INTERNATIONAL STANDARD

ISO 8571-3

First edition 1988-10-01

AMENDMENT 2 1993-08-15

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

iTeh Spart 3 DARD PREVIEW (File Service Definitioni)

AMENDIAENTn2:10verlapped access https://standards.iteh.ai/catalog/standards/sist/00c87397-4b7d-4892-9ae4-3a702ec18a3a/iso-8571-3-1988-amd-2-1993

Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Transfert, accès et gestion de fichiers –

Partie 3: Définition du service de fichiers

AMENDEMENT 2 : Chevauchement d'accès



Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national VIEW bodies casting a vote.

Amendment 2 to International Standard ISO 8571-3:1988 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology.

ISO 8571 consists of the following parts, under the general title Information processing systems – Open Systems Interconnection ¹² File Transfer, Access and ^{1b7d-4892-9ae4-} Management: 3a702ec18a3a/iso-8571-3-1988-amd-2-1993

- Part 1 : General introduction
- Part 2 : Virtual Filestore Definition
- Part 3 : File Service Definition
- Part 4 : File Protocol Specification
- Part 5 : Protocol Implementation Conformance Statement Proforma

© ISO/IEC 1993

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland Printed in Switzerland

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

Part 3: File Service Definition AMENDMENT 2 : Overlapped access

iTeh STANDARD PREVIEW

0 Introduction

(standards.iteh.ai) be overlapped, more than one association is necessary, The overlapped access takes place within a constant set of

presentation contexts established as at present when the file Clause 0 provides an introduction to this amendment. The is opened, for previously 4text in this clause is not intended for inclusion in ISO 8571 ds/sist a3a/iso-8571-3-1988-amd-2-1993 part 3.

0.1 General

ISO 8571 part 3 defines in an abstract way the externally visible file transfer, access and management service.

This amendment extends this service definition to incorporate the services offered by overlapped access.

0.2 Rationale

The objective in introducing overlapped access is to allow more efficient access to structured files when a single initiator has a need to perform many reading and updating operations; the serial nature of the current FTAM data transfer services introduces a significant control overhead if the FADUs are small. In this context, an FADU is small if its transmission time is comparable with the time to complete a confirmed service on the association (the association's round trip delay).

0.3 Summary

The current design envelope that there should be at most one file selection per association and one file open per file selection is maintained. If access to more than one file is to

Two different degrees of overlap have been identified. Firstly, requests for future accesses may be issued whilst a previously requested BDT action is in progress, allowing the creation of a queue of read and write requests. In general, PCI relating to a given BDT action may be overlapped with other BDT actions, subject to restrictions; this is called consecutive access. Secondly, read and write actions can be performed in parallel, so that both directions of data transfer are exploited at any one time. Requests are then taken from the queue whenever either direction of transfer becomes free; this is called concurrent access.

The transfer of a single FADU, specified in a single F-READ request has the same interpretation as in ISO 8571. The resultant effect on the virtual filestore of a set of overlapped requests using consecutive access shall be the same as that of the equivalent set of requests issued in series; the service provided is serializable. If concurrent access is used then the resultant effect of a set of write actions on the virtual filestore, is also serializable. However, due to the non-determinism introduced by the use of concurrent access, it is also possible that, in some uses of the service, the data transferred as a result of a read action is not consistent with the current state of the file.

1 Scope and field of application

This amendment makes no additions to clause 1.

2 References

This amendment makes no additions to clause 2.

3 Definitions

This amendment makes no additions to clause 3.

4 Abbreviations

This amendment makes no additions to clause 4.

5 Conventions

This amendment makes no additions to clause 5.

iTeh STANDARD PREVIEW (standards.iteh.ai)

<u>ISO 8571-3:1988/Amd 2:1993</u> https://standards.iteh.ai/catalog/standards/sist/00c87397-4b7d-4892-9ae4-3a702ec18a3a/iso-8571-3-1988-amd-2-1993

Section one: General

6 Model of the file service

6.2 File service levels

Replace second sentence of a):

Transfer of file data is modelled in the external file service as error free operations.

6.3 Regimes of the file service

Replace figure 2.

Replace d) and g):

d) the data transfer regime during which particular bulk data transfer specifications are in force.

g) a sequence of data transfer regimes within a file open regime; the data transfer regimes may each be for either read transfer, write transfer or both if overlapped access is R in use. Write data transfer permits the operations insert, replace, or extend. NOTE - If overlapped access is in use, then a data transfer regime is deemed to have terminated when there are no requests for bulk data transfer outstanding or in progress.

7 Services of the file service

7.8 Bulk data transfer

Replace sub-clause 7.8:

Bulk data transfer refers to the transfer, optionally with checkpointing, of one or more file access data units (see 20.1). There are nine additional services associated with bulk data transfer.

a) the consecutive access service may be requested by an initiator to allow the overlap of the protocol control information for read and write bulk data transfer procedures (see ISO 8571-1);

b) the concurrent access service may be requested by an initiator to allow the concurrent progression of read and write bulk data transfer procedures ((see ISO 8571-1)

Add note after last paragraph: https://standards.iteh.ai/catalog/standards/sistinitiator to initiate a bulk data transfer from the responder (in 3a702ec18a3a/iso-8571-3-1988-amd-2-1993



Figure 2 - File service regimes and related primitivees

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

| U15 Consecutive access | 0 | | see note |
|------------------------|---|--|----------|
| U16 Concurrent access | 0 | | see note |

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

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

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

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

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

h) the cancel data transfer service (see 24.6) is used by A either the sender or the receiver to cancel a data transfer procedure. When overlapped access is allowed either are direction may be cancelled independently of the other.

Add entries to Table 1 and re-number entries accordingly:

Add note after Table 1:

Note - the consecutive access and concurrent access functional units allow the services provided by the read and write functional units to be overlapped. The description of the overlap allowed is contained within the definitions of individual services.

8.2 Service classes

8.2.2 File access class

Insert the following items and re-label accordingly:

d) optionally, the consecutive access functional unit;

e) optionally, the concurrent access functional unit;

ISO 8571-3:1988/Amd 2:1993

7.10 Checkpointing and restarting lards.iteh.ai/catalog/standards/ist/00.87397,4674-4892-914 3a702ec18a3a/iso-8571-3-1988-amd-2-1993

Add sentence to end of b):

If overlapped access is in use then either direction of transfer may be restarted.

8 Functional units and service classes

8.1 Functional units

Add the following clauses:

8.1.15 Consecutive access functional unit

The consecutive access functional unit allows the overlap of read and write data transfer procedures as defined in ISO 8571-1.

8.1.16 Concurrent access functional unit

The concurrent access functional unit allows the overlap of read and write data transfer procedures as defined in ISO 8571-1.

8.1.17 Service classes and functional units

Replace Table 3 - Functional units in the file services

Insert note:

Note - Pending the specification of presentation symmetric synchronisation services, recovery mechanisms will not be available for use during overlapped access.

Table 3 - Functional units in the file service

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

10 Negotiation of service class, **FTAM** QoS and functional units

Amend third paragraph:

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

iTeh STANDARD PREVIEW (standards.iteh.ai)

<u>ISO 8571-3:1988/Amd 2:1993</u> https://standards.iteh.ai/catalog/standards/sist/00c87397-4b7d-4892-9ae4-3a702ec18a3a/iso-8571-3-1988-amd-2-1993

Section two : Definition of file service primitives

11 File service primitives

Amend entries in Table 6.

12 Sequence of primitives

This amendment makes no additions to clause 12.

13 Common file service parameters

13.8 Concurrency Control

Replace fifth paragraph:

If FADU locking is negotiated the locks set at file open have the scope of the relevant bulk data transfer procedure. If overlapped access is not in use then this is equivalent to the bulk data transfer regime.

If overlapped access is in use then the scope of the FADU locking is that of the read and write action.

13.12 File Access Data Unit identity

Add note.

Note: if concurrent access is in use then the FADU identifiers previous, current and next are dependent on whether the action is a read or a write.

Add the following clauses:

13.14 Bulk Transfer Number

The bulk transfer number parameter is only visible in the internal file service. The parameter is used to identify a particular bulk data transfer within an open regime. If the use of concurrent access is negotiated when the file open regime is established then the first read data transfer within an open regime is established then the first read data transfer within an open regime is numbered 1, and subsequent read transfers are consecutively numbered. Similarly, the first write data transfer within the open regime is numbered 1, and subsequent write transfers are consecutively numbered. If overlapped access is not negotiated, or if consecutive access is negotiated, then read and write bulk transfers are numbered consecutively, starting from 1 for (standard transfer bulk data transfer procedure i.e. there is one

| Primitive | Confirmed 3a702e | 18a3a/iso-8571-3-1988-an | d-2-1993 Parameters |
|-------------|------------------|--------------------------|---|
| F-OPEN | Yes | Initiator | State result Action result Processing mode Contents type Concurrency control Shared ASE information Enable FADU locking Diagnostic Degree of overlap Transfer window [Activity identifier] [Recovery mode] |
| [F-RECOVER] | Yes | Initiator | [State result Action result Activity identifier Bulk transfer number Requested access Access passwords Contents type Recovery point Diagnostic Concurrent bulk transfer number Concurrent recovery point Last transfer end read request Last transfer end read response Last transfer end write request Last transfer end write request |

Table 6 - File service primitives

numbering sequence.

13.15 Transfer Number

The transfer number parameter is conditional on the use of overlapped access. The parameter is used by the initiator and responder to identify an uncompleted bulk data transfer. The value of the parameter is an integer. If overlapped access is not in use, or consecutive access is in use, then overlapping read and write transfers are consecutively numbered, starting from 1. The value is reset when all outstanding requests are completed.

Note - The transfer number of a bulk data transfer may exceed the value of the transfer window (17.1.2.10) as the latter parameter constrains the maximum number of transfers that are uncompleted at any one time.

13.16 Last transfer end read request

The last transfer end read request parameter is only visible in the internal file service, and is conditional on the use of overlapped access for the open regime. The value of the parameter is a transfer number (see 13.14). For the initiator it indicates the transfer number of the last read data transfer for which a F-TRANSFER-END request has been issued; for the responder it indicates the transfer number of the last S.I read data transfer for which a F-TRANSFER-END indication has been received. ISO 8571-3:1988/Amd 2:1993

The default value 0 indicates that there are no outstanding 1 2 10 15 File selection regime control 3-1988-amd-2-1 F-TRANSFER-END requests or indications to be issued or received.

13.17 Last transfer end read response

The last transfer end read response parameter is only visible in the internal file service, and is conditional on the use of overlapped access for the open regime. The value of the parameter is a transfer number (see 13.13). For the initiator it indicates the transfer number of the last read data transfer for which a F-TRANSFER-END confirm has been received: for the responder it indicates the transfer number of the last read data transfer for which a F-TRANSFER-END response has been issued.

The default value 0 indicates that there are no outstanding F-TRANSFER-END responses or confirms to be issued or received.

13.18 Last transfer end write request

The last transfer end write request parameter is only visible in the internal file service, and is conditional on the use of overlapped access for the open regime. The value of the parameter is a transfer number (see 13.14). For the initiator it indicates the transfer number of the last write data transfer for which a F-TRANSFER-END request has been issued; for the responder it indicates the transfer number of the last write data transfer for which a F-TRANSFER-END indication has been received.

The default value 0 indicates that there are no outstanding F-TRANSFER-END requests or indications to be issued or received.

13.19 Last transfer end write response

The last transfer end write response parameter is only visible in the internal file service, and is conditional on the use of concurrent access for the open regime. The value of the parameter is a transfer number (see 13.14). For the initiator it indicates the transfer number of the last write data transfer for which a F-TRANSFER-END confirm has been received; for the responder it indicates the transfer number of the last write data transfer for which a F-TRANSFER-END response has been issued.

The default value zero indicates that there are no outstanding F-TRANSFER-END responses or confirms to be issued or received.

14 FTAM regime control

This amendment makes no additions to clause 14.

This amendment makes no additions to clause 15.

16 File management

This amendment makes no additions to clause 16.

17 File open regime control

Amend second sentence:

This regime establishes the degree of overlapped access, processing mode, presentation contexts, and concurrency control for the data transfer activity which is to be performed.

17.1 File open service

17.1.1 Function

Amend last sentence of first paragraph:

It also establishes concurrency control, the degree of overlapped access and possible processing modes.

17.1.2 Types of primitives and parameters

Replace entries in table 21.

Add the following sub clauses and re-number:

17.1.2.9 Degree Of Overlap

The degree of overlap parameter indicates the degree of overlapped access required during the file open regime: either no overlapped access (normal), consecutive access or concurrent access.

The degree of overlap available during the open regime is the lowest of the values offered by the initiator and responder. For the purposes of negotiation, the degrees of overlapped access are rated as: concurrent access (highest), consecutive access, no overlapped access (lowest). This value is used to set the current degree of overlap attribute.

17.1.2.10 Transfer Window

The transfer window parameter indicates the number of bulk data transfers that may be uncompleted at any time. There is an independent transfer window for reads and writes in concurrent access, although the value of the limit is the same, otherwise the transfer window is for both reads and writes.

The bulk transfer number is defined in 13.14. The parameter indicates the bulk transfer procedure that is to be recovered.

If concurrent access is in use then this parameter takes the value of the read transfer that is to be recovered (the write transfer is identified by the concurrent bulk transfer number parameter). The default value 0 indicates that there is no read bulk transfer to be recovered.

Add the following clauses.

19.1.2.9 Recovery Point

Add last paragraph.

If concurrent overlapped access is in use then the parameter is with respect to the read data transfer that is identified by the bulk data transfer number. If the bulk transfer number is zero then the recovery point parameter shall take the value zero.

19.1.2.10 Concurrent Bulk Transfer Number

The concurrent bulk transfer number parameter is conditional on the selection of the concurrent access functional unit and the degree of overlap (see 17.1.2.9) being set to concurrent access. The parameter takes the value of a bulk transfer number (defined in 13.14) and identifies the write transfer that is to be recovered (see 19.1.2.4).

ISO 8571-3:1988/Amd 2:1993

The value of the transfer window is the lowest of the values standar The default value 70_indicates that there is no write bulk offered by the initiator or responder. 3a702ec18a3a/iso-8571 transfer to be recovered.

18 Grouping control

This amendment makes no additions to Clause 18 of ISO 8571 part 3.

19 Recovery (Internal service only)

19.1 Regime recovery

19.1.2 Types of primitives and parameters

Replace table 24.

19.1.2.4 Bulk Transfer Number

19.1.2.11 Concurrent Recovery Point

The concurrent recovery point parameter is conditional on the concurrent access functional unit and the degree of overlap (see 17.1.2.9) being set to concurrent access. The parameter indicates that recovery is to a point before the start of the write bulk data transfer (value zero), or to a checkpoint within the bulk data transfer, or to a point following its completion (see note). The recovery point is within the write bulk data transfer identified by the concurrent bulk transfer number (see 19.1.2.10); if the write bulk transfer number parameter is zero then the write recovery point parameter shall take the value zero. The recovery point is determined by the entity which was receiving data at the time of failure. Consequently the parameter is required to be present on the response primitive.

| Degree of overlap | Conditional | Conditional (=) | Conditional | Conditional (=) |
|-------------------|-------------|--------------------|-------------|--------------------|
| Transfer window | Conditional | Conditional (=) | Conditional | Conditional (=) |

Table 21 - F-OPEN parameters

If the concurrent bulk transfer number is zero, the concurrent recovery point parameter shall take the value zero.

NOTE - Recovery to a point following a write bulk data transfer completion is defined for the case where the receiver has received an F-DATA-END indication but the F-TRANSFER-END exchange has not completed. In such a situation it would be unnecessary to go back to the last confirmed checkpoint. The only requirement is for both sender and receiver to have an agreed understanding of the completion of the transfer. A value one larger than the highest checkpoint number issued by the sender is used to indicate recovery after bulk data transfer.

19.1.2.12 Last Transfer End Read Request

The last transfer end read request parameter is defined in 13.16. It is only present on the response primitive, and therefore indicates the bulk transfer number of the last read data transfer for which a transfer end read indication primitive had been received by the responder. It is only present if recovery is to an open regime that had negotiated the use of concurrent access.

19.1.2.13 Last Transfer End Read Response R

The last transfer end read response parameter is defined in S. 13.17. It is only present on the request primitive, and therefore indicates the bulk transfer number of the last read data transfer for which a transfer end read confirm primitive 8/Ar

had been received by the initiator of ish only present if ds/sist/00.87397-467d-4892-9ac4recovery is to an open regime that had negotilated the use -3-1988-and -2-1995 and -2-1995 and

Amd 2:1993

19.1.2.14 Last Transfer End Write Request

The last transfer end write request parameter is defined in 13.18. It is only present on the response primitive, and therefore indicates the bulk transfer number of the last write data transfer for which a transfer end read indication primitive had been received by the responder. It is only present if recovery is to an open regime that had negotiated the use of concurrent access.

19.1.2.15 Last Transfer End Write Response

The last transfer end write response parameter is defined in 13.19. It is only present on the request primitive, and therefore indicates the bulk transfer number of the last write data transfer for which a transfer end read confirm primitive had been received by the initiator. It is only present if recovery is to an open regime that had negotiated the use of concurrent access.

Replace Table 24.

20 Access to file contents

20.1 Bulk data transfer

Replace second paragraph:

When overlapped access is not in use, these procedures start and finish in a single data transfer idle state, and so can be considered as a self-contained procedural unit, which is primitive in the definition of the remainder of the file service.

If overlapped access is in use then these procedures may be overlapped. It is therefore not necessarily true that read and write procedures start and finish in a data transfer idle state.

Replace Table 30.

Add sentence to end of fourth paragraph:

If overlapped access is in use then the initiator and responder may each be acting as both sender and receiver, with respect to the read and write procedures, as previously described.

| Parameter | F-RECOVER | F-RECOVER | F-RECOVER | F-RECOVER |
|-------------------------------------|------------------------|------------------------------------|-------------|--------------------|
| | request | Indication | response | comm |
| State Result | | | Mandatory | Mandatory |
| Action Result | | | Mandatory | Mandatory |
| Activity Identifier | Mandatory | Mandatory (=) | | |
| Bulk Transfer number | Mandatory | Mandatory (=) | | |
| Requested Access | Mandatory | Mandatory (=) | | |
| Access Passwords | Optional | Optional (=) | | |
| Contents Type | | | Mandatory | Mandatory (=) |
| Recovery Point | Conditional | Conditional (=) | Conditional | Conditional (=) |
| Diagnostic | | | Optional | Optional |
| Concurrent Bulk Transfer Number | Conditional | Conditional (=) | Conditional | Conditional (=) |
| Concurrent Recovery Point | Conditional | Conditional (=) | Conditional | Conditional (=) |
| Last Transfer End Read Request | | | Conditional | Conditional (=) |
| Last Transfer End Read Response | Conditional | Conditional RE (=) | VIEW | |
| Last Transfer End Write Request | (stand | ards.iteh.ai | Conditional | Conditional (=) |
| Last Transfer End Write Response | Conditional ISO 857 | Conditional -3:1988/Ar(≅)2:1993 | | |

Table 24 - F-RECOVER parameters

https://standards.iteh.ai/catalog/standards/sist/00c87397-4b7d-4892-9ae4-

3a702ec18a3a/iso-8571-3-1988-amd-2-1993

Section three: Definition of bulk data transfer primitives

22 Sequence of bulk data transfer primitives

22.1 Normal sequences

The normal progress of individual read and write bulk data transfer procedures is illustrated by the state transition diagram shown in Figure 8. Full state transition diagrams are contained in Annex E.

The sequences of primitives allowed as a result of the use

of the bulk data transfer service with overlapped access, is specified in Annex F.

Note - The use of a common Data Transfer Idle state for read and write bulk data transfer procedures in figure 8, serves to illustrate the progress of use of the bulk transfer service itself when overlapped access is not in use.

Re-label figure 8.

Figure 8 - Simplified State Transition Diagram for Read and Write Bulk Data Transfer Procedures

22.2 Constraints on issue of primitives

| Primitive Name | Confirmed Service | Request by | Parameters | Failure Notification |
|-------------------|---------------------------|------------------------------------|---|-----------------------------|
| F-READ | iTe№STA | N Initiator | PREBulk data transfer specification | F-DATA-END action result |
| F-WRITE | No (Sta | Initiator 8571-3:1988/Ar | Bulk data transfer specification 11 2:1993 Transfer number | F-CANCEL |
| F-DATAh | tps://standNos_iteb.ai/ca | talog/senders/si | t/00c87397-4b7Datavalue-4- | F-CANCEL |
| F-DATA-END | Ng a702ec18 | a3a/is Sender 3-1 | 988-amd-2-199 Action result Diagnostic | F-CANCEL |
| F-TRANSFER-END | Yes | Initiator | Action result Shared ASE information Diagnostic Request type Transfer number [Last transfer end read response] [Last transfer end write response] | action result |
| F-CANCEL | Yes | Either | Action result Shared ASE information Diagnostic Request type Transfer number [Last transfer end read request] [Last transfer end write request] [Last transfer end write response] | none |
| [F-CHECK] | Yes | Either | [Checkpoint identifier Transfer number] | F-CANCEL |
| [F-RESTART] | Yes | Either | [Checkpoint Identifier Request type Transfer number Last transfer end read request Last transfer end read response Last transfer end write request Last transferend write response] | F-CANCEL |

Table 30 - Bulk data transfer service primitives

The primitives may be issued in any sequence consistent with the LOTOS specification in Annex F. In addition, for the non-overlapped case, the constraints are presented in tables 31 and 32. The sequences given with the individual primitive definitions apply.

22.3 Key to tables 31 and 32

This amendment makes no changes to sub-clause 22.3.

23 Common bulk data transfer parameters

23.2 Checkpoint identifier

Replace last sentence of first paragraph:

If the use of concurrent access has been negotiated for the open regime then there may be more than one checkpoint number sequence valid at any one time. The bulk transfer number is used to identify the correct sequence. For the F-RESTART primitive, the value is between 0 and 999998 for each direction of transfer.

23.43 Request type

The request type parameter identifies the type of transfer to single the procedure then the responder may issue F-DATA requests which a primitive is related i.e. read of write sitch alcatalog/standar after /issuing/the F-DATA-END request for the preceding 3a702ec18a3a/iso-8571 read procedure;993

24 Bulk data transfer

Replace the first paragraph:

This group of services performs the transfer of bulk data. Each procedure begins with the service initiator issuing either an F-READ request or an F-WRITE request as appropriate. This leads to the issue of a sequence of F-DATA requests followed by an F-DATA-END request by the sender of the data. A procedure is completed by the initiator issuing an F-TRANSFER-END request, as appropriate. If overlapped access has been negotiated, then the service primitives for read and write procedures may be interspersed, subject to the constraints defined for consecutive access and concurrent access. The primitives are defined in 24.1 to 24.6, and the valid sequences, with and without overlapped access, are defined in 24.7 and 24.8.

24.1 Read bulk data service

Replace sub-clause 24.1.1:

24.1.1 Function

The F-READ service specifies a data transfer from the service responder (that is the sender) to the service initiator (that is the receiver).

24.1.1.1 Service without overlapped access

Only one F-READ procedure may be in progress at any time on a single application association. The direction of data flow established continues until the exchange of F-TRANSFER-END primitives. Rejection of an F-READ indication is by issue of an F-DATA-END with an action result indicating unsuccessful.

These primitives signal a transfer of control from the initiator to the sender. They mark a reversal of the service asymmetry for the duration or the data transfer.

24.1.1.2 Service with consecutive access

tiated for the e checkpoint bulk transfer e. For the Fd 999998 for TANDA then the F-DATA requests are issued subject to the then the F-DATA requests are issued subject to the standard constraints:

a) if the preceding bulk transfer procedure was a read

b) If the preceding bulk transfer procedure was a write procedure then the responder may issue F-DATA requests after issuing the F-TRANSFER-END response for that write procedure.

A F-READ procedure concludes with the exchange of F-TRANSFER-END primitives.

Rejection of an F-READ indication is by issue of an F-DATA-END request with an action result indicating unsuccessful, although this may not be issued until the responder has issued a F-DATA-END indication or F-TRANSFER-END (write) response, as appropriate, for the preceding bulk transfer procedure.

24.1.1.3 Service with concurrent access

F-READ procedures may be requested at any time during the open regime, up to the limit negotiated as the transfer window when the open regime was established. The responder maintains two independent queues for the read and write bulk transfer requests. If there are no outstanding requests for read procedures then the responder may issue F-DATA requests immediately, otherwise the F-DATA requests are issued after the issue of an F-DATA-END request for the preceding read procedure. A F-READ procedure concludes with the exchange of F-TRANSFER-END primitives.

Rejection of an F-READ indication is by issue of an F-DATA-END request with an action result indicating unsuccessful, although this may not be issued until the responder has issued a F-DATA-END indication for the preceding read procedure.

24.1.2 Types of primitives and parameters

Replace table 33.

Add sub-clause:

24.1.2.2 Transfer Number

The transfer number parameter is defined in 13.15.

24.2 Write bulk data service

Replace sub-clause 24.2.1:

24.2.1 Function

The F-WRITE service specifies a data transfer from the S.I service initiator (that is the sender) to the service responder (that is the receiver). ISO 8571-3:1988/Am

iTeh STANDARD

https://standards.iteh.ai/catalog/standards/sist 24.2.1.1 Service without overlapped access 8571-3-19

Only one F-WRITE procedure may be in progress at any time on a single application association. The direction of data flow established continues until the exchange of F-TRANSFER-END primitives. An F-WRITE indication can be rejected by issuing an F-CANCEL request (see 24.6). If the transfer is rejected, no further F-DATA indication primitives are received by the responder.

24.2.1.2 Service with consecutive overlapped access

F-WRITE procedures may be requested at any time during the open regime, up to the limit negotiated as the transfer window when the open regime was established. The requests for bulk transfer procedures are queued. If there are no outstanding bulk transfer requests then the F-DATA requests may be issued immediately, otherwise the F-DATA requests are issued subject to the following constraints:

Table 33 - F-READ parameters

| Parameter | F-READ request | F-READ indication |
|----------------------------------|-------------------|--------------------|
| Bulk data transfer specification | Mandator | Mandatory (=) |
| Transfer number | Conditional | Conditional (=) |

a) if the preceding bulk transfer procedure was a read procedure then the initiator may issue F-DATA requests after issuing a F-TRANSFER-END request for the preceding read procedure;

b) if the preceding bulk transfer procedure was a write procedure then the initiator may issue F-DATA requests after issuing a F-TRANSFER-END request for the preceding write procedure.

An F-WRITE procedure concludes with the exchange of F-TRANSFER-END primitives.

A responder may reject an F-WRITE indication by issuing an F-CANCEL request, although this may not be issued until the responder has received an F-TRANSFER-END (read) indication or issued an F-TRANSFER-END (write) response, as appropriate, for the preceding bulk transfer procedure (see 24.7).

24.2.1.3 Service with concurrent access

F-WRITE procedures may be requested at any time during the open regime, up to the limit negotiated as the transfer window when the open regime was established. Two independent queues for the read and write bulk transfer requests. If there are no outstanding requests for write procedures then the initiator may issue F-DATA requests immediately, otherwise the F-DATA requests are issued after the issue of an F-TRANSFER-END request for the preceding write⁹ procedure. A F-WRITE procedure concludes with the exchange of F-TRANSFER-END primitives.

Rejection of an F-WRITE indication is by issue of an F-CANCEL request, although this may not be issued until the responder has issued a F-TRANSFER-END response, for the preceding write procedure (see 24.6).

24.2.2 Types of primitives and parameters

Replace table 34.

Add sub-clause:

24.2.2.2 Transfer Number

The transfer number parameter is defined in 13.15.

24.4 End of data transfer service

24.4.1 Function

Replace last sentence:

The sender may issue an F-DATA-END request with an unsuccessful action as a rejection of an F-READ indication, subject to the constraints defined in 24.1.