completed its recovery from RESTART and is now back in DATA-XFER state;

j) L-RESEND — a local signal from the FERPM to the sending local file system. This signal includes the negotiated restart checkpoint and eliminates the need to store data values in the docket. Upon receipt of this local signal the local file system can either

1) resend data from the checkpoint if the local file system is aware of the algorithm used by the FERPM to generate the checkpoint; or

2) resend data from the start of the file if it has no knowledge of the checkpoint generating algorithm. The FERPM in the RESTART state will discard all data from the local file system that occurs before the negotiated checkpoint. The FERPM will only forward data past the negotiated checkpoint to the FPM;

k) L-DATRQ — signal representing a re-issued F-DATA request primitive from the local file system;

l) L-DAERQ — signal representing a re-issued F-DATA-END request primitive when a "data-end" marker becomes available from the docket;

m) L-CHKRQ — signal representing a re-issued F-CHECK request primitive when a checkpoint identifier becomes available from the docket;

n) L-EORIN — end of restart indication signal indicating that all checkpoint identifiers, "data-end" markers and all data to be resent have been sent;

o) L-GIVEUP — signal indicating that the responding FERPM should abandon the recovery process. This is introduced so that if the initiating FERPM is unable to re-establish the association then the responding FERPM can inform its users of the irrecoverable error.

p) L-ERRCTX — signal indicating that the P-ALTER-CONTEXT negotiation is unsuccessful. The error is to be processed following the F-OPEN or F-RECOVER confirm as applicable.

A.1.2 Incoming Events

Events in the tables are depicted by abbreviated names. A list of these abbreviations, arranged alphabetically within categories, precedes each table. Wherever possible, these events have been arranged in the order in which a normal error-free sequence of events would occur.

It is assumed that incoming PDUs have been extracted from the user data fields of Presentation or ACSE indication or confirm primitives prior to being considered as incoming events.

The following naming conventions are used:

a) The prefix "F-" (as in F-INIRQ) indicates a primitive issued by the external file service (EFS) user or issued by the FERPM to the EFS user;

b) The prefix "P-" (as in P-CONRQ) indicates a primitive issued by the Presentation Service (PS) provider.

c) The prefix "A-" (as in A-PABIN) indicates a primitive issued by the ACSE provider;

d) the prefix "L-" (as in L-ERRABT) indicates a local signal from the FPM or the FERPM to the local system or from the local system environment to the FPM or the FERPM;

e) the prefix "I-" (as in I-OPNRQ) indicates a primitive issued by the FERPM to the FPM or vice-versa;

f) Where no prefix is used, the event is the receipt of an FTAM PDU or, in the case of GRPRQ and GRPRP, a grouped sequence of PDUs. The only exception is "DATIN" which indicates a data value in user context in the data transfer state.

g) The following suffices indicate the basic types of primitives and PDUs:

"RQ"	request
"IN"	indication
"RP"	response
"CF"	confirm

A.1.3 Outgoing Events

The same naming conventions are used for outgoing events as for incoming events. Where the outgoing event is the issue of a PDU, it will normally give rise to the similarly named incoming event for the other protocol entity. For example, the outgoing event SELRP of the responding entity will become the incoming event SELRP for the initiating entity.

A.1.4 States

The suffix "-PD" indicates a pending state, waiting for some known primitive or PDU type. The suffix "-EX" indicates an expectant state in FERPM, waiting for some expected primitive.

The prefix "P-" normally indicates waiting for a primitive from the Presentation Service provider. Similarly, the prefix "F-" indicates waiting for a primitive from the EFS user. If neither of the above prefices is used, the state normally indicates waiting for a PDU, or a non-pending state.

A.1.5 Predicates

The following symbols used in predicates have their normal Boolean algebraic meanings:

&	AND
|	OR
~	NOT

An incoming event may satisfy more than one predicate, in which case the conditional actions for satisfied predicates will be executed.

Some actions may be conditional upon the negotiation of particular service functional units;

those in the FPM are:

U1:	Kernel functional unit
U2:	Read functional unit
U3:	Write functional unit
U4:	File access functional unit
U5:	Limited management functional unit
U6:	Enhanced management functional unit
U7:	Grouping functional unit
U8:	FADU locking functional unit

those in the FERPM are:

U9:	Recovery functional unit
U10:	Restart data transfer functional unit

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.1.6 Actions

Actions may be conditional on specified predicates, or they may be unconditional. When a line in a "detailed entries" specification commences with a predicate specifier followed by a colon, this indicates that all actions on that line are conditional upon the indicated predicate. An action may consist of one or more of the following:

a) an outgoing event, indicated by its abbreviated name;

b) a specified action, indicated by a number in square brackets [ ] and separated from any preceding items by a comma;

c) a specified qualifier, indicated by a number in square brackets [ ] following a preceding item without an intervening comma;

Actions and qualifiers are described in a single list preceding the table.

The next state which the protocol entity will enter is indicated by an arrow preceding a state name, e.g. "⇒ SELECTED". For the null transition back to the current state, the notation "⇒ *same state*".

Once a state change occurs, all subsequent actions in that entry are ignored.

A.1.7 Implicit Action

The following entity actions have not been explicitly specified in the State tables, but constitute part of the entity behaviour:

a) A blank square in the table indicates an invalid event.

b) Unless otherwise stated in the tables, an invalid event would cause the action specified in 10.2 to be executed.

c) The Presentation Service is used throughout to identify "active" FTAM PCI from data with an equivalent encoding, but in a "passive" user data context.

d) For each incoming PDU, a check is made that the appropriate functional unit has been negotiated for the connection. If the check fails, the procedures for protocol violation specified in 10.2 are followed.

A.1.8 Additional State Information

The tables make use of the indicators and other state variables defined in 6.2 and 11.2. In addition, the file regime management tables for the initiator make use of a threshold indicator and an expected response list, as a means of specifying the entity state when a concatenated PDU group is outstanding.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.2 FTAM regime management protocol machine (Kernel functional unit)

A.2.1 States - FTAM regime management

UNINITIALIZED FTAM regime ended.

INITIALIZE-PD Initialize pending; wait for initialize response PDU.

INITIALIZED FTAM regime started.

TERMINATE-PD Terminate pending; wait for terminate response PDU.

I-INITIALIZE-PD Initialize pending; wait for F-INITIALIZE response primitive from the internal file service user

I-TERMINATE-PD Terminate pending; wait for F-TERMINATE response primitive from the internal file service user

ANY-OTHER Any other state of the file regime management protocol machine or the bulk data transfer protocol machine.

A.2.2 Incoming events - FTAM regime management

In the following lists, the functional unit in which the event occurs is included after the name, where applicable.

A.2.2.1 Incoming events — PDUs

UABRQ	U-Abort request PDU (on A-ABORT indication primitive)	U1
PABRQ	P-Abort request PDU (on A-ABORT indication primitive)	"
INIRQ	Initialize request PDU	"
INIRP	Initialize response PDU	"
TERRQ	Terminate request PDU	"
TERRP	Terminate response PDU	"

A.2.2.2 Incoming events from the internal file service user

I-UABRQ F-U-ABORT request primitive

I-INIRQ F-INITIALIZE request primitive

I-INIRP F-INITIALIZE response primitive

I-TERRQ F-TERMINATE request primitive

I-TERRP F-TERMINATE response primitive

A.2.2.3 Incoming events from the ACSE provider

A-PABIN A-P-ABORT indication primitive

A-ABIN A-ABORT indication primitive without user data

A-ASSCF A-ASSOCIATE confirm primitive without user data (with result indicating reject by ACSE or presentation service provider)

A.2.2.4 Incoming events from the local system

L-ERRABT Local signal indicating error leading to abort

L-PABORT Local signal indicating that a F-P-ABORT request PDU and a F-P-ABORT indication primitive to the internal file service user, both with a transient error value.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.2.3 Outgoing events — FTAM regime management

A.2.3.1 Outgoing events — PDUs

UABRQ	U-Abort request PDU
PABRQ	P-Abort request PDU
INIRQ	Initialize request PDU
INIRP	Initialize response PDU
TERRQ	Terminate request PDU
TERRP	Terminate response PDU

A.2.3.2 Outgoing events to the internal file service user

I-UABIN	F-U-ABORT indication primitive
I-PABIN	F-P-ABORT indication primitive
I-INIIN	F-INITIALIZE indication primitive
I-INICF	F-INITIALIZE confirm primitive
I-TERIN	F-TERMINATE indication primitive
I-TERCF	F-TERMINATE confirm primitive

A.2.4 Specific actions - FTAM regime management

[1] Send the PDU constructed as user data and map parameters on the appropriate ACSE form.

[3] Initialize state information - unset all additional state information indicators, set outstanding checkpoint counter, synch point number to zero.

[4] Set state result parameters to "success".

[5] In case of ACSE provider abort, if any diagnostic indicates communications failure then set the action result to transient error.

[6] Set state result parameter to "failure".

[7] Record the FQOS required and select the restart and/or recovery functional units if necessary.

[12] Establish the list of presentation contexts needed to support FTAM PCI and ACSE PCI abstract syntaxes. If necessary, determine, on the basis of the abstract syntaxes derived from the contents type list supplied by the external file service user, the list of presentation contexts needed to support the file contents, and add it to the previous list. The resultant list is used to construct the presentation context definition list parameter.

[13] Update the value of the contents type list parameter in the F-INITIALIZE indication primitive issued to the internal file service user, according to the abstract syntaxes rejected by the presentation service provider, if necessary.

[14] According to the value of the contents type list on the F-INITIALIZE response primitive received from the internal file service user, construct the presentation context definition result parameter.

[20] Set the action result parameter according to the result parameter of the A-ASSOCIATE confirm service primitive.

[62] Record the peer entity's checkpoint window.

A.2.5 Predicates — FTAM regime management

P1: I-INITIALIZE request primitive is acceptable.

P2: Result parameter of the ACSE confirm primitive indicates success of the operation.

P3: State result parameter of the incoming response primitive indicates success of the operation.

P5: Initialize request PDU is acceptable.

P25: Negotiable parameters have values consistent with request.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.2.6 Initiating entity state table FTAM regime management

STATE EVENT	UNINITIALIZED	INITIALIZE-PD	INITIALIZED	TERMINATE-PD	ANY-OTHER
I-INIRQ	1				
A-ASSCF		10			
INIRP		2			
I-TERRQ			3		
TERRP				4	
A-PABIN		5	5	5	5
A-ABIN		7	7	7	7
UABRQ		6	6	6	6
PABRQ		7	7	7	7
I-UABRQ		8	8	8	8
L-ERRABT		9	9	9	9
L-PABORT		9	9	9	9

A.2.7 Initiating entity state table: detailed entries

1:	P1:	[3],[12],INIRQ[1]	⇒ INITIALIZE-PD
	~P1:	I-INICF[6]	⇒ *same state*
2:	P2&P25:	[62],I-INICF[4],[7]	⇒ INITIALIZED
	P2&~P25:	I-INICF[6],PABRQ[1]	⇒ UNINITIALIZED
	~P2:	I-INICF[6]	⇒ UNINITIALIZED
3:		TERRQ[1]	⇒ TERMINATE-PD
4:		I-TERCF	⇒ UNINITIALIZED
5:		I-PABIN[5]	⇒ UNINITIALIZED
6:		I-UABIN	⇒ UNINITIALIZED
7:		I-PABIN	⇒ UNINITIALIZED
8:		UABRQ[1]	⇒ UNINITIALIZED
9:		PABRQ[1],I-PABIN	⇒ UNINITIALIZED
10:		I-PABIN[20]	⇒ UNINITIALIZED

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.2.8 Responding entity state table — FTAM regime management

STATE EVENT	UNINITIALIZED	I-INITIALIZE-PD	INITIALIZED	I-TERMINATE-PD	ANY-OTHER
INIRQ	1				
I-INIRP		2			
TERRQ			3		
I-TERRP				4	
A-PABIN		5	5	5	5
A-ABIN		7	7	7	7
UABRQ		6	6	6	6
PABRQ		7	7	7	7
I-UABRQ		8	8	8	8
L-ERRABT		9	9	9	9
L-PABORT		9	9	9	9

A.2.9 Responding entity state table: detailed entries

1:	P5:	I-ININ,[3],[62],[13]	⇛	I-INITIALIZE-PD
	~P5:	INIRP[6][1]	⇛	*same state*
2:	P3:	[7],[14],INIRP[1]	⇛	INITIALIZED
	~P3:	INIRP[6][1]	⇛	UNINITIALIZED
3:		I-TERIN	⇛	I-TERMINATE-PD
4:		TERRP[1]	⇛	UNINITIALIZED
5:		I-PABIN[5]	⇛	UNINITIALIZED
6:		I-UABIN	⇛	UNINITIALIZED
7:		I-PABIN	⇛	UNINITIALIZED
8:		UABRQ[1]	⇛	UNINITIALIZED
9:		PABRQ[1],I-PABIN	⇛	UNINITIALIZED

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3 File regime management protocol machine

A.3.1 States — file regime management

In the FPM:

INITIALIZED	FTAM regime started
SELECT-PD	Select pending, wait for select response PDU
I-SELECT-PD	Select pending, wait for F-SELECT response primitive from the internal file service user
SELECTED	Selected
DESELECT-PD	Deselect pending, wait for deselect response PDU
I-DESELECT-PD	Deselect pending, wait for F-DESELECT response primitive from the internal file service user
CREATE-PD	Create pending, wait for create response PDU
I-CREATE-PD	Create pending, wait for F-CREATE response primitive from the internal file service user
DELETE-PD	Delete pending, wait for delete response PDU
I-DELETE-PD	Delete pending, wait for F-DELETE response primitive from the internal file service user
READ-ATT-PD	Read attribute pending, wait for read attribute response PDU
I-READ-ATT-PD	Read attribute pending, wait for F-READ-ATTRIB response primitive from the internal file service user
CHG-ATT-PD	Change attribute pending, wait for change attribute response PDU.
I-CHG-ATT-PD	Change attribute pending, wait for F-CHANGE-ATTRIB response primitive from the internal file service user
OPEN-PD	Open pending, wait for open response PDU
I-OPEN-PD	Open pending, wait for F-OPEN response primitive from the internal file service user
P-ALTIN-PD	Presentation alter context pending, wait for P-ALTER-CONTEXT indication primitive.
P-ALTCF-PD	Presentation alter context pending, wait for P-ALTER-CONTEXT confirm primitive.
CLOSE-PD	Close pending, wait for close response PDU
I-CLOSE-PD	Close pending, wait for F-CLOSE response primitive from the internal file service user
DXFRIDLE	Data transfer idle
LOCATE-PD	Locate pending, wait for locate response PDU
I-LOCATE-PD	Locate pending, wait for F-LOCATE response primitive from the internal file service user
ERASE-PD	Erase pending, wait for erase response PDU
I-ERASE-PD	Erase pending, wait for F-ERASE response primitive from the internal file service user
GROUPING	Building PDU group, wait for required primitives
GROUP-PD	Group pending, wait for response PDU group
I-GROUP-PD	Group pending, wait for F-BEGIN-GROUP response primitive from the internal file service user

In the FERPM:

RECOVER-PD	Recover pending, wait for recover response PDU
I-RECOVER-PD	Recover pending, wait for F-RECOVER response primitive from the internal file service user
P-ALTIN-REC-PD	Presentation alter context pending, wait for P-ALTER-CONTEXT indication primitive during recovery.
P-ALTCF-REC-PD	Presentation alter context pending, wait for P-ALTER-CONTEXT confirm primitive during recovery.
DXFRIDLE-REC	Data transfer idle during recovery

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.2 Incoming events — file regime management

A.3.2.1 Incoming events — PDUs

CATRQ	Change-attrib request PDU	U6
CATRP	Change-attrib response PDU	U6
CLORQ	Close request PDU	U2,U3
CLORP	Close response PDU	U2,U3
CRERQ	Create request PDU	U5
CRERP	Create response PDU	U5
DELRQ	Delete request PDU	U5
DELRP	Delete response PDU	U5
DESRQ	Deselect request PDU	U1
DESRP	Deselect response PDU	U1
ERARQ	Erase request PDU	U4
ERARP	Erase response PDU	U4
LOCRQ	Locate request PDU	U4
LOCRP	Locate response PDU	U4
OPNRQ	Open request PDU	U2,U3
OPNRP	Open response PDU	U2,U3
RATRQ	Read-attrib request PDU	U5
RATRP	Read-attrib response PDU	U5
RECRQ	Recover request PDU	U9
RECRP	Recover response PDU	U9
SELRQ	Select request PDU	U1
SELRP	Select response PDU	U1
GRPRQ	A sequence of request PDUs (PDU-Group) preceded by an F-BEGIN-GROUP request PDU and followed by a F-END-GROUP request PDU.	U7
GRPRP	A sequence of response PDUs (PDU-Group) preceded by an F-BEGIN-GROUP response PDU and followed by a F-END-GROUP response PDU.	U7

A.3.2.2 Incoming events from the internal file service user

I-CATRQ	F-CHANGE-ATTRIB request primitive
I-CATRP	F-CHANGE-ATTRIB response primitive
I-CLORQ	F-CLOSE request primitive
I-CLORP	F-CLOSE response primitive
I-CRERQ	F-CREATE request primitive
I-CRERP	F-CREATE response primitive
I-DELRQ	F-DELETE request primitive
I-DELRP	F-DELETE response primitive
I-DESRQ	F-DESELECT request primitive
I-DESRP	F-DESELECT response primitive
I-ERARQ	F-ERASE request primitive
I-ERARP	F-ERASE response primitive
I-LOCRQ	F-LOCATE request primitive
I-LOCRP	F-LOCATE response primitive

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

I-OPNRQ	F-OPEN request primitive
I-OPNRP	F-OPEN response primitive
I-RATRQ	F-READ-ATTRIB request primitive
I-RATRP	F-READ-ATTRIB response primitive
I-RECRQ	F-RECOVER request primitive
I-RECRP	F-RECOVER response primitive
I-SELRQ	F-SELECT request primitive
I-SELRP	F-SELECT response primitive
I-BGPRQ	F-BEGIN-GROUP request primitive
I-BGPRP	F-BEGIN-GROUP response primitive
I-EGPRQ	I-END-GROUP request primitive
I-EGPRP	F-END-GROUP response primitive

A.3.2.3 Incoming events from the presentation service provider

P-ALTIN	P-ALTER-CONTEXT indication primitive
P-ALTCF	P-ALTER-CONTEXT confirm primitive

A.3.3 Outgoing events — file regime management

A.3.3.1 Outgoing events — PDUs

CATRQ	Change-attrib request PDU
CATRP	Change-attrib response PDU
CLORQ	Close request PDU
CLORP	Close response PDU
CRERQ	Create request PDU
CRERP	Create response PDU
DELRQ	Delete request PDU
DELRP	Delete response PDU
DESRQ	Deselect request PDU
DESRP	Deselect response PDU
ERARQ	Erase request PDU
ERARP	Erase response PDU
LOCRQ	Locate request PDU
LOCRP	Locate response PDU
OPNRQ	Open request PDU
OPNRP	Open response PDU
PABRQ	P-Abort request PDU
RATRQ	Read-attrib request PDU
RATRP	Read-attrib response PDU
RECRQ	Recover request PDU
RECRP	Recover response PDU
SELRQ	Select request PDU
SELRP	Select response PDU

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.3.2 Outgoing events to the internal file service user

I-CATIN	F-CHANGE-ATTRIB indication primitive
I-CATCF	F-CHANGE-ATTRIB confirm primitive
I-CLOIN	F-CLOSE indication primitive
I-CLOCF	F-CLOSE confirm primitive
I-CREIN	F-CREATE indication primitive
I-CRECF	F-CREATE confirm primitive
I-DELIN	F-DELETE indication primitive
I-DELCF	F-DELETE confirm primitive
I-DESIN	F-DESELECT indication primitive
I-DESCF	F-DESELECT confirm primitive
I-ERAIN	F-ERASE indication primitive
I-ERACF	F-ERASE confirm primitive
I-LOCIN	F-LOCATE indication primitive
I-LOCCF	F-LOCATE confirm primitive
I-OPNIN	F-OPEN indication primitive
I-OPNCF	F-OPEN confirm primitive
I-PABIN	F-P-ABORT indication primitive
I-RATIN	F-READ-ATTRIB indication primitive
I-RATCF	F-READ-ATTRIB confirm primitive
I-RECIN	F-RECOVER indication primitive
I-RECCF	F-RECOVER confirm primitive
I-SELIN	F-SELECT indication primitive
I-SELCF	F-SELECT confirm primitive

A.3.3.3 Outgoing events to the presentation service provider

P-ALTRQ	P-ALTER-CONTEXT request primitive
P-ALTRP	P-ALTER-CONTEXT response primitive

A.3.3.4 Outgoing events to the local system

L-ERRCTX	Local signal indicating P-ALTER-CONTEXT error

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.4 Specific actions — file regime management

[1] Send the PDU constructed as user data and map parameters on the appropriate ACSE form.

[2] Add a PDU to the current PSDU and terminate the current PSDU.

[8] Includes a state result parameter indicating failure.

[9] Set the remove contexts parameter of the F-OPEN or the F-RECOVER request PDU.

[10] Determine, on the basis of the abstract syntax derived from the contents type supplied by the external file service user, the set of presentation contexts necessary to support the file contents. Set the p-alter indicator if the necessary context needs to be defined. All contexts identified in the remove context parameter of the OPNRQ or RECRQ PDUs shall be deleted unless required in the new open regime; optionally the responder may elect to delete any other contexts in the defined context set which are not required.

[11] Set the presentation-action parameter of the OPNRP or RECRP PDU if a P-ALTER-CONTEXT exchange is needed.

[15] Preserve the PDU or PDUs for further processing.

[28] Set the bulk data transfer number to that specified in the F-RECOVER request primitive or PDU.

[31] Set the bulk data transfer number to zero.

[32] Erase the expected response list. Set the threshold indicator to the value given by the primitive parameter. Add an F-BEGIN-GROUP request PDU to the current PSDU.

[33] Add a request PDU corresponding to the primitive function to the current PSDU, and add the corresponding element to the expected response list.

[34] Add an F-END-GROUP request PDU to the current PSDU and terminate the PSDU.

[35] Check that every PDU in the PDU-Group is one of SELRP, CRERP, CLORP, RATRP, CATRP, DESRP, DELRP, or OPNRP; and that the composition and ordering of the PDUs is consistent with the group specified by the expected response list (recognizing that the response group may be truncated because of a detected error).

[36] Check that the number of PDUs in the PDU-Group is consistent with the value of the threshold parameter and that the other parameter values are consistent with negotiation.

[37] Considering each PDU in the PDU-Group, issue the corresponding confirm primitive to the IFS user with parameters derived from the PDU.

[38] Add a response PDU corresponding to the primitive function to the current PSDU.

[39] Add an F-END-GROUP response PDU to the current PSDU and terminate the PSDU.

[40] Check that the composition and ordering of the PDU-Group is consistent with one of the concatenated sequences "A", "C", or "D", as defined in Part 3, Annex E.

[41] Check that the composition and ordering of the PDU-Group is consistent with the concatenated sequence "E" as defined in Part 3, Annex E.

[42] Check that the composition and ordering of the PDU-Group is consistent with the concatenated sequence "B" as defined in Part 3, Annex E.

[43] Considering in turn each PDU in the PDU-Group, issue the corresponding indication primitive to the IFS user with parameters derived from the PDU.

[45] Set the checkpoint identifier expected to the value negotiated plus one.

[49] Set the synchronization offset to the value of the synchronization point serial number minus the checkpoint identifier expected.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.5 Predicates — file regime management

P4: The state result parameter on the incoming response PDU or response primitive indicates successful execution of operation.

P6: The P-ALTER-CONTEXT negotiation fails.

P7: A group of PDUs is preserved.

P9: The presentation context management functional unit is available and there are presentation contexts to be deleted or defined.

P10: The presentation-action parameter in the OPNRP PDU or in the RECRP PDU is TRUE, in which case the presentation context management functional unit should be available.

P11: The p-alter indicator is set.

P17: The PDU-Group contains a SELRP or CRERP PDU with a state result parameter indicating failure, or a DESRP or DELRP PDU.

P18: Checking the PDU-Group reveals a protocol error.

P19: The PDU-Group contains an OPNRP PDU with a state result parameter indicating success.

P43: The recovery procedure is in progress.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.6 Initiating entity state table — file regime management

STATE / EVENT	INITIALIZED	SELECT-PD	SELECTED	DESELECT-PD	CREATE-PD	DELETE-PD	READ-ATT-PD	CHG-ATT-PD	OPEN-PD	P-ALTIN-PD	DXFRIDLE	DXFRIDLE-REC	CLOSE-PD	RECOVER-PD	P-ALTIN-REC-PD	LOCATE-PD	ERASE-PD	GROUPING	GROUP-PD
I-SELRQ	1																	28	
SELRP		2																	
I-DESRQ			3															28	
DESRP				4															
I-CRERQ	5																	28	
CRERP					6														
I-DELRQ			7															28	
DELRP						8													
I-RATRQ			9															28	
RATRP							10												
I-CATRQ			11															28	
CATRP								12											
I-OPNRQ			13															22	
OPNRP									14										
P-ALTIN										15					21				
I-CLORQ											17	17						28	
CLORP													18						
I-RECRQ	19																		
RECRP														20					
I-LOCRQ											23	23							
LOCRP																24			
I-ERARQ											25	25							
ERARP																	26		
I-BGPRQ	27		27								27	27							
I-EGPRQ																		29	
GRPRP																			30

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.7 Initiating entity state table (Part 2) — detailed entries

1:		SELRQ[2]	⇒ SELECT-PD
2:	P4: ~P4:	I-SELCF I-SELCF[8]	⇒ SELECTED ⇒ INITIALIZED
3:		DESRQ[2]	⇒ DESELECT-PD
4:		I-DESCF	⇒ INITIALIZED
5:		CRERQ[2]	⇒ CREATE-PD
6:	P4: ~P4:	I-CRECF I-CRECF[8]	⇒ SELECTED ⇒ INITIALIZED
7:		DELRQ[2]	⇒ DELETE-PD
8:		I-DELCF	⇒ INITIALIZED

9:		RATRQ[2]	⇒ READ-ATT-PD
10:		I-RATCF	⇒ SELECTED
11:		CATRQ[2]	⇒ CHG-ATT-PD
12:		I-CATCF	⇒ SELECTED
13:	P9:	[9],	
		OPNRQ[2],[31]	⇒ OPEN-PD
14:	~P4:	I-OPNCF[8]	⇒ SELECTED
	P4 & ~P10:	I-OPNCF	⇒ DXFRIDLE
	P4 & P10:	[15]	⇒ P-ALTIN-PD
15:	P6:	L-ERRCTX,	
		P-ALTRP,	
	P7:	[37]	⇒ DXFRIDLE
	~P7:	I-OPNCF	⇒ DXFRIDLE
17:		CLORQ[2]	⇒ CLOSE-PD
18:		I-CLOCF	⇒ SELECTED
19:	P9:	[9],	
		RECRQ[2],[28]	⇒ RECOVER-PD
20:	~P4:	I-RECCF[8]	⇒ INITIALIZED
	P4 & ~P10:	[45],I-RECCF	⇒ DXFRIDLE-REC
	P4 & P10:	[15]	⇒ P-ALTIN-REC-PD
21:	P6:	L-ERRCTX,	
		P-ALTRP,	
		[45],I-RECCF	⇒ DXFRIDLE-REC
22:	P9:	[9],	
		[31],[33]	⇒ *same state*
23:		LOCRQ[2]	⇒ LOCATE-PD
24:	~P43:	I-LOCCF	⇒ DXFRIDLE
	P43:	I-LOCCF	⇒ DXFRIDLE-REC
25:		ERARQ[2]	⇒ ERASE-PD
26:	~P43:	I-ERACF	⇒ DXFRIDLE
	P43:	I-ERACF	⇒ DXFRIDLE-REC
27:		[32]	⇒ GROUPING
28:		[33]	⇒ *same state*
29:		[34]	⇒ GROUP-PD
30:		[35],[36],	
	P18:	I-PABIN,PABRQ[1]	⇒ UNINITIALIZED
	P17 & ~P18:	[37]	⇒ INITIALIZED
	~P17 & ~P18 & P19 & ~P10:	[37]	⇒ DXFRIDLE
	~P17 & ~P18 & P19 & P10:	[15]	⇒ P-ALTIN-PD
	~P17 & ~P18 & ~P19:	[37]	⇒ SELECTED

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.3.8 Responding entity state table — file regime management

STATE EVENT	INITIALIZED	I-SELECT-PD	SELECTED	I-DESELECT-PD	I-CREATE-PD	I-DELETE-PD	I-READ-ATT-PD	I-CHG-ATT-PD	I-OPEN-PD	P-ALTCF-PD	DXFRIDLE	DXFRIDLE-REC	I-CLOSE-PD	I-RECOVER-PD	P-ALTCF-REC-PD	I-LOCATE-PD	I-ERASE-PD	I-GROUP-PD	GROUPING
SELRQ	1																		
I-SELRP		2																	32
DESRQ			3																
I-DESRP				4															32
CRERQ	5																		
I-CRERP					6														32
DELRQ			7																
I-DELRP						8													32
RATRQ			9																
I-RATRP							10												32
CATRQ			11																
I-CATRP								12											32
OPNRQ			13																
I-OPNRP									14										22
P-ALTCF										15					21				
CLORQ											17	17							
I-CLORP													18						32
RECRQ	19																		
I-RECRP														20					
LOCRQ											24	24							
I-LOCRP																25			
ERARQ											26	26							
I-ERARP																	27		
I-BGPRP																		31	
I-EGPRP																			33
GRPRQ	28		29								30	30							

A.3.9 Responding entity state table (Part 2) — detailed entries

1:		I-SELIN	⇒ I-SELECT-PD
2:	P4: ~P4:	SELRP[2] SELRP[8][2]	⇒ SELECTED ⇒ INITIALIZED
3:		I-DESIN	⇒ I-DESELECT-PD
4:		DESRP[2]	⇒ INITIALIZED
5:		I-CREIN	⇒ I-CREATE-PD
6:	P4: ~P4:	CRERP[2] CRERP[8][2]	⇒ SELECTED ⇒ INITIALIZED
7:		I-DELIN	⇒ I-DELETE-PD
8:		DELRP[2]	⇒ INITIALIZED

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

9:		I-RATIN	⇒ I-READ-ATT-PD
10:		RATRP[2]	⇒ SELECTED
11:		I-CATIN	⇒ I-CHG-ATT-PD
12:		CATRP[2]	⇒ SELECTED
13:		I-OPNIN,[31]	⇒ I-OPEN-PD
14:	~P4:	OPNRP[8][2]	⇒ SELECTED
	P4:	[10],	
	P4 & ~P11:	OPNRP[2]	⇒ DXFRIDLE
	P4 & P11:	[11],OPNRP[2],P-ALTRQ	⇒ P-ALTCF-PD
15:	P6:	L-ERRCTX,	
			⇒ DXFRIDLE
17:		I-CLOIN	⇒ I-CLOSE-PD
18:		CLORP[2]	⇒ SELECTED
19:		I-RECIN,[28]	⇒ I-RECOVER-PD
20:	~P4:	RECRP[8][2]	⇒ INITIALIZED
	P4:	[10],[45],	
	P4 & ~P11:	RECRP[2]	⇒ DXFRIDLE-REC
	P4 & P11:	[11],RECRP[2],P-ALTRQ	⇒ P-ALTCF-REC-PD
21:	P6:	L-ERRCTX,	
			⇒ DXFRIDLE-REC
22:		[38],[31]	⇒ *same state*
24:		I-LOCIN	⇒ I-LOCATE-PD
25:	~P43:	LOCRP[2]	⇒ DXFRIDLE
	P43:	LOCRP[2]	⇒ DXFRIDLE-REC
26:		I-ERAIN	⇒ I-ERASE-PD
27:	~P43:	ERARP[2]	⇒ DXFRIDLE
	P43:	ERARP[2]	⇒ DXFRIDLE-REC
28:		[40],	
	P18:	I-PABIN,PABRQ[1]	⇒UNINITIALIZED
	~P18:	[43]	⇒ I-GROUP-PD
29:		[41],	
	P18:	I-PABIN,PABRQ[1]	⇒UNINITIALIZED
	~P18:	[43]	⇒ I-GROUP-PD
30:		[42],	
	P18:	I-PABIN,PABRQ[1]	⇒UNINITIALIZED
	~P18:	[43]	⇒ I-GROUP-PD
31:		[38]	⇒ GROUPING
32:		[38]	⇒ *same state*
33:		[39],	
	P17:		⇒ INITIALIZED
	~P17 & P19:	[10],	
	~P17 & P19 & ~P11:		⇒ DXFRIDLE
	~P17 & P19 & P11:	P-ALTRQ	⇒ P-ALTCF-PD
	~P17 & ~P19:		⇒ SELECTED

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4 Bulk data transfer protocol machine

A.4.1 States — bulk data transfer

DXFRIDLE	Data transfer idle
READ	Read data transfer
I-READ-ENDING	Read transfer end, wait for F-TRANSFER-END request primitive from the internal file service user
READ-ENDING	Read transfer ending, wait for transfer end request PDU
I-R-XFR-ENDING	Wait for F-TRANSFER-END response primitive after READ from the internal file service user
R-XFER-ENDING	Wait for transfer-end response PDU after READ
WRITE	Write data transfer
I-WRITE-ENDING	Write transfer-end, wait for F-TRANSFER-END request primitive from the internal file service user
WRITE-ENDING	Write transfer-end, wait for transfer end request PDU
I-W-XFR-ENDING	Wait for F-TRANSFER-END response primitive after WRITE from the internal file service user
W-XFER-ENDING	Wait for transfer-end response PDU after WRITE
CANCEL-PD	Cancel pending, wait for cancel response PDU
I-CANCEL-PD	Cancel pending, wait for F-CANCEL response primitive from the internal file service user
RRESTART-PD	Read restart pending, wait for restart response PDU in read operation
I-RRESTART-PD	Read restart pending, wait for F-RESTART response primitive in read operation from the internal file service user
WRESTART-PD	Write restart pending, wait for restart response PDU in write operation
I-WRESTART-PD	Write restart pending, wait for F-RESTART response primitive in write operation from the internal file service user
P-TOKEN-PD	Wait for sync-minor token
DXFRIDLE-REC	Data transfer idle during recovery
REA-SYMIN-PD	Wait for first P-SYNC-MINOR indication primitive after READ
RES-SYMIN-PD	Wait for first P-SYNC-MINOR indication primitive, restart requested
CAN-SYMIN-PD	Wait for first P-SYNC-MINOR indication primitive, cancel requested
WRT-SYMIN-PD	Wait for first P-SYNC-MINOR indication primitive after WRITE
REA-SYMCF-PD	Wait for first P-SYNC-MINOR confirm primitive after READ
RES-SYMCF-PD	Wait for first P-SYNC-MINOR confirm primitive, restart requested
CAN-SYMCF-PD	Wait for first P-SYNC-MINOR confirm primitive, cancel requested
WRT-SYMCF-PD	Wait for first P-SYNC-MINOR confirm primitive after WRITE
RESTART-CAN-PD	Restart cancel pending, wait for cancel or restart response PDU

A.4.2 Incoming events — bulk data transfer

A.4.2.1 Incoming events — PDUs

CANRP	Cancel response PDU (in user data of P-RESYNC(abandon) confirm if resync functional unit has been negotiated).	U2,U3
CANRQ	Cancel request PDU (in user data of P-RESYNC(abandon) indication if resync functional unit has been negotiated).	U2,U3
DAERQ	Data end request PDU.	U2,U3
TRERP	Transfer end response PDU.	U2,U3
TRERQ	Transfer end request PDU.	U2,U3
RESRQ	Restart request PDU, in user data of P-RESYNC (restart) indication.	U10
RESRP	Restart response PDU, in user data of P-RESYNC (restart) confirm.	U10
REARQ	Read request PDU.	U2
WRTRQ	Write request PDU.	U3
DATIN	data value in user context (not syntactically classed as a PDU).	U2,U3

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4.2.2 Incoming events from the internal file service user

I-CANRP	F-CANCEL response primitive.
I-CANRQ	F-CANCEL request primitive.
I-CHKRQ	F-CHECK request primitive.
I-CHKRP	F-CHECK response primitive.
I-DATRQ	F-DATA request primitive.
I-DAERQ	F-DATA-END request primitive.
I-REARQ	F-READ request primitive.
I-RESRQ	F-RESTART request primitive.
I-RESRP	F-RESTART response primitive.
I-TRERQ	F-TRANSFER-END request primitive.
I-TRERP	F-TRANSFER-END response primitive.
I-WRTRQ	F-WRITE request primitive.

A.4.2.3 Incoming events from the presentation service provider

P-SYMIN	P-SYNC-MINOR indication primitive.
P-SYMCF	P-SYNC-MINOR confirm primitive.
P-TOKIN	P-TOKEN-GIVE indication with minor-sync. token.

A.4.3 Outgoing events — bulk data transfer

A.4.3.1 Outgoing events — PDUs

CANRP	Cancel response PDU (in user data of P-RESYNC(abandon) response if resync functional unit has been negotiated)
CANRQ	Cancel request PDU (in user data of P-RESYNC(abandon) request if resync functional unit has been negotiated)
DAERQ	Data end request PDU
TRERP	Transfer end response PDU
TRERQ	Transfer end request PDU
RESRQ	Restart request PDU, in user data of P-RESYNC (restart) request
RESRP	Restart response PDU, in user data of P-RESYNC (restart) response
REARQ	Read request PDU
WRTRQ	Write request PDU

A.4.3.2 Outgoing events to the internal file service user

I-CANCF	F-CANCEL confirm primitive
I-CANIN	F-CANCEL indication primitive
I-CHKIN	F-CHECK indication primitive
I-CHKCF	F-CHECK confirm primitive
I-DATIN	F-DATA indication primitive
I-DAEIN	F-DATA-END indication primitive
I-REAIN	F-READ indication primitive
I-RESIN	F-RESTART indication primitive
I-RESCF	F-RESTART confirm primitive
I-TREIN	F-TRANSFER-END indication primitive
I-TRECF	F-TRANSFER-END confirm primitive
I-WRTIN	F-WRITE indication primitive

A.4.3.3 Outgoing events to the presentation service provider

P-SYMRQ	P-SYNC-MINOR request primitive
P-SYMRP	P-SYNC-MINOR response primitive
P-DATRQ	P-DATA request primitive
P-TOKRQ	P-TOKEN-GIVE request primitive with minor-synch token.

A.4.3.4 Outgoing events to the local system

L-ERRABT	Local signal indicating an error leading to abort

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4.4 Specific actions — bulk data transfer

[2] Add a PDU to the current PSDU and terminate the current PSDU.

[16] Add a PDU to the current PSDU and optionally terminate the current PSDU, according to local decision.

[15] Preserve the PDU for further processing

[17] Use for the sync point, the checkpoint received plus the offset.

[18] Add the data given on the F-DATA request to the current PSDU. Optionally (depending upon local system considerations) terminate the PSDU.

[19] Use for the checkpoint, the sync point number received minus the offset.

[21] Unset discard indicator

[22] Set discard indicator and set the outstanding checkpoint counter to zero.

[23] Discard any user data not yet delivered.

[24] Add 1 to the sync point number.

[25] Terminate the current PSDU.

[26] If Presentation resynchronize functional unit is available, send PDU as user data on a P-RESYNC(abandon) request or response primitive; otherwise add PDU to the current PSDU and terminate the current PSDU.

[27] Increment the checkpoint identifier expected.

[29] Increment the outstanding checkpoint counter.

[30] Decrement the outstanding checkpoint counter according to the checkpoint number confirmed (see 15.3.2).

[44] Increment the bulk data transfer number.

[45] Set the checkpoint identifier expected to the value negotiated plus one.

[47] Preserve the primitive parameters for subsequent processing. In some states, it happens that, when F-CANCEL primitive is preserved, there is already an F-RESTART primitive preserved: in this case, the F-RESTART primitive shall be discarded.

[48] Set the checkpoint identifier expected to 0.

[49] Set the synchronization offset to the value of the synchronize point serial number minus the checkpoint identifier expected. (In a bulk data transfer that does not involve recovery, the checkpoint identifier is zero, at this point in the protocol).

[50] PDU is sent as user data on a P-RESYNCH (restart) request or response primitive, with a synchronization point serial number equal to the sum of the checkpoint identifier, on the F-RESTART primitive, and the synchronization offset.

[51] The checkpoint identifier shall be equal to the value received on the PDU.

[52] Set the synch point serial number equal to the synch point serial number negotiated with the presentation service provider.

[53] Store the synchronization point serial number provided by the presentation service provider.

A.4.5 Predicates — bulk data transfer

P8: The context is not within the defined context set.

P13: The transfer service class has been negotiated and the bulk data transfer number is greater than 1.

P14: The synchronization point exceeds 999 998, exceeding the session service limitation.

P15: Discard indicator is set

P20: The resync functional unit has been negotiated.

P21: The sync-minor functional unit has been negotiated.

P22: The entity possesses the sync-minor token.

P27: The outstanding checkpoint counter exceeds the agreed maximum.

P41: The expected checkpoint identifier equals the sync point number minus the offset.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4.6 Initiating state table — bulk data transfer

STATE / EVENT	DXFRIDLE	DXFRIDLE-REC	REA-SYMIN-PD	RES-SYMIN-PD	CAN-SYMIN-PD	READ	I-READ-ENDING	R-XFER-ENDING	P-TOKEN-PD	WRT-SYMCF-PD	RES-SYMCF-PD	CAN-SYMCF-PD	WRITE	I-WRITE-ENDING	W-XFER-ENDING	CANCEL-PD	I-CANCEL-PD	R-RESTART-PD	W-RESTART-PD	I-R-RESTART-PD	I-W-RESTART-PD	RESTART-CAN-PD
I-REARQ	1	35																				
I-WRTRQ	2	36																				
P-TOKIN	42								3													
I-DATRQ										4			4									
I-DAERQ										5			5									
DATIN						6										41		41				
DAERQ						8										41		41				
I-TRERQ							9							10								
TRERP								11							3							
I-CANRQ			31	31		12	12			34	34		12					38	38	12	12	
CANRP																13						
CANRQ						14							14	14	14	15		14	14	14	14	39
I-CANRP																	16					
I-CHKRQ										17			17									
P-SYMCF										32	22	12	18	37	37							
P-SYMIN			29	21	12	19																
I-CHKRP						20	20															
I-RESRQ			30			21	21			33			22									
RESRP																		23	24			40
RESRQ						25							26	26	26			43	44			
I-RESRP																				27	28	

A.4.7 Initiating entity state table (Part 3) — detailed entries

1:		[44],	
	~P13 & P21:	REARQ[2],P-TOKRQ,[48],[49]	⇒ REA-SYMIN-PD
	~P13 & ~P21:	REARQ[2]	⇒ READ
	P13:	L-ERRABT	⇒ *same state*
2:		[44],	
	P13:	L-ERRABT	⇒ *same state*
	~P13 & ~P21:	WRTRQ[16]	⇒ WRITE
	~P13 & P21:	WRTRQ[16],[24],	
	~P13 & P14 & P21:	L-ERRABT	⇒ *same state*
	~P13 & ~P14 & P21:	P-SYMRQ,[48],[49]	⇒ WRT-SYMCF-PD
3:		I-TRECF	⇒ DXFRIDLE
4:	~P8:	P-DATRQ[18]	⇒ *same state*
	P8:	L-ERRABT	⇒ *same state*
5:		DAERQ[16]	⇒ I-WRITE-ENDING

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

6:	P15:		⇒ *same state*
	~P15:	I-DATIN	⇒ *same state*
8:	P15:		⇒ *same state*
	~P15:	I-DAEIN	⇒ I-READ-ENDING
9:		TRERQ[2]	⇒ R-XFER-ENDING
10:		TRERQ[2]	⇒ W-XFER-ENDING
11:	P21:	[15]	⇒ P-TOKEN-PD
	~P21:	I-TRECF	⇒ DXFRIDLE
12:		[22],[23],CANRQ[26]	⇒ CANCEL-PD
13:	P20:	[53],	
		[21],I-CANCF	⇒ DXFRIDLE
14:	P20:	[53],	
		[22],[23],I-CANIN	⇒ I-CANCEL-PD
15:	P20:	CANRP[26],[53],	
		[21],I-CANCF	⇒ DXFRIDLE
16:	P20:	[52],	
		CANRP[26],[21]	⇒ DXFRIDLE
17:		[25],[27],[24],	
	~P14:	P-SYMRQ	⇒ *same state*
	P14:	L-ERRABT	⇒ *same state*
18:		I-CHKCF[19]	⇒ *same state*
19:		[29],	
	~P27 & P41:	I-CHKIN,[24],[27]	⇒ *same state*
	P27 | ~P41:	L-ERRABT	⇒ *same state*
20:		[30],P-SYMRP[17]	⇒ *same state*
21:		[22],RESRQ[50]	⇒ RRESTART-PD
22:		[22],RESRQ[50]	⇒ WRESTART-PD
23:		[52],[45],[49],[27],[21],I-RESCF	⇒ READ
24:		[52],[45],[49],[27],[21],I-RESCF	⇒ WRITE
25:		[23],I-RESIN	⇒ I-RRESTART-PD
26:		I-RESIN	⇒ I-WRESTART-PD
27:		[52],[45],[49],[27],RESRP[50]	⇒ READ

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

28:		[52],[45],[49],[27],RESRP[50]	⇒ WRITE
29:		[24],[49],[27],P-SYMRP	⇒ READ
30:		[47],[22]	⇒ RES-SYMIN-PD
31:		[47],[22]	⇒ CAN-SYMIN-PD
32:		[49],[27]	⇒ WRITE
33:		[47],[22]	⇒ RES-SYMCF-PD
34:		[47],[22]	⇒ CAN-SYMCF-PD
35:	P21:	REARQ[2],P-TOKRQ,[27],[49]	⇒ REA-SYMIN-PD
	~P21:	REARQ[2]	⇒ READ
36:		WRTRQ[2],[24],	
	P14:	L-ERRABT	⇒ *same state*
	P21 & ~P14:	P-SYMRQ,[27],[49]	⇒ WRT-SYMCF-PD
	~P21 & ~P14:		⇒ WRITE
37:		I-CHKCF[19]	⇒ *same state*
38:		[47]	⇒ RESTART-CAN-PD
39:	P20:	[52], [21],I-CANCF,CANRP[26]	⇒ DXFRIDLE
40:		CANRQ[26]	⇒ CANCEL-PD
41:			⇒ *same state*
42:			⇒ *same state*
43:		[52],[45],[49],[27],[21],I-RESCF,RESRP[50]	⇒ READ
44:		[52],[45],[49],[27],[21],I-RESCF,RESRP[50]	⇒ WRITE

NOTE - The entries 38, 39, 40 are due to the Session service behaviour in the resynchronize management, in collision cases.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4.8 Responding entity state table — bulk data transfer

STATE EVENT	DXFR-IDLE	DXFR-IDLE-REC	P-TOKEN-PD	REA-SYMCF-PD	RES-SYMCF-PD	CAN-SYMCF-PD	READ	READ-ENDING	I-RXFR-ENDING	WRT-SYMIN-PD	RES-SYMIN-PD	CAN-SYMIN-PD	WRITE	WRITE-ENDING	I-WXFR-ENDING	CANCEL-PD	I-CANCEL-PD	RRESTART-PD	WRESTART-PD	I-RRESTART-PD	I-WRESTART-PD	RESTART-CAN-PD
REARQ	1	36																				
WRTRQ	2	37																				
P-TOKIN			3																			
I-DATRQ				4			4															
I-DAERQ				5			5															
DATIN													6			42			42			
DAERQ													8			42			42			
TRERQ								9						10								
I-TRERP									11						43							
I-CANRQ				30	30		12			33	33		12	12	12			39	39	12	12	
CANRP																13						
CANRQ				14	14		14	14					14			44		14	14	14	14	40
I-CANRP																	15					
I-CHKRQ				16			16															
P-SYMCF				28	20	12	18	38														
P-SYMIN										31	34	35	17									
I-CHKRP													19	19								
I-RESRQ				29			20			32			21	21	21							
RESRP																		22	23			41
RESRQ				24			24	24	24				25					45	46			
I-RESRP																				26	27	

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.4.9 Responding entity state table (Part 3) — detailed entries

1:		[44],	
	~P13 & P21:	[15],[48],[49]	⇒ P-TOKEN-PD
	~P13 & ~P21:	I-REAIN	⇒ READ
	P13:	L-ERRABT	⇒ *same state*
2:		[44],	
	~P13 & P21:	I-WRTIN,[48],[49]	⇒ WRT-SYMIN-PD
	~P13 & ~P21:	I-WRTIN	⇒ WRITE
	P13:	L-ERRABT	⇒ *same state*
3:		I-REAIN,[24],	
	~P14:	P-SYMRQ	⇒ REA-SYMCF-PD
	P14:	L-ERRABT	⇒ *same state*
4:	P8:	P-DATRQ[18]	⇒ *same state*
	~P8:	L-ERRABT	⇒ *same state*
5:		DAERQ[16]	⇒ READ-ENDING

6:	P15:		⇒ *same state*
	~P15:	I-DATIN	⇒ *same state*
8:	P15:		⇒ *same state*
	~P15:	I-DAEIN	⇒ WRITE-ENDING
9:	P15:		⇒ *same state*
	~P15:	I-TREIN	⇒ I-R-XFR-ENDING
10:	P15:		⇒ *same state*
	~P15:	I-TREIN	⇒ I-W-XFR-ENDING
11:	P21:	TRERP[2],P-TOKRQ	⇒ DXFRIDLE
	~P21:	TRERP[2]	⇒ DXFRIDLE
12:		[22],[23],CANRQ[26]	⇒ CANCEL-PD
13:	P21 & P22:	P-TOKRQ	⇒ DXFRIDLE
	P20 & (~P21 | ~P22):	[53],	
	~P21 | ~P22:	[21],I-CANCF	⇒ DXFRIDLE
14:		[53],[22],[23],I-CANIN	⇒ I-CANCEL-PD
15:	P21 & P22:	P-TOKRQ,	
	P20:	[52],	
		CANRP[26],[21]	⇒ DXFRIDLE
16:		[25],[27],[24],	
	~P14:	P-SYMRQ	⇒ *same state*
	P14:	L-ERRABT	⇒ *same state*
17:		[29],	
	~P27 & P41:	I-CHKIN,[24],[27]	⇒ *same state*
	P27 | ~P41:	L-ERRABT	⇒ *same state*
18:		I-CHKCF[19]	⇒ *same state*
19:		[30],P-SYMRP[17]	⇒ WRITE
20:		[22],RESRQ[50]	⇒ RRESTART-PD
21:		[22],RESRQ[50]	⇒ WRESTART-PD
22:		[52],[45],[49],[27],[21],I-RESCF	⇒ READ
23:		[52],[45],[49],[27],[21],I-RESCF	⇒ WRITE
24:		I-RESIN	⇒ I-RRESTART-PD
25:		[23],I-RESIN	⇒ I-WRESTART-PD
26:		[52],[45],[49],[27],RESRP[50]	⇒ READ

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

27:		[52],[45],[49],[27],RESRP[50]	⇒ WRITE
28:		[49],[27]	⇒ READ
29:		[22],[47]	⇒ RES-SYMCF-PD
30:		[22],[47]	⇒ CAN-SYMCF-PD
31:		[24],[49],[27],P-SYMRP	⇒ WRITE
32:		[22],[47]	⇒ RES-SYMIN-PD
33:		[22],[47]	⇒ CAN-SYMIN-PD
34:		[24],P-SYMRP,[22],RESRQ[50]	⇒ WRESTART-PD
35:		[24],P-SYMRP,[22],CANRQ[26]	⇒ CANCEL-PD
36:	P21:	[15],[27],[49]	⇒ P-TOKEN-PD
	~P21:	I-REAIN	⇒ READ
37:	P21:	I-WRTIN,[27],[49]	⇒ WRT-SYMIN-PD
	~P21:	I-WRTIN	⇒ WRITE
38:		I-CHKCF[19]	⇒ *same state*
39:		[22],[47]	⇒ RESTART-CAN-PD
40:	P21 & P22:	P-TOKRQ,	
	P20:	[52],	
		[21],I-CANCF,CANRP[26]	⇒DXFRIDLE
41:		CANRQ[26]	⇒ CANCEL-PD
42:			⇒ CANCEL-PD
43:		TRERP[2]	⇒ DXFRIDLE
44:	P21 & P22:	P-TOKRQ,	
	P20:	[53],[52],	
		CANRP[26],[21],I-CANCF	⇒ DXFRIDLE
45:		[52],[45],[49],[27],[21],I-RESCF[51],RESRQ[50]	⇒ READ
46:		[52],[45],[49],[27],[21],I-RESCF,RESRQ[50]	⇒ WRITE

NOTE - The entries 14, 39, 40, 41, 45 and 46 are due to the Session service behaviour in the resynchronize management, in collision cases.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.5 File error recovery protocol machine (FERPM)

NOTE - The description of the file error recovery protocol machine given here omits the description of grouping, which does not materially affect the recovery behaviour. The transitions for the grouped cases can be derived from those in the basic protocol machine.

A.5.1 States — file error recovery

INIT-PD Initialize pending, wait for an I-INICF event as an F-INITIALIZE confirm primitive from the IFS.

PASSIVE The basic file protocol is currently in operation but no transfer of file contents is in progress. It is used when the FERPM becomes inactive in normal error free activity, or when it cannot provide its services any more after issuing an L-ERRABT signal.

XFER-IDLE Data transfer idle.

XFER Normal transfer of file contents has been established.

RESTART-PD Restart pending, wait for an I-RESCF event as an F-RESTART confirm primitive from the IFS.

RESTART A restart of a data transfer is in progress. A checkpoint identifier has been negotiated, and data following that checkpoint, but prior to the error, is being retransmitted.

CANCEL-PD Cancel pending, wait for an I-CANCF event as an F-CANCEL confirm primitive from the IFS.

CLOSE-EX Close expected, wait for an I-CLOIN event as an F-CLOSE indication primitive from the IFS.

DESELECT-EX Deselect expected, wait for an I-DESIN event as an F-DESELECT indication primitive from the IFS.

RECOVER-PD Recover pending, wait for an I-RECCF event as an F-RECOVER confirm primitive from the IFS.

SEL-PD SELECT pending, wait for an I-SELCF event as an F-SELECT confirm primitive from the IFS.

OPN-PD OPEN pending, wait for an I-OPNCF event as an F-OPEN confirm primitive form the IFS.

SEL-EX SELECT expected, wait for an I-SELIN event as an F-SELECT indication primitive from the IFS.

OPN-EX OPEN expected, wait for an I-OPNIN event as an F-OPEN indication primitive form the IFS.

XFER-EX Data transfer expected, wait for an I-REAIN/I-WRTIN as an F-READ/F-WRITE indication primitive from the IFS.

INIT-EX Initiation expected, wait for an I-ININ event as an F-INITIALIZE indication primitive from the IFS.

CLOSE-PD Close pending, wait for an I-CLOCF event as an F-CLOSE confirm primitive from the IFS.

DESELECT-PD Deselect pending, wait for an I-DESCF event as an F-DESELECT confirm primitive from the IFS.

A.5.2 Incoming events — file error recovery

A.5.2.1 Incoming events from the external file service user

F-OPNRQ	F-OPEN request primitive
F-CLORQ	F-CLOSE request primitive
F-REARQ	F-READ request primitive
F-WRTRQ	F-WRITE request primitive
F-DATRQ	F-DATA request primitive
F-DAERQ	F-DATA-END request primitive
F-TRERQ	F-TRANSFER-END request primitive
F-ANYRQ	[Any of those request primitives implied in 19.1.1a.]
F-OPNRP	F-OPEN response primitive
F-CLORP	F-CLOSE response primitive
F-TRERP	F-TRANSFER-END response primitive
F-ANYRP	[Any of those response primitives implied in 19.1.1a.]

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.5.2.2 Incoming events from the internal file service

I-INICF	F-INITIALIZE confirm primitive
I-OPNCF	F-OPEN confirm primitive
I-CLOCF	F-CLOSE confirm primitive
I-DATIN	F-DATA indication primitive
I-DAEIN	F-DATA-END indication primitive
I-CHKIN	F-CHECK indication primitive
I-CHKCF	F-CHECK confirm primitive
I-TRECF	F-TRANSFER-END confirm primitive
I-ANYCF	[Any of the confirm primitives implied in 19.1.1a.]
I-SELCF	F-SELECT confirm primitive
I-DESCF	F-DESELECT confirm primitive
I-INIIN	F-INITIALIZE indication primitive
I-SELIN	F-SELECT indication primitive
I-DESIN	F-DESELECT indication primitive
I-OPNIN	F-OPEN indication primitive
I-CLOIN	F-CLOSE indication primitive
I-TREIN	F-TRANSFER-END indication primitive
I-CANIN	F-CANCEL indication primitive
I-CANCF	F-CANCEL confirm primitive
I-PABIN	F-P-ABORT indication primitive
I-RECCF	F-RECOVER confirm primitive
I-RESCF	F-RESTART confirm primitive
I-RECIN	F-RECOVER indication primitive
I-RESIN	F-RESTART indication primitive
I-ANYIN	[Any of the indication primitives implied in 19.1.1a.]
I-REAIN	F-READ indication primitive
I-WRTIN	F-WRITE indication primitive

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.5.2.3 Incoming events from the Local System Environment

L-ERROR1	Class I error (as defined in 18.1)
L-ERROR2	Class II error (as defined in 18.1)
L-ERROR3	Class III error (as defined in 18.1)
L-GIVEUP	Timer for recovery expires
L-DATRQ	Reissued F-DATA request primitive
L-CHKRQ	Reissued F-CHECK request primitive
L-DAERQ	Reissued F-DATA-END request primitive
L-EORIN	Indicates that all checkpoint identifiers and "data end" markers in the docket have been used, and all data to be resent have been sent.

A.5.3 Outgoing events — file error recovery

A.5.3.1 Outgoing events to the external file service user

F-OPNCF	F-OPEN confirm primitive
F-ANYCF	[Any of the confirm primitives implied in 19.1.1a]
F-CLOCF	F-CLOSE confirm primitive
F-DATIN	F-DATA indication primitive
F-DAEIN	F-DATA-END indication primitive
F-TRECF	F-TRANSFER-END confirm primitive
F-OPNIN	F-OPEN indication primitive
F-WRTIN	F-WRITE indication primitive
F-CLOIN	F-CLOSE indication primitive
F-REAIN	F-READ indication primitive
F-TREIN	F-TRANSFER-END indication primitive
F-ANYIN	[Any of the indication primitives implied in 19.1.1a]
F-PABIN	F-P-ABORT indication primitive

A.5.3.2 Outgoing events to the internal file service

I-SELRQ	F-SELECT request primitive
I-OPNRQ	F-OPEN request primitive
I-CLORQ	F-CLOSE request primitive
I-DATRQ	F-DATA request primitive
I-DAERQ	F-DATA-END request primitive
I-CHKRQ	F-CHECK request primitive
I-TRERQ	F-TRANSFER-END request primitive
I-REARQ	F-READ request primitive
I-WRTRQ	F-WRITE request primitive
I-ANYRQ	[Any of the request primitives implied in 19.1.1a]
I-INIRP	F-INITIALIZE response primitive
I-OPNRP	F-OPEN response primitive
I-SELRP	F-SELECT response primitive
I-CLORP	F-CLOSE response primitive
I-TRERP	F-TRANSFER-END response primitive
I-ANYRP	[Any of the response primitives implied in 19.1.1a]
I-CHKRP	F-CHECK response primitive
I-RESRQ	F-RESTART request primitive
I-RESRP	F-RESTART response primitive
I-CANRQ	F-CANCEL request primitive
I-CANRP	F-CANCEL response primitive
I-DESRQ	F-DESELECT request primitive
I-RECRQ	F-RECOVER request primitive
I-INIRQ	F-INITIALIZE request primitive
I-RECRP	F-RECOVER response primitive
I-DESRP	F-DESELECT response primitive

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988

A.5.3.3 Outgoing events to the local system environment

L-ERROR2 Signal indicating class II error

L-RESEND Resend data request to the local file system

L-SUSPND Suspend issuing of F-DATRQ, F-DAERQ by the local file system

L-RESUME Resume issuing of F-DATRQ, F-DAERQ by the local file system

L-ERRABT Signal FPM to issue an F-P-ABORT request PDU, with permanent error value.

L-PABORT Signal FPM to issue a F-P-ABORT request PDU, and a F-P-ABORT indication primitive to the FERPM, both with transient error value.

A.5.4 Specific actions — file error recovery

[54] Mark the checkpoint identifiers to be reissued.

[55] Delete the docket.

[56] Set the activity state indicator to "finished".

[57] Set the activity state indicator to "in-progress".

[58] Set the activity type indicator to "read".

[59] Set the activity type indicator to "write".

[60] Add the checkpoint identifier to the list in the docket. Increment the count of outstanding checkpoints by one.

[61] Delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued. Decrement the count of outstanding checkpoints by the number of the deleted identifiers.

[64] Set the activity state indicator to "data transfer finishing".

[66] Use the last checkpoint identifier in the docket as the parameter.

[67] Increment the checkpoint identifier count.

[68] Set the checkpoint identifier count to zero.

[69] Record the bulk data transfer number in the docket (the number is maintained by the FPM).

[70] Record which data values have been already delivered to the user.

[71] Use as the parameter the checkpoint identifier available in the docket that is both

a) after the checkpoint identifier negotiated, and
b) not yet reissued during this restart of the BDT.

[72] Use as parameter the checkpoint identifier count.

[73] Use as the parameter the oldest (first) checkpoint identifier in the list in the docket.

[74] Use as the parameter the checkpoint identifier of the primitive received.

[75] Use as the diagnostics parameter "activity identifier unknown".

[76] Record which data values have been sent.

[77] Create a docket and record in it the activity indentifier, the information needed to issue or check an I-INIRQ (including the locations of the initiator and responder), the recovery mode, the access context, the presentation context, a null checkpoint list and set the activity state indicator to "starting".

[78] The issuer of the I-RESRQ or I-RECRQ primitive identifies a checkpoint identifier which is

a) for the sender, the last point acknowledged,

b) for the receiver, the last point received and secured.

[79] The issuer of the I-RESRP primitive identifies a checkpoint identifer which is

c) for the sender, equal to the value provided by the issuer of the request, and

d) for the receiver, the last point received and secured.

STANDARDSISO.COM : Click to view the full PDF of ISO 8571-4:1988