INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

T.433

(09/92)

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

DOCUMENT TRANSFER, ACCESS AND MANIPULATION (DTAM) – SERVICES AND PROTOCOLS – PROTOCOL SPECIFICATION

Recommendation T.433

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation T.433 was revised by Study Group VIII and was approved under the Resolution No. 2 procedure on the18th September 1992.

CCITT NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized private operating agency.

© ITU 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 ITU.

CONTENT

Page

Introduction ii

1 Scope and field of application 1

2 References 1

3 Definitions and abbreviations 1

4 Conventions 1

5 Overview of the protocol 2
- 5.1 Service provision 2
- 5.2 Relationship with other ASEs and lower layer services 2
- 5.3 Model of Telematic Protocol Architecture (TPA) 3

6 Elements of procedure 5
- 6.1 Summary list of DTAM protocol data units 6
- 6.2 DTAM association establishment 6
- 6.3 Normal termination of a DTAM association 13
- 6.4 Abnormal termination of a DTAM association 17
- 6.5 Capability 20
- 6.6 Document bulk transfer 26
- 6.7 Document unconfirmed manipulation 32
- 6.8 Document confirmed manipulation 35
- 6.9 Typed Data Transfer 36
- 6.10 Remote document access 36
- 6.11 Remote document management 36
- 6.12 Token control 37
- 6.13 Exception report 39
- 6.14 Rules of extensibility 41

7 Mapping to the lower services 41
- 7.1 Mapping to the presentation and ACSE services 41
- 7.2 Mapping to the presentation and RTSE services 45
- 7.3 Mapping to the Session service (Recommendation X.215) in Transparent Mode 49

8 Abstract syntax definition of APDUs 57
- 8.1 Abstract syntax definition of APDUs in Normal Mode 57
- 8.2 Abstract syntax definition of APDUs for use of the session service in Transparent Mode 60

9 Conformance 61

Annex A – Examples of protocol sequence for document bulk transfer 62

Annex B – DTAM-PM state tables 65
- B.1 General 65
- B.2 Conventions 74
- B.3 Actions to be taken by the DTAM-PM 74
- B.4 Definition of variables 75

Annex C – DTAM-PM state tables 76
- C.1 General 76

INTRODUCTION

This Recommendation specifies the protocol for the services provided by an application-service-element, the Document Transfer and Manipulation Service Element (DTAM) to support applications in a distributed Telematic systems environment. This Recommendation is one of a set of Recommendations specifying the protocols for sets of application-service-elements specifically used by a number of applications.

Recommendation T.433

DOCUMENT TRANSFER, ACCESS AND MANIPULATION (DTAM) – SERVICES AND PROTOCOLS – PROTOCOL SPECIFICATION

(revised 1992)

1 Scope and field of application

This Recommendation specifies the protocol and procedures for the Document Transfer and Manipulation Service Element. The DTAM services are provided in conjunction with the Association Control Service Element (ACSE) service (see Recommendation X.217), Reliable Transfer Service Element (RTSE) service (see Recommendation X.218) and the Presentation-service (see Recommendation X.216) or the Session service (see Recommendation X.215). Depending on the mapping, Recommendation T.62 *bis* may also apply.

The DTAM procedures are defined in terms of:

a) the interactions between peer DTAM protocol machines through the use of the ACSE-service, RTSE-service and Presentation-service or Session service; and

b) the interactions between the DTAM protocol machine and its service-user.

This Recommendation specifies conformance requirements for systems implementing these procedures.

The use of Remote Operation Service Element (ROSE: see Recommendation X.219) is for further study.

2 References

References are listed in Recommendation T.431.

3 Definitions and abbreviations

Terms and abbreviations are defined in Recommendation T.431. The definitions of service primitive names given in Recommendation T.432 are used in this Recommendation.

4 Conventions

This Recommendation specifies the APDU fields. In section 6, tables are presented for each DTAM APDU. Each field is summarized by the following notation:

M Presence is mandatory

U Presence is optional

req Source is related request primitive

ind Sink is related indication primitive

rsp Source is related response primitive

cnf Sink is related confirm primitive

sp Source or sink is the DTAM-PM

The structure of each DTAM APDU is specified in section 8 using the abstract syntax notation of Recommendation X.208.

5 Overview of the protocol

5.1 *Service provision*

The protocol specified in this Recommendation provides the DTAM services defined in Recommendation T.432. These services are listed in Table 1/T.433.

TABLE 1/T.433

DTAM services summary

Service	Type
D-INITIATE	Confirmed
D-TERMINATE	Confirmed
D-P-ABORT	Provider-initiated
D-U-ABORT	Unconfirmed
C-CAPABILITY	Confirmed
D-TRANSFER	Provider-confirmed
D-TYPED-DATA	Unconfirmed
D-CREATE	Unconfirmed
D-DELETE	Unconfirmed
D-MODIFY	Unconfirmed
D-CALL	Unconfirmed
D-REBUILD	Unconfirmed
D-TOKEN-GIVE	Unconfirmed
D-CONTROL-GIVE	Unconfirmed
D-TOKEN-PLEASE	Unconfirmed
D-P-EXCEPTION-REPORT	Provider-initiated
D-U-EXCEPTION-REPORT	Unconfirmed

Note – D-REBUILD, D-P-EXCEPTION-REPORT and D-U-EXCEPTION-REPORT services are for further study.

5.2 *Relationship with other ASEs and lower layer services*

5.2.1 *ACSE service (when RTSE is not used)*

The DTAM services require access to the A-ASSOCIATE, A-RELEASE, A-ABORT and A-P-ABORT services. The inclusion of the DTAM in an application-context precludes the use of any of the above ACSE services by any other ASE or the user-element.

In the Transparent mode of DTAM, DTAM primitives are mapped directly onto the session service primitives, consequently, ACSE is not used.

5.2.2 *RTSE service*

The RTSE is used to support document bulk transfer in the normal mode for document bulk transfer.

The RTSE provides for the reliable transfer of application-protocol-data units (APDUs). The RTSE ensures that each APDU is completely transferred exactly once, or that the sender is warned of an exception. The RTSE recovers from communication and end-system failure and minimizes the amount of retransmission needed for recovery.

5.2.3 *ROSE service*

The use of this ASE is for further study.

5.2.4 *Presentation-service*

DTAM services may require access to the P-CAPABILITY-DATA, P-DATA, P-U-EXCEPTION-REPORT, P-P-EXCEPTION-REPORT, P-TOKEN-PLEASE and P-TOKEN-GIVE services. This Recommendation recognizes that the ACSE services require access to the P-CONNECT, P-RELEASE, P-U-ABORT and P-P-ABORT services.

5.2.5 *X.215 Session service*

In the Transparent Mode, APDUs defined in DTAM are directly mapped to the Session service defined in Recommendation X.215. The procedures described in Recommendation T.62 *bis* also apply.

DTAM services may require access to the S-CONNECT, S-CAPABILITY-DATA, S-ACTIVITY-START, S-DATA, S-MINOR-SYNCHRONIZE, S-ACTIVITY-END, S-ACTIVITY-INTERRUPT, S-ACTIVITY-DISCARD, S-U-EXCEPTION-REPORT, S-ACTIVITY-RESUME, S-P-EXCEPTION-REPORT, S-TOKEN-PLEASE, S-CONTROL-GIVE, S-RELEASE, S-U-ABORT and S-P-ABORT services.

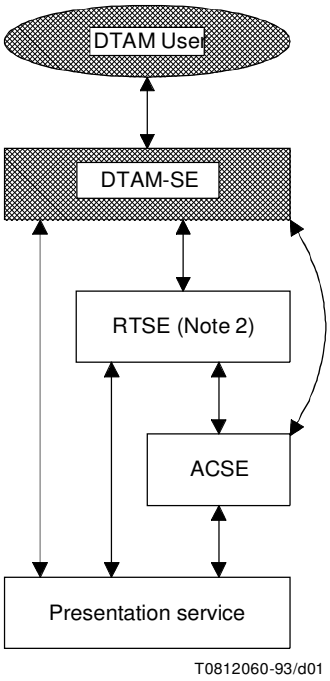
5.3 *Model of Telematic Protocol Architecture (TPA)*

The DTAM operates between two DTAM Protocol Machines (DTAM-PMs) in the Application layer of the OSI model. Protocol elements are exchanged between DTAM-PMs, using the Session service as defined in Recommendation X.215 or the services of RTSE, ACSE and of the Presentation Layer as defined in Recommendations X.218, X.217 and X.216 respectively. The model for Telematic Protocol Architecture (TPA) is illustrated in Figure 1/T.433. This application layer protocol architecture is composed of the ACSE (Association Control Service Element), RTSE (Reliable Transfer Service Element), DTAM-SE (Service Element) and DTAM users. The use of RTSE is only for document bulk transfer in the Normal Mode. Use of the Remote Operation Service Element (ROSE) is for further study.

5.3.1 *Functions of DTAM user*

DTAM users have the role of accurately reflecting the actual Telematic user (i.e. terminal user or system user) intentions in communication, and have functions to perform the applications (document bulk transfer, document manipulation, document transfer and manipulation, etc.) on behalf of the actual user. This mechanism is provided by the use of the DTAM-SE through the DTAM service defined in Recommendation T.432. The DTAM service is the logical interface between the DTAM user and DTAM service-provider for data handling, and is independent of specific hardware and software technique.

The DTAM user as an Application Service Element may be capable of interpreting the meaning of the content of an exchange document. For example, the retrieval command carried during information retrieval is not interpreted by the DTAM, but by the DTAM user.



Note 1 – In the case of use of the Session service (Transparent Mode), the appropriate DTAM APDUs are directly mapped to the Session service primitives.

Note 2 – The use of RTSE is only for the Document Bulk Transfer in the Normal mode.

FIGURE 1/T.433
Telematic Protocol Architecture (TPA) model in Application Layer

5.3.2 *Functions of DTAM service-provider*

To support Telematic Applications in single-source management of documents, DTAM service-provider provides the following communication functions.

1) *Association use control (kernel)*

DTAM provides the trigger for the use of the association given in ACSE, and controls association use during communication (termination, abort, etc.) either directly or by means of the RTSE. Applying the Session service to the lower layer functions of DTAM, this association use control will be mapped directly onto the session kernel functional unit.

2) *DTAM capability*

The DTAM capability is defined by sets of parameters in order to specify the communication features:

ODA application capabilities

a) document application profile;

b) operational application profile;

c) non-basic document characteristics; and

d) non-basic structural characteristics, etc.

Filetransfer capabilities

a) BFT capabilities

b) transparent data capability

3) *Data transmission function*

DTAM provides functions for document bulk transfer, document manipulations and typed data transmission as follows:

a) Document bulk transfer

DTAM provides a function to transmit the document in bulk under the communications environment negotiated by D-INITIATE service and additionally by D-CAPABILITY service.

b) Document manipulations

DTAM provides a function partially modifying a document seen by both users, by generating, revising or deleting structures (pages, blocks, etc.) of an existing document or to create a new document by generating structure of ODA and operational structure.

c) Typed data transmission

DTAM optionally provides a typed data transmission function which is independent of the data token control.

4) Document remote access

For further study.

5) Document remote management

For further study.

6) Token control

DTAM optionally provides the function of token control to handle the data token for dialogue.

7) Exception report

DTAM optionally provides the exception reporting function for error control during the DTAM communication.

6 Elements of procedure

This section identifies all the types of protocol data units which constitute the elements of the DTAM protocol between two DTAM-protocol-machines (DTAM-PMs). A protocol data unit (PDU) is the smallest quantity of information exchanged between DTAM-PMs which has a self-contained semantic significance.

When a DTAM service primitive is received from the DTAM user, DTAM transmits the DTAM primitive data to the opposite DTAM through the DTAM protocol, then the opposite DTAM generates the DTAM service primitives and notifies its DTAM user. The DTAM protocol data units (D-PDU) are shown in Table 2/T.433.

Individual parameters of DTAM service primitives are, in principle, all mapped to individual PDU parameters, but there are PDU including parameters, other than those specified in service primitives, such as those generated by DTAM itself. For example, D-INITIATE-REQ PDU also includes the DTAM protocol version parameter, which is used to negotiate the version of protocol between the DTAM-PMs. Note that the DTAM user is not concerned with this DTAM negotiation.

The PDUs are here identified symbolically with minimal reference to their mapping on to the lower layer service functions which implement them, thus no differentiation is made, in this section, between PDUs which are effected as specific Presentation service primitives and PDUs which are transferred as DTAM PDUs using the Presentation service data transfer functions. Details of PDU mapping and encoding are given in Section 8.

PDUs are given both full names, which should be used outside the context of this Recommendation, and abbreviated names which are used within this Recommendation for brevity. The full names consist of one or two words descriptive for the purpose of the PDU, prefixed by D- and, in the case of request/response pairs of PDUs, suffixed by -REQ or -RESP as appropriate. The abbreviated names are three letters each, with Q or R appended in the case of request/response pairs.

6.1 *Summary list of DTAM protocol data units*

TABLE 2/T.433

DTAM protocol data units

Functional units	PDU abbreviation	Protocol elements (PDU)	Cross reference
Association use control (kernel)	DINQ DINR DTEQ DTER DAB	D-INITIATE-REQ D-INITIATE-RESP D-TERMINATE-REQ D-TERMINATE-RESP D-ABORT	6.2 6.2 6.3 6.3 6.4
Capability	DCPQ DCPR	D-CAPABILITY-REQ D-CAPABILITY-RESP	6.5 6.5
Document bulk transfer	None	None	6.6
Document unconfirmed manipulation	DCR DDL DMD DCL DRD	D-CREATE D-DELETE D-MODIFY D-CALL D-REBUILD (Further study)	6.7 6.7 6.7 6.7 6.7
Document confirmed manipulation		(Further study)	6.8
Typed data transmission	DTD	D-TYPED-DATA	6.9
Remote document access		(Further study)	6.10
Remote document management		(Further study)	6.11
Token control	DTP	D-TOKEN-PLEASE	6.12
Exception report		(Further study)	6.13

6.2 *DTAM association establishment*

6.2.1 *Purpose*

The DTAM association establishment procedure is used to establish an association of DTAM between two AEs. It supports the D-INITIATE service.

6.2.2 *APDUs used*

The DTAM association establishment procedure uses the D-INITIATE-REQ (DINQ) and the D-INITIATE-RESP (DINR) APDUs.

6.2.2.1 *DINQ APDU*

The fields of the DINQ APDU are listed in Tables 3/T.433 and 4/T.433 for Normal Mode and Transparent Mode respectively.

TABLE 3/T.433

DINQ APDU fields for Normal Mode

Field name	Presence	Source	Sink
Service classes	(Note 1)	req	ind
Telematic requirements	M	req	ind
Application capabilities	M	req	ind
Protocol version	U (Note 2)	sp	sp
DTAM QOS	(Note 1)	req	ind
Account	(Note 1)	req	ind
User information	U	req	ind

Note 1 – The use of this parameter is for further study.

Note 2 – This parameter is defaultable.

TABLE 4/T.433

DINQ APDU fields for Transparent Mode

Field name	Presence	Source	Sink
Application capabilities	M	req	ind

6.2.2.2 *DINR APDU*

The fields of the DINR APDU are listed in Tables 5/T.433 and 6/T.433 for Normal Mode and Transparent Mode respectively.

TABLE 5/T.433

DINR APDU fields for Normal Mode

Field name	Presence	Source	Sink
Telematic requirements	U	rsp	cnf
Application capabilities	U	rsp	cnf
Protocol version	U (Note 2)	sp	sp
DTAM QOS	(Note 1)	rsp	cnf
Result	M	rsp	cnf
User information	U	rsp	cnf

Note 1 – The use of this parameter is for further study.

Note 2 – This parameter is defaultable.

TABLE 6/T.433

DINR APDU fields for Transparent Mode

Field name	Presence	Source	Sink
Application capabilities	M	req	ind

6.2.3 *DTAM association establishment procedure*

6.2.3.1 *DTAM association establishment procedure mapped onto ACSE service (Normal Mode.)*

This procedure is driven by the following events:

a) a D-INITIATE request primitive from the requestor;

b) a DINQ APDU as User Data on an A-ASSOCIATE indication primitive;

c) a D-INITIATE response primitive from the responder; and

d) an A-ASSOCIATE confirm primitive (that may contain a DINR APDU).

6.2.3.1.1 *D-INITIATE request primitive*

6.2.3.1.1.1 The requesting DTAM-PM forms a DINQ APDU from parameter values of the D-INITIATE request primitive and its stored data in DTAM-PM (the Protocol Version field, etc.). It issues an A-ASSOCIATE request primitive also using information from the D-INITIATE request primitive. The User Data parameter of the A-ASSOCIATE request primitive contains the DINQ APDU.

6.2.3.1.1.2 The requesting DTAM-PM waits for a primitive from the ACSE service-provider, and does not accept any primitive from the requestor other than a D-U-ABORT request primitive.

6.2.3.1.2 *DINQ APDU*

6.2.3.1.2.1 The responding DTAM-PM receives a DINQ APDU from its peer as User Data on an A-ASSOCIATE indication primitive. If any of the parameters of the A-ASSOCIATE indication primitive or the fields in the DINQ APDU are unacceptable to this DTAM-PM, it forms a DINR APDU with the appropriate rejecting Result field, and sends the DINR APDU as User Data on an A-ASSOCIATE response primitive. The Result parameter on the A-ASSOCIATE response primitive specifies “rejected (permanent)”. The DTAM-PM does not issue a D-INITIATE indication primitive to the responder, and the association is not established.

6.2.3.1.2.2 If the A-ASSOCIATE indication primitive and its DINQ APDU are acceptable to the responding DTAM-PM, it issues a D-INITIATE indication primitive to the responder. The D-INITIATE indication primitive parameters are derived from the DINQ APDU and from the A-ASSOCIATE indication primitive. The DTAM-PM waits for a D-INITIATE response primitive from the responder and does not accept any other primitives from the responder except a D-U-ABORT request primitive.

6.2.3.1.3 *D-INITIATE response primitive*

6.2.3.1.3.1 When the DTAM-PM receives the D-INITIATE response primitive, the Result parameter specifies whether the responder has accepted or rejected the DTAM association. The DTAM-PM forms a DINR APDU using the D-INITIATE response primitive parameters. The DINR APDU is sent as the User Data parameter on the A-ASSOCIATE response primitive.

6.2.3.1.3.2 If the responder accepted the DTAM association request, the Result parameter on the related A-ASSOCIATE response primitive specifies “accepted”, and the Result field of the outgoing DINR APDU also specifies “accepted”. The DTAM association is established.

6.2.3.1.3.3 If the responder rejected the DTAM association request, the Result parameter on the related A-ASSOCIATE response primitive specifies “Result: rejected (permanent or transient)”, and the Result field of the outgoing DINR APDU contains the appropriate rejection value. The DTAM association is not established.

6.2.3.1.4 *A-ASSOCIATE confirm primitive*

6.2.3.1.4.1 The requesting DTAM-PM receives an A-ASSOCIATE confirm primitive. The following situations are possible:

a) the DTAM association has been accepted;

b) the responding DTAM-PM or the responder has rejected the DTAM association; or

c) the association service-provider has rejected the related association.

6.2.3.1.4.2 If the DTAM association was accepted, the A-ASSOCIATE confirm primitive Result parameter specifies "accepted". The User Data parameter contains a DINR APDU, and the Result field of the DINR APDU also specifies "accepted". The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the A-ASSOCIATE confirm primitive and from the DINR APDU. The D-INITIATE confirm primitive Result parameter specifies "accepted", and the DTAM association is established.

6.2.3.1.4.3 If the DTAM association was rejected by either the responding DTAM-PM or by the responder, the A-ASSOCIATE confirm primitive Result parameter specifies "Result: rejected (permanent or transient)" and "Result source: ACSE service-user". The User Data parameter contains a DINR APDU, and the Result field of the DINR APDU indicates the reason for rejection. The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the A-ASSOCIATE confirm primitive and from the DINR APDU.

The D-INITIATE confirm primitive Result parameter contains the appropriate rejection value. The DTAM association is not established.

6.2.3.1.4.4 If the association was rejected by the association service-provider, the A-ASSOCIATE confirm primitive Result parameter specifies "Result: rejected (permanent or transient)" and "Result source: ACSE service-provider". In this situation, the User Data parameter does not contain any APDU. The requesting DTAM-PM issues a D-INITIATE confirm primitive with the appropriate Result parameter. The DTAM association is not established.

6.2.3.2 *DTAM association establishment procedure mapped onto RTSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-INITIATE request primitive from the requestor;

b) a DINQ APDU as User Data on an RT-OPEN indication primitive;

c) a D-INITIATE response primitive from the responder; and

d) an RT-OPEN confirm primitive (that may contain a DINR APDU).

6.2.3.2.1 *D-INITIATE request primitive*

6.2.3.2.1.1 The requesting DTAM-PM forms a DINQ APDU from parameter values of the D-INITIATE request primitive and its stored data in DTAM-PM (the Protocol Version field, etc.). It issues an RT-OPEN request primitive also using information from the D-INITIATE request primitive. The User Data parameter of the RT-OPEN request primitive contains the DINQ APDU.

6.2.3.2.1.2 The requesting DTAM-PM waits for a primitive from the RTSE service-provider, and does not accept any primitive from the requestor other than a D-U-ABORT request primitive.

6.2.3.2.2 *DINQ APDU*

6.2.3.2.2.1 The responding DTAM-PM receives a DINQ APDU from its peer as User Data on an RT-OPEN indication primitive. If any of the parameters of the RT-OPEN indication primitive or the fields in the DINQ APDU are unacceptable to this DTAM-PM, it forms a DINR APDU with the appropriate rejecting Result field, and sends the DINR APDU as User Data on an RT-OPEN response primitive. The Result parameter on the RT-OPEN response primitive specifies "rejected (permanent)". The DTAM-PM does not issue a D-INITIATE indication primitive to the responder, and the association is not established.

6.2.3.2.2.2 If the RT-OPEN indication primitive and its DINQ APDU are acceptable to the responding DTAM-PM, it issues a D-INITIATE indication primitive to the responder. The D-INITIATE indication primitive parameters are derived from the DINQ APDU and from the RT-OPEN indication primitive. The DTAM-PM waits for a D-INITIATE response primitive from the responder and does not accept any other primitives from the responder except a D-U-ABORT request primitive.

6.2.3.2.3 *D-INITIATE response primitive*

6.2.3.2.3.1 When the DTAM-PM receives the D-INITIATE response primitive, the Result parameter specifies whether the responder has accepted or rejected the DTAM association. The DTAM-PM forms a DINR APDU using the D-INITIATE response primitive parameters. The DINR APDU is sent as the User Data parameter on the RT-OPEN response primitive.

6.2.3.2.3.2 If the responder accepted the DTAM association request, the Result parameter on the related RT-OPEN response primitive specifies "accepted", and the Result field of the outgoing DINR APDU also specifies "accepted". The DTAM association is established.

6.2.3.2.3.3 If the responder rejected the DTAM association request, the Result parameter on the related RT-OPEN response primitive specifies "Result: rejected (permanent or transient)", and the Result field of the outgoing DINR APDU contains the appropriate rejection value. The DTAM association is not established.

6.2.3.2.4 *RT-OPEN confirm primitive*

6.2.3.2.4.1 The requesting DTAM-PM receives an RT-OPEN confirm primitive. The following situations are possible:

a) the DTAM association has been accepted;

b) the responding DTAM-PM or the responder has rejected the DTAM association; or

c) the responding RTSE-PM has rejected the DTAM association.

6.2.3.2.4.2 If the DTAM association was accepted, the RT-OPEN confirm primitive Result parameter specifies "accepted". The User Data parameter contains a DINR APDU, and the Result field of the DINR APDU also specifies "accepted". The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the RT-OPEN confirm primitive and from the DINR APDU. The D-INITIATE confirm primitive Result parameter specifies "accepted", and the DTAM association is established.

6.2.3.2.4.3 If the DTAM association was rejected by either the responding DTAM-PM or by the responder, the RT-OPEN confirm primitive Result parameter specifies "Result: rejected (permanent or transient)" and "Result source: ACSE service-user". The User Data parameter contains a DINR APDU, and the Result field of the DINR APDU indicates the reason for rejection. The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the RT-OPEN confirm primitive and from the DINR APDU. The D-INITIATE confirm primitive Result parameter contains the appropriate rejection value. The DTAM association is not established.

6.2.3.2.4.4 If the association was rejected by the association service-provider, the RT-OPEN confirm primitive Result parameter specifies "Result: rejected (permanent or transient)" and "Result source: ACSE service-provider". In this situation, the User Data field is not used by the requesting DTAM-PM. The requesting DTAM-PM issues a D-INITIATE confirm primitive with the appropriate Result parameter. The DTAM association is not established.

6.2.3.2.4.5 If the association was rejected by the responding RTSE-PM, the RT-OPEN confirm primitive Result parameter specifies "Result: rejected (permanent or transient)" and "Result source: ACSE service-user". In this situation, the User Data parameter does not contain any APDU. The requesting DTAM-PM issues a D-INITIATE confirm primitive with the appropriate Result parameter. The DTAM association is not established.

6.2.3.3 *DTAM association establishment procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-INITIATE request primitive from the requestor;

b) a DINQ APDU as User Data on an S-CONNECT indication primitive;

c) a D-INITIATE response primitive from the responder; and

d) an S-CONNECT confirm primitive (that may not contain a DINR APDU).

6.2.3.3.1 *D-INITIATE request primitive*

6.2.3.3.1.1 The requesting DTAM-PM forms a DINQ APDU from parameter values of the D-INITIATE request primitive and its stored data in DTAM-PM (window size, etc.). It issues an S-CONNECT request primitive also using information from the D-INITIATE request primitive. The User Data parameter of the S-CONNECT request primitive contains the DINQ APDU.

6.2.3.3.1.2 The requesting DTAM-PM waits for a primitive from the Session service-provider and does not accept any primitive from the requestor other than a D-U-ABORT request primitive.

6.2.3.3.2 *DINQ APDU*

6.2.3.3.2.1 The responding DTAM-PM receives a DINQ APDU from its peer as User Data on an S-CONNECT indication primitive. If any of the parameters of the S-CONNECT indication primitive or the fields in the DINQ APDU are unacceptable to this DTAM-PM (e.g. no Session User Data in the S-CONNECT indication), it issues an S-CONNECT response primitive with a Result parameter specifying “ss-user-rejection”. The DTAM-PM does not issue a D-INITIATE indication primitive to the responder. The association is not established.

6.2.3.3.2.2 If the S-CONNECT indication primitive and its DINQ APDU are acceptable to the responding DTAM-PM, it issues a D-INITIATE indication primitive to the responder. The D-INITIATE indication primitive parameters are derived from the DINQ APDU. The DTAM-PM waits for a D-INITIATE response primitive from the responder and does not accept any other primitives from the responder except a D-U-ABORT request primitive.

6.2.3.3.3 *D-INITIATE response primitive*

6.2.3.3.3.1 When the DTAM-PM receives the D-INITIATE response primitive, the Result parameter specifies whether the responder has accepted or rejected the DTAM association. If the DTAM association is accepted, the DTAM-PM forms a DINR APDU using the D-INITIATE response primitive parameters. The DINR APDU is sent as the User Data parameter on the S-CONNECT response primitive.

6.2.3.3.3.2 If the responder accepted the DTAM association request, the Result parameter on the related S-CONNECT response primitive specifies “accept”. The DTAM association is established.

6.2.3.3.3.3 If the responder rejected the DTAM association request, the Result parameter on the related S-CONNECT response primitive specifies “ss-user-rejection” and DTAM-PM does not send DINR APDU.

6.2.3.3.4 *S-CONNECT confirm primitive*

6.2.3.3.4.1 The requesting DTAM-PM receives an S-CONNECT confirm primitive. The following situations are possible:

a) the DTAM association has been accepted;

b) the responding DTAM-PM or the responder has rejected the DTAM association; or

c) the Session service-provider has rejected the related association.

6.2.3.3.4.2 If the DTAM association was accepted, the S-CONNECT confirm primitive Result parameter specifies “accept”. The User Data parameter contains a DINR APDU. The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the S-CONNECT confirm primitive and from the DINR APDU. The D-INITIATE confirm primitive Result parameter specifies “accepted”. The DTAM association is established.

6.2.3.3.4.3 If the DTAM association was rejected by either the responding DTAM-PM or by the responder, the S-CONNECT confirm primitive Result parameter specifies “user-rejection” and there is no User Data (DINR APDU) in this confirm primitive. The requesting DTAM-PM issues a D-INITIATE confirm primitive to the requestor based on parameters from the S-CONNECT confirm primitive. The D-INITIATE confirm primitive Result parameter contains the value of “user-rejection”, and the DTAM association is not established.

6.2.3.3.4.4 If the association was rejected by the Session service-provider, the S-CONNECT confirm primitive Result parameter specifies “provider-rejection”. In this situation, the User Data field is not used by the requesting DTAM-PM. The requesting DTAM-PM issues a D-INITIATE confirm primitive with the appropriate Result parameter. The DTAM association is not established.

6.2.4 *Use of the DINQ/DINR APDU fields*

The DINQ APDU and DINR APDU fields are used as follows.

6.2.4.1 Service classes

The use of this parameter is for further study.

6.2.4.2 *Telematic Requirements*

This is the Telematic Requirements parameter value from the D-INITIATE request/response primitives. It appears as the Telematic Requirements parameter value of D-INITIATE indication/confirm primitives respectively. If the Telematic Requirements proposed by the requestor are not acceptable to the responder, the DTAM association fails to be established.

6.2.4.3 *Application Capabilities*

This is the Application Capabilities parameter value from the D-INITIATE request/response primitives. It appears as the Application Capabilities parameter value of the D-INITIATE indication/confirm primitives respectively. This parameter consists of sets of the sub-parameters. Each set specifies the capability to handle either ODA documents or files.

6.2.4.3.1 *ODA application capabilities*

6.2.4.3.1.1 *Document application profile*

The value of this parameter is either an Octet String or an ASN.1 object identifier. The Octet String designates the document application profile in line with Recommendation T.73 (Document application profile – T.73). The ASN.1 object identifier must conform to the rules specified in ISO 8824 and designate an application profile defined in accordance with the rules specified in Recommendation T.411 (Document application profiles).

6.2.4.3.1.2 *Document architecture class*

The value of this parameter is “formatted”.

This parameter is only used in the transparent mode.

6.2.4.3.1.3 *Non-basic document characteristics*

The value of this parameter is any combination of non-basic document characteristics defined in Recommendation T.414.

6.2.4.3.1.4 *Non-basic structural characteristics*

The value of this parameter is any combination of non-basic structural characteristics defined in Recommendation T.414.

6.2.4.3.1.5 *Operational application profile*

The detailed specification of the operational application profile is for further study.

6.2.4.3.2 *Filetransfer capabilities*

6.2.4.3.2.1 *BFT capabilities*

This parameter indicates the capability to receive files according to Recommendation T.434.

6.2.4.3.2.2 *Transparent data capability*

This parameter indicates the capability to receive data that are completely transparent.

6.2.4.4 *Protocol version*

The value assigned to this parameter is determined by the DTAM-PM. It is a variable length bit string where each bit that is set to one indicates the version of DTAM protocol that the DTAM-PM supports: bit 0 represents version 1, bit 1 represents version 2, etc.

The version to be used on the association is the highest number which both DTAM-PM support. If there is no common protocol version number for both DTAM-PMs, the association cannot be established. When this parameter is absent, the default value "version 1" is applied.

6.2.4.5 *DTAM QOS*

DTAM QOS is left for further study.

6.2.4.6 *Account*

The account parameter identifies the account to which costs incurred in the DTAM association which is being established are to be charged.

Note – The use of this parameter is for further study.

6.2.4.7 *Result*

If the DINQ APDU was rejected by the responding DTAM-PM (i.e. a D-INITIATE indication primitive was not issued to the responder), this field is supplied by the responding DTAM-PM, otherwise, this field is the Result parameter from the D-INITIATE response primitive. In either situation, it appears as the Result parameter on the D-INITIATE RESP (DINR) APDU. This field can take one of the following symbolic values:

– accepted;

– rejected by responder (reason-not-specified);

– rejected by responder (protocolVersion-not-supported);

– rejected by responder (DTAM-QOS-not-supported);

– rejected by responder (application-context-name-not-supported);

– rejected by responding DTAM-PM.

6.2.4.8 *User Information*

This is the User Information parameter from the D-INITIATE request and response primitive. It appears as the User Information parameter of the D-INITIATE indication and confirm primitive respectively, if issued.

6.2.5 *Collisions and interactions*

For further study.

6.3 *Normal termination of a DTAM association*

6.3.1 *Purpose*

This procedure is used for the normal termination of a DTAM association by an AE without loss of information in transit. It supports the D-TERMINATE service.

6.3.2 *APDUs used*

The normal termination procedure uses the D-TERMINATE-REQ (DTEQ) APDU and the D-TERMINATE-RESP (DTER) APDU.

6.3.2.1 *DTEQ APDU*

The fields of the DTEQ APDU are listed in Table 7/T.433.

TABLE 7/T.433

DTEQ APDU fields

Field name	Presence	Source	Sink
User information (Note)	U	req	ind

Note – This parameter is not applicable in Transparent Mode.

6.3.2.2 *DTER APDU*

The fields of the DTER APDU are listed in Table 8/T.433.

TABLE 8/T.433

DTER APDU fields

Field name	Presence	Source	Sink
Charging (Note)	U	rsp	cnf
User information (Note)	U	rsp	cnf

Note – This parameter is not applicable in Transparent Mode.

6.3.3 *Normal termination procedure*

6.3.3.1 *Normal termination procedure mapped onto ACSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TERMINATE request primitive from the requestor;

b) a DTEQ APDU as User Data on an A-RELEASE indication primitive;

c) a D-TERMINATE response primitive from the responder; and

d) a DTER APDU as User Data on an A-RELEASE confirm primitive.

6.3.3.1.1 *D-TERMINATE request primitive*

6.3.3.1.1.1 When a D-TERMINATE request primitive is received, the DTAM-PM sends a DTEQ APDU as User Data on an A-RELEASE request primitive using the parameters from the D-TERMINATE request primitive.

Note – The requestor is required to meet the association (presentation and session) requirements in order to issue a D-TERMINATE request primitive.

6.3.3.1.1.2 The requesting DTAM-PM now waits for a primitive from the association service-provider. It does not accept any primitives from the requestor other than a D-U-ABORT request primitive.

6.3.3.1.2 *DTEQ APDU*

6.3.3.1.2.1 When the responding DTAM-PM receives the DTEQ APDU as User Data on an A-RELEASE indication primitive, it issues a D-TERMINATE indication primitive to the responder.

6.3.3.1.3 *D-TERMINATE response primitive*

6.3.3.1.3.1 The responding DTAM-PM forms a DTER APDU from the response primitive parameters. The DTER APDU is sent as User Data on an A-RELEASE response primitive. The Result parameter of A-RELEASE response has the value “affirmative”.

Note – The responder is able to reject the termination request of DTAM association only in the case of selecting a negotiated release session functional unit. The use of this functional unit is for further study.

6.3.3.1.4 *DTER APDU*

6.3.3.1.4.1 The requesting DTAM-PM receives an A-RELEASE confirm primitive containing a DTER APDU from its peer. The Result parameter on the A-RELEASE confirm specifies that the responder agrees that the DTAM association may be terminated. The requesting DTAM-PM forms a D-TERMINATE confirm primitive from the DTER APDU.

6.3.3.2 *Normal termination procedure mapped onto RTSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TERMINATE request primitive from the requestor;

b) a DTEQ APDU as User Data on an RT-CLOSE indication primitive;

c) a D-TERMINATE response primitive from the responder; and

d) a DTER APDU as User Data on an RT-CLOSE confirm primitive.

6.3.3.2.1 *D-TERMINATE request primitive*

6.3.3.2.1.1 When a D-TERMINATE request primitive is received, the DTAM-PM sends a DTEQ APDU as User Data on an RT-CLOSE request primitive using the parameters from the D-TERMINATE request primitive.

Note – The requestor is required to be the initiator to meet the RTSE requirements.

6.3.3.2.1.2 The requesting DTAM-PM now waits for a primitive from the RTSE service-provider. It does not accept any primitives from the requestor other than a D-U-ABORT request primitive.

6.3.3.2.2 *DTEQ APDU*

6.3.3.2.2.1 When the responding DTAM-PM receives the DTEQ APDU as User Data on an RT-CLOSE indication primitive, it issues a D-TERMINATE indication primitive to the responder.

6.3.3.2.3 *D-TERMINATE response primitive*

6.3.3.2.3.1 The responding DTAM-PM forms a DTER APDU from the response primitive parameters. The DTER APDU is sent as User Data on an RT-CLOSE response primitive.

6.3.3.2.4 *DTER APDU*

6.3.3.2.4.1 The requesting DTAM-PM receives an RT-CLOSE confirm primitive containing a DTER APDU from its peer. The Result parameter on the RT-CLOSE confirm specifies that the responder agrees that the DTAM association may be terminated. The requesting DTAM-PM forms a D-TERMINATE confirm primitive from the DTER APDU.

6.3.3.3 *Normal termination procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-TERMINATE request primitive from the requestor;

b) an S-RELEASE indication primitive without sending DTEQ APDU;

c) a D-TERMINATE response primitive from the responder; and

d) an S-RELEASE confirm primitive without sending DTER APDU.

6.3.3.3.1 *D-TERMINATE request primitive*

6.3.3.3.1.1 When a D-TERMINATE request primitive is received, the DTAM-PM issues an S-RELEASE request primitive without any SS-User-Data.

Note – The requestor is required to meet the association (presentation and session) requirements in order to issue a D-TERMINATE request primitive.

6.3.3.3.1.2 The requesting DTAM-PM now waits for a primitive from the Session service-provider. It does not accept any primitives from the requestor other than a D-U-ABORT request primitive.

6.3.3.3.2 *Implicit DTEQ ADPU*

6.3.3.3.2.1 When the responding DTAM-PM receives an S-RELEASE indication primitive, it issues a D-TERMINATE indication primitive to the responder without any parameters.

6.3.3.3.3 *D-TERMINATE response primitive*

6.3.3.3.3.1 The responding DTAM-PM forms an S-RELEASE response from the D-TERMINATE response primitive parameters. The Result parameter of S-RELEASE response has the value “affirmative”.

6.3.3.3.4 *Implicit DTER APDU*

6.3.3.3.4.1 The requesting DTAM-PM receives an S-RELEASE confirm primitive containing no DTAM APDU from its peer. The Result parameter on the S-RELEASE confirm always specifies “affirmative”. The requesting DTAM-PM forms a D-TERMINATE confirm primitive from the S-RELEASE confirm primitive and issues it to the requestor with no parameters.

6.3.4 *Use of the DTEQ APDU fields*

The DTEQ APDU fields are used as specified below.

6.3.4.1 *User Information*

This is the User Information parameter on the D-TERMINATE request primitive. It appears as the User Information parameter of the D-TERMINATE indication primitive.

6.3.5 *Use of the DTER APDU fields*

The DTER APDU fields are used as specified below.

6.3.5.1 *Charging*

The charging parameter conveys information on the costs attributed to the account during the DTAM association which is being released. The value of this parameter is for further study. The charging parameter is present at the end of a DTAM association, only if the account parameter was present at the beginning of that DTAM association. It is not mandatory to return a charge if that charge is zero.

6.3.5.2 *User Information*

This is the User Information parameter from the D-TERMINATE response primitive. It appears as the User Information parameter on the D-TERMINATE confirm primitive.

6.3.6 *Collisions and interactions*

6.3.6.1 *D-TERMINATE service*

A D-TERMINATE service collision cannot occur if session tokens were selected for the association (e.g. when RTSE is used). Only the AE that owns all of the available session tokens can issue the D-TERMINATE request primitive.

In the case where no token is available, overlapping attempts by both AEs to terminate the DTAM association are governed by the rule of Session service: initiator of the association is the winner of the collisions A-RELEASE service or S-RELEASE Session service. The DTAM association is terminated.

6.3.6.2 *D-U-ABORT service, DAB APDU or A(or RT or S)-P-ABORT service*

If either DTAM-PM receives a D-U-ABORT request primitive, a DAB APDU [as User Data on an A(or RT or S)-U-ABORT indication primitive] or an A(or RT or S)-P-ABORT indication primitive, it discontinues the normal DTAM association termination procedure, and instead follows abnormal termination procedure.

6.4 *Abnormal termination of a DTAM association*

6.4.1 *Purpose*

6.4.1.1 The abnormal termination can be used at any time to force the abrupt termination of the DTAM association by a requestor in either DTAM user, by either DTAM-PM, by the RTSE service-provider, by the ACSE service-provider or by the Session service-provider. It supports the D-U-ABORT, D-P-ABORT, RT-U-ABORT, RT-P-ABORT, A-ABORT, A-P-ABORT, S-U-ABORT and S-P-ABORT services.

6.4.1.2 The abnormal termination provides the following three procedures:

a) *user-abort procedure*

This procedure is defined as the following series of events received and actions taken by the DTAM-PM:

i) receiving a D-U-ABORT request primitive;

ii) sending D-ABORT(DAB) APDU as User Data on an A-ABORT request primitive or RT-U-ABORT request primitive (Normal Mode); or

issuing S-U-ABORT request primitive without User Data (Transparent Mode);

iii) the DTAM association is terminated.

b) *association-provider-abort procedure*

This procedure is defined as the following series of events received and actions taken by the DTAM-PM:

i) receiving A-P-ABORT indication primitive or RT-P-ABORT indication primitive (Normal Mode); or

receiving an S-P-ABORT indication primitive (Transparent Mode);

ii) issuing a D-P-ABORT indication primitive;

iii) the DTAM association is terminated.

c) *DTAM provider-abort procedure*

This procedure is defined as the following series of events detected and actions taken by the DTAM-PM:

i) detecting severe error situations in DTAM-PM;

ii) sending D-ABORT(DAB) APDU as User Data on an A-ABORT or RT-U-ABORT request primitive and issuing a D-P-ABORT indication primitive (normal mode); or

issuing an S-U-ABORT request primitive and a D-P-ABORT indication primitive (transparent mode).

6.4.2 *APDUs used*

The abnormal termination uses the D-ABORT(DAB) APDU.

6.4.2.1 *DAB APDU*

The fields of the DAB APDU are listed in Table 9/T.433.

TABLE 9/T.433

DAB APDU fields

Field name	Presence	Source	Sink
Abort source (Note)	M	sp	sp
Abort reason (Note)	U	sp	sp
Reflect parameter (Note)	U	sp	sp
User information (Note)	U	req	ind

Note – These parameter are not applicable in Transparent Mode.

6.4.3 *Abnormal termination procedure*

6.4.3.1 *Abnormal termination procedure mapped onto ACSE service (Normal Mode)*

6.4.3.1.1 *D-U-ABORT request primitive*

6.4.3.1.1.1 When a DTAM-PM receives a D-U-ABORT request primitive, it sends a D-ABORT(DAB) APDU as User Data on an A-ABORT request primitive. The DAB APDU "Abort Source" field is specified as a "requestor". If the User Information parameter was included on the D-U-ABORT request primitive, it is included in the DAB APDU. The DTAM association is terminated.

6.4.3.1.2 *DAB APDU*

6.4.3.1.2.1 When a DTAM-PM receives an A-ABORT indication primitive, the User Data parameter contains the DAB APDU. The DTAM-PM issues a D-U-ABORT indication primitive when the Abort Source field of the DAB APDU has the value "requestor". If a User Information field was contained in the DAB APDU, it is included in the D-U-ABORT indication primitive. The DTAM association is terminated.

6.4.3.1.3 *A-P-ABORT indication primitive*

6.4.3.1.3.1 When a DTAM-PM receives an A-P-ABORT indication primitive, the DTAM-PM issues a D-P-ABORT indication primitive to the DTAM user. The DTAM association is terminated.

6.4.3.1.3.2 An association-provider-abort is indicated to both DTAM-PMs by an A-P-ABORT indication primitive and may occur at any time. After such an event, when the Reliable Transfer Mode 2 was selected, the association-initiating DTAM-PM starts the association-recovery procedure.

Note – The association-recovery procedure is for further study.

6.4.3.1.3.3 If the association-provider-abort procedure was performed during the transfer procedure the requesting DTAM-PM starts the transfer-resumption procedure after the association-recovery procedure is successfully completed. If the association-recovery procedure was not successfully completed the requesting DTAM-PM performs the transfer-error procedure and the provider-abort procedure.

6.4.3.1.4 *Error detections by a DTAM-PM*

6.4.3.1.4.1 When a DTAM-PM detects severe error situations, it performs the DTAM provider-abort procedure.

6.4.3.1.4.2 The DTAM provider-abort procedure is performed to send a DAB APDU as User Data on an A-ABORT request primitive and to issue a D-P-ABORT indication primitive. The DAB APDU "Abort Source" field takes the value "DTAM service- provider" and additional DAB APDU parameters are specified to form the peer DTAM-PM of the error situation.

6.4.3.1.4.3 The use of association-recovery procedure (see 6.6.8) is for further study.

6.4.3.2 *Abnormal termination procedure mapped onto RTSE service (Normal Mode)*

6.4.3.2.1 *D-U-ABORT request primitive*

6.4.3.2.1.1 When a DTAM-PM receives a D-U-ABORT request primitive, it issues an RT-U-ABORT request primitive without DAB APDU as the User Data. The DTAM association is terminated.

6.4.3.2.2 *Implicit DAB APDU*

6.4.3.2.2.1 When a DTAM-PM receives an RT-U-ABORT indication primitive, the DTAM-PM issues a D-U-ABORT indication primitive. The DTAM association is terminated.

6.4.3.2.3 *RT-P-ABORT indication primitive*

6.4.3.2.3.1 When a DTAM-PM receives an RT-P-ABORT indication primitive, the DTAM-PM issues a D-P-ABORT indication primitive to the DTAM user. The DTAM association is terminated.

6.4.3.2.3.2 A reliable-transfer-provider-abort is indicated to both DTAM-PMs by an RT-P-ABORT indication primitive and may occur at any time.

6.4.3.2.4 *Error detections by a DTAM-PM*

6.4.3.2.4.1 When a DTAM-PM detects severe error situations, it performs the DTAM provider-abort procedure.

6.4.3.2.4.2 The DTAM provider-abort procedure is performed to issue an RT-U-ABORT request primitive without DAB APDU as the User Data. The DTAM-PM also issues a D-P-ABORT indication primitive to its service user. The DTAM association is terminated.

6.4.3.3 *Abnormal termination procedure mapped onto Session service (Transparent Mode)*

6.4.3.3.1 *D-U-ABORT request primitive*

6.4.3.3.1.1 When a DTAM-PM receives a D-U-ABORT request primitive, it issues an S-U-ABORT request primitive without DAB APDU as the User Data. The use of S-U-ABORT service will be interpreted as "Local Terminal Error". The DTAM association is terminated.

6.4.3.3.2 *Implicit DAB APDU*

6.4.3.3.2.1 When a DTAM-PM receives an S-U-ABORT indication primitive, the DTAM-PM issues a D-U-ABORT indication primitive. The DTAM association is terminated.

6.4.3.3.3 *S-P-ABORT indication primitive*

6.4.3.3.3.1 When a DTAM-PM receives an S-P-ABORT indication primitive, the DTAM-PM issues a D-P-ABORT indication primitive to its DTAM user. The DTAM association is terminated.

6.4.3.3.4 *Protocol errors*

6.4.3.3.4.1 When a DTAM-PM detects an invalid condition such as an unexpected APDU, it issues an S-U-ABORT request primitive without DAB APDU as the User Data.

The DTAM-PM also issues a D-P-ABORT indication primitive to its service-user. The DTAM association is terminated.

6.4.4 *Use of the ABORT APDU fields*

The ABORT APDU fields are used as specified below.

6.4.4.1 *Abort source*

This is supplied by the requesting DTAM-PM. This field can take one of the following symbolic values:

– DTAM service-provider; or

– requestor.

6.4.4.2 *Abort reason*

This field may contain one of the following values:

– local-system-problem	
– invalid-parameter	the invalid parameters are specified in the Reflect parameter field
– unrecognized-activity	
– temporary-problem	no attempt at association-recovery should be made for a period of time determined by a local rule
– protocol-error	of the DTAM-PM
– permanent-error	this value is used solely by the DTAM provider-abort procedure in normal-mode
– transfer completed	the responding DTAM-PM could not discard an already completed transfer.

6.4.4.3 *Reflect-parameter*

The Reflect-parameter field is a bit string that identifies which parameters are regarded as invalid parameters in the primitive received from the used service by the aborting DTAM-PM before the association-abort. The order of the bits in the bit string is the same as the order of the parameters in the tables of service parameters in Recommendations X.217 and X.216 (i.e. bit 1 represents the first parameter, etc.).

6.4.4.4 *User Information*

This is the information parameter from the D-U-ABORT request primitive. It appears as the User Information parameter on the D-U-ABORT indication primitive.

6.4.5 *Collisions and interactions*

The abnormal termination procedure may be used whenever a DTAM association is established, is in process of being established, or is being normally terminated. This procedure disrupts any other currently active procedure. An A-P-ABORT or RT-P-ABORT indication primitive can disrupt the D-U-ABORT exchange with loss of the user information in D-U-ABORT service. Collisions of DAB APDUs are governed by the A-P-ABORT or RT-U-ABORT service.

6.5 *Capability*

6.5.1 *Purpose*

It supports the D-CAPABILITY service.

6.5.2 *APDUs used*

The DTAM capability procedure uses the D-CAPABILITY-REQ (DCPQ) and the D-CAPABILITY-RESP (DCPR) APDUs.

6.5.2.1 *DCPQ APDU*

The fields of the DCPQ APDU are listed in Tables 10/T.433 and 11/T.433 for Normal Mode and Transparent Mode respectively.

TABLE 10/T.433

DCPQ APDU fields for Normal Mode

Field name	Presence	Source	Sink
Application capabilities			
ODA application capabilities			
Document application profile	U	req	ind
Non-basic structural characteristics	U	req	ind
Non-basic document characteristics	U	req	ind
Operational application profile	U	req	ind
Filetransfer capabilities			
BFT capabilities	U	req	ind
Transparent data capability	U	req	ind
User Information	U	req	ind

TABLE 11/T.433

DCPQ APDU fields for Transparent Mode

Field name	Presence	Source	Sink
Application capabilities			
Document application profile	U	req	ind
Document architecture class	U	req	ind
Non-basic structural characteristics	U	req	ind
Non-basic document characteristics	U	req	ind

6.5.2.2 *DCPR APDU*

The fields of the DCPR APDU are listed in Tables 12/T.433 and 13/T.433 for Normal Mode and Transparent Mode respectively.

TABLE 12/T.433

DCPR APDU fields for Normal Mode

Field name	Presence	Source	Sink
Application capabilities			
ODA application capabilities			
Document application profile	U	rsp	cnf
Non-basic structural characteristics	U	rsp	cnf
Non-basic document characteristics	U	rsp	cnf
Operational application profile	U	rsp	cnf
Filetransfer capabilities			
BFT capabilities	U	rsp	cnf
Transparent data capability	U	rsp	cnf
Capability result	M	rsp	cnf
User Information	U	rsp	cnf

TABLE 13/T.433

DCPR APDU for Transparent Mode

Field name	Presence	Source	Sink
Application capabilities			
Document application profile	U	rsp	cnf
Document architecture class	U	rsp	cnf
Non-basic structural characteristics	U	rsp	cnf
Non-basic document characteristics	U	rsp	cnf

6.5.3 *DTAM capability procedure*

6.5.3.1 *DTAM capability procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-CAPABILITY request primitive from the requestor;

b) a DCPQ APDU as User Data on a P-CAPAB-DATA indication primitive;

c) a D-CAPABILITY response primitive from the responder; and

d) a P-CAPAB-DATA confirm primitive (that may contain a DCPR APDU).

6.5.3.1.1 *D-CAPABILITY request primitive*

6.5.3.1.1.1 The requesting DTAM-PM forms a DCPQ APDU from parameter values of the D-CAPABILITY request primitive. It issues a P-CAPAB-DATA request primitive. The User Data parameter of the P-CAPAB-DATA request primitive contains the DCPQ APDU.

6.5.3.1.1.2 The requesting DTAM-PM waits for a primitive from the Presentation service-provider, and does not accept any primitive from the requestor other than a D-U-ABORT request primitive.

6.5.3.1.2 *DCPQ APDU*

6.5.3.1.2.1 The responding DTAM-PM receives a DCPQ APDU from its peer as User Data on a P-CAPAB-DATA indication primitive.

6.5.3.1.2.2 In order that the DCPQ APDU may always be acceptable to the responding DTAM-PM, it issues a D-CAPABILITY indication primitive to the responder. The D-CAPABILITY indication primitive parameters are derived from the DCPQ APDU. The DTAM-PM waits for a D-CAPABILITY response primitive from the responder, and does not accept any other primitives from the responder except for the D-U-ABORT request primitive.

6.5.3.1.3 *D-CAPABILITY response primitive*

6.5.3.1.3.1 When the DTAM-PM receives the D-CAPABILITY response primitive, the Result parameter specifies whether the responder has accepted or rejected the DTAM capability requested. The DTAM-PM forms a DCPR APDU using the D-CAPABILITY response primitive parameters. The DCPR APDU is sent as the User Data parameter on the P-CAPAB-DATA response primitive.

6.5.3.1.3.2 If the responder accepted the DTAM capability request, the capability Result field of the outgoing DCPR APDU also specifies the appropriate acceptance value. The DTAM capability is invoked.

6.5.3.1.3.3 If the responder rejected the DTAM capability request, the Result field of the outgoing DCPR APDU contains the appropriate rejection value. The DTAM capability is not established.

6.5.3.1.4 *P-CAPAB-DATA confirm primitive*

6.5.3.1.4.1 The requesting DTAM-PM receives a P-CAPAB-DATA confirm primitive. The following situations are possible:

a) the DTAM capability has been accepted; or

b) the responder has rejected the DTAM capability requested by the requestor.

6.5.3.1.4.2 If the DTAM capability was accepted, the capability Result field of the DCPR APDU specifies the appropriate acceptance value. The requesting DTAM-PM issues a D-CAPABILITY confirm primitive to the requestor based on parameters from the DCPR APDU. The D-CAPABILITY confirm primitive capability Result parameter specifies the appropriate acceptance value. The DTAM capability is invoked. The requesting DTAM user is responsible for the capabilities of his document to be within the responder's capabilities.

6.5.3.1.4.3 If the DTAM capability was rejected by the responder, the Capability Result field of the DCPR APDU on the P-CAPAB-DATA confirm primitive indicates the reason for rejection. The requesting DTAM-PM issues a D-CAPABILITY confirm primitive to the requestor based on parameters from the DCPR APDU. The Capability Result parameter in the D-CAPABILITY confirm primitive contains the appropriate rejection value. The DTAM capability is not established.

6.5.3.2 *DTAM capability procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-CAPABILITY request primitive from the requestor;

b) a DCPQ APDU as User Data on an S-CAPAB-DATA indication primitive;

c) a D-CAPABILITY response primitive from the responder; and

d) an S-CAPAB-DATA confirm primitive (that may contain a DCPR APDU).

6.5.3.2.1 *D-CAPABILITY request primitive*

6.5.3.2.1.1 The requesting DTAM-PM forms a DCPQ APDU from parameter values of the D-CAPABILITY request primitive, and issues an S-CAPAB-DATA request primitive. The User Data parameter of the S-CAPAB-DATA request primitive contains the DCPQ APDU.

6.5.3.2.1.2 The requesting DTAM-PM waits for a primitive from the Session service-provider, and does not accept any primitive from the requestor other than a D-U-ABORT request primitive.

6.5.3.2.2 *DCPQ APDU*

6.5.3.2.2.1 The responding DTAM-PM receives a DCPQ ADPU from its peer as User Data on an S-CAPAB-DATA indication primitive.

6.5.3.2.2.2 In order that the S-CAPAB-DATA indication primitive and its DCPQ APDU may always be acceptable to the responding DTAM-PM, it issues a D-CAPABILITY indication primitive to the responder. The D-CAPABILITY indication primitive parameters are derived from the DCPQ APDU. The DTAM-PM waits for a D-CAPABILITY response primitive from the responder and does not accept any other primitives from the responder except a D-ABORT request primitive.

6.5.3.2.3 *D-CAPABILITY response primitive*

6.5.3.2.3.1 When the DTAM-PM receives the D-CAPABILITY response primitive, the parameters specified in its response primitive contain the Application Capabilities available at the responder. There is no way to issue the result of the capability negotiation explicitly. The DTAM-PM forms a DCPR APDU using the D-CAPABILITY response primitive parameters, and the DCPR APDU is sent as the User Data parameter on the S-CAPAB-DATA response primitive.

6.5.3.2.3.2 In this way, the DTAM capability is negotiated by exchanging the Application Capabilities parameters available at the responder.

6.5.3.2.4 *S-CAPAB-DATA confirm primitive*

6.5.3.2.4.1 The requesting DTAM-PM receives an S-CAPAB-DATA confirm primitive. The DTAM capability is always negotiated by exchanging the Application Capabilities parameters.

6.5.3.2.4.2 If the DTAM capability was accepted, the requesting DTAM-PM issues a D-CAPABILITY confirm primitive to the requestor based on parameters from the DCPR APDU. The final decision of DTAM capability used in the transmission of a document will be made by the requesting DTAM-PM.

6.5.4 *Use of the DCPQ/DCPR APDU fields*

The DCPQ APDU and DCPR APDU fields are used as follows.

6.5.4.1 *Application Capabilities*

This is the Application Capabilities parameter value from the D-CAPABILITY request/response primitives. It appears as the Application Capabilities parameter value of the D-CAPABILITY indication/confirm primitives respectively. This parameter consists of sub-parameters for ODA application capabilities or filetransfer capabilities.

6.5.4.1.1 *ODA application capabilities*

ODA application capabilities consist of sub-parameters specifying the capabilities necessary to handle ODA documents.

6.5.4.1.1.1 *Document application profile*

The value of this parameter is either an Octet String or an ASN.1 object identifier. The Octet String designates the document application profile in line with Recommendation T.73 (Document Application Profile – T.73). The ASN.1 object identifier must conform to the rules specified in ISO 8824 and designate an application profile defined in accordance with the rules specified in Recommendation T.411 (Document Application Profiles).

6.5.4.1.1.2 *Document architecture class*

The value of this parameter is "formatted".

This parameter is only used in the Transparent Mode.

6.5.4.1.1.3 *Non-basic document characteristics*

The value of this parameter is any combination of non-basic document characteristics defined in Recommendation T.414.

6.5.4.1.1.4 *Non-basic structural characteristics*

The value of this parameter is any combination of non-basic structural characteristics defined in Recommendation T.414.

6.5.4.1.1.5 *Operational Application Profile*

The detailed specification of Operational Application Profile is for further study.

6.5.4.1.2 *Filetransfer capabilities*

Filetransfer capabilities are applicable to the DTAM Normal Mode only. It may have the value of one of the following sub-parameters.

6.5.4.1.2.1 *BFT capabilities*

See 6.2.4.3.2.1.

6.5.4.1.2.2 *Transparent data capability*

See 6.2.4.3.2.2.

6.5.4.2 *Capability Result*

If the DCPQ APDU was rejected by the responder, this field is supplied by the responder and is the Capability Result parameter from the D-CAPABILITY response primitive. In this situation, it appears as the Capability Result parameter on the D-CAPABILITY confirm primitive. This field can take one of the following:

- confirmation that all the requested capabilities are available at the DTAM responder (acceptance);
- a list of the requested capabilities that are available at the DTAM responder (acceptance);
- a complete list of non-basic receiving capabilities (acceptance);
- an indication that no extended capabilities are available at the DTAM responder, or that none of the capabilities requested by the requestor are available (rejection).

6.5.4.3 *User Information*

This is the User Information parameter from the D-CAPABILITY request and response primitive. It appears as the User Information parameter of the D-CAPABILITY indication and confirm primitive, if issued.

6.6 *Document bulk transfer*

6.6.1 *Purpose*

6.6.1.1 The document bulk transfer is used to convey the document which contains the ODA, and other types of data to the remote DTAM user. The requestor who requests the document bulk transfer must have the data token. It is provided by the D-TRANSFER service.

6.6.1.2 The document bulk transfer is composed of two different sets of procedures depending on the mode.

1) *Transparent Mode*

a) Transfer procedure for transmission of a complete document;

b) Transfer-Resume procedure for retransmission of a partial document for resuming purpose. This procedure is controlled by the DTAM user;

c) Transfer-Interrupt procedure to interrupt the transmission of a document in case of error;

d) Transfer-Discard procedure to interrupt the transmission of a document in case of error and indicate that the part of the document already transmitted has to be deleted.

In Transparent Mode, the Transfer-Interrupt and the Transfer-Discard procedures result in a D-TRANSFER indication/confirm to the DTAM user to indicate the failure of the transfer. The user is then responsible for initiating a new transfer (complete or partial document).

Figures A-1/T.433 and A-2/T.433 illustrate both protocol sequences for the Transparent Mode.

2) *Normal Mode*

In this procedure, the RTSE services are used to support document bulk transfer.

Figures A-3/T.433 and A-4/T.433 illustrate the basic protocol sequences for the Normal Mode.

6.6.2 *APDUs used*

6.6.2.1 No APDUs are used in this procedure. The Document Information corresponds to a D-TRANSFER request service primitive. There is no D-TRANSFER REQ APDU as such.

6.6.2.2 In the Normal Mode, RTSE is used.

6.6.2.3 In the Transparent Mode, the document bulk transfer is basically managed by DTAM as follows:

– each Document Information, conveyed in a D-TRANSFER request, constitutes an Activity. For each application association, at most one Activity or one interrupted Activity awaiting resumption may exist at any one time;

– the Document Information, which consists of one or more interchange-data-elements as defined in § 9.6.1.1 of Recommendation T.432, is segmented and reassembled into/from one or more segments. Each segment consists of one or more groups of interchange-data-elements and is transferred by the Session data transfer services;

– a Document Information is transferred as a single User Data of the Session data transfer services if checkpointing is not used within the Document Information, otherwise, the Document Information is transferred as a series of Session data transfer services primitives. An example of document segmenting mechanism is given in Figure 2/T.433.

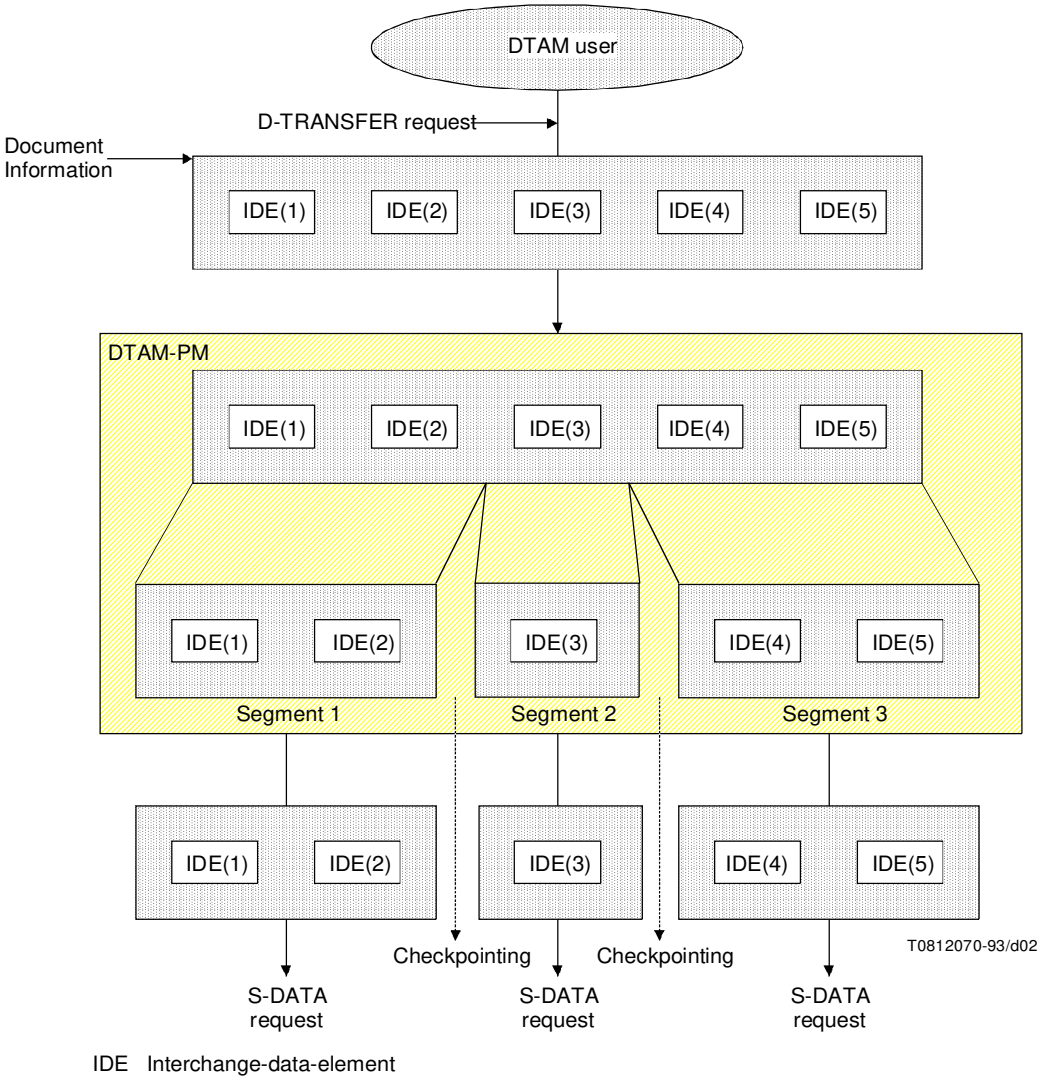


FIGURE 2/T.433

An example of document segmenting mechanism in bulk transfer in the Transparent Mode

6.6.3 *Transfer procedure*

This procedure is used to transfer a complete document.

6.6.3.1 *Transfer procedure mapped onto RTSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TRANSFER request primitive from the requestor (sender of document);

b) an RT-TRANSFER indication primitive;

c) an RT-TRANSFER confirm primitive.

Note – In the case of multiple documents transmission within one association, the above procedure will be applied repeatedly.

6.6.3.1.1 *D-TRANSFER request primitive*

6.6.3.1.1.1 If the requesting DTAM-PM possesses the Data Token and receives a D-TRANSFER request from the requestor, it issues an RT-TRANSFER request primitive and the Document Information in the D-TRANSFER request primitive is mapped onto the User Data in RT-TRANSFER request primitive.

6.6.3.1.2 *RT-TRANSFER indication primitive*

6.6.3.1.2.1 An RT-TRANSFER indication primitive indicates to the responding DTAM-PM that a complete Document Information has been transferred.

6.6.3.1.2.2 If the responding DTAM-PM has secured the complete Document Information, it issues a D-TRANSFER indication primitive to the responder.

6.6.3.1.3 *RT-TRANSFER confirm primitive*

6.6.3.1.3.1 When the requesting DTAM-PM receives the RT-TRANSFER confirm primitive, it issues a D-TRANSFER confirm primitive to the requestor with a Result parameter value taken from the RT-TRANSFER confirm primitive (positive confirm: APDU-transferred or negative confirm: APDU-not-transferred).

6.6.3.2 *Transfer procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-TRANSFER request primitive from the requestor (sender of document);

b) an S-ACTIVITY-START indication primitive, followed by one or more interchange-data-elements as user-data of S-DATA indication primitives each, except the last, followed by a S-MINOR-SYNCHRONIZE indication primitive;

c) an S-MINOR-SYNCHRONIZE confirm primitive;

d) an S-ACTIVITY-END indication primitive;

e) an S-ACTIVITY-END confirm primitive.

Note – In the case of multiple document transmission within one association, the above procedure will be applied repeatedly.

6.6.3.2.1 *D-TRANSFER request primitive*

6.6.3.2.1.1 If the requesting DTAM-PM possesses the Data Token and receives a D-TRANSFER request from the requestor, Document Information in D-TRANSFER request primitive which has an abstract form is segmented by the group (segment) of interchange-data-elements. The segmenting unit (e.g. page, block) depends upon the characteristics of the DTAM-PM. The segmented abstract form is then transformed into the User Data in S-DATA.

6.6.3.2.1.2 The parameter “Document Information Type” contained in the D-TRANSFER request should indicate the “Transfer of a document from its beginning”, and the requesting DTAM-PM issues an S-ACTIVITY-START request primitive and may start transmitting the first segment of interchange-data-elements in an S-DATA request primitive immediately after the S-ACTIVITY-START request primitive is issued, since the S-ACTIVITY-START service is not a confirmed service. All data transfer should take place within an activity.

6.6.3.2.1.3 If the segment of interchange-data-elements transferred is not the last in a series of those segments, the requesting DTAM-PM inserts a checkpoint by issuing an S-MINOR-SYNCHRONIZE request primitive. The requesting DTAM-PM uses only the “explicit confirmation expected” type of minor synchronization. The requesting DTAM-PM may issue further S-DATA request primitives and S-MINOR-SYNCHRONIZE request primitives unless the window-size indicated has been reached.

6.6.3.2.1.4 S-Minor-Synchronization Points shall be located at the end of each segment of interchange-data-elements. Additional Minor Synchronization Points can be requested depending on the evaluation of the storage capacity of the sink and the amount of data to be transmitted. This additional Minor Synchronization Points shall only be located at the end of any interchange-data-elements and not within the element.

6.6.3.2.1.5 If the segment of interchange-data-elements is the only one, or the last in a series of segments of interchange-data-elements, the requesting DTAM-PM issues an S-ACTIVITY-END request primitive. All data transfer must take place within an activity.

6.6.3.2.2 *S-ACTIVITY-START indication primitive, S-DATA PDUs and S-MINOR-SYNCHRONIZE indication primitives*

6.6.3.2.2.1 The responding DTAM-PM receives an S-ACTIVITY-START indication primitive, indicating the start of transfer of Document Information. The responding DTAM-PM receives an S-MINOR-SYNCHRONIZE indication primitive. If the responding DTAM-PM has secured the segment of interchange-data-elements, it issues an S-MINOR-SYNCHRONIZE response primitive.

6.6.3.2.3 *S-MINOR-SYNCHRONIZE confirm primitive*

6.6.3.2.3.1 When the requesting DTAM-PM receives an S-MINOR-SYNCHRONIZE confirm primitive, it assumes that the responding DTAM-PM has secured the segments of interchange-data-elements up to that point.

6.6.3.2.3.2 The requesting DTAM-PM may issue further S-DATA request primitives and S-MINOR-SYNCHRONIZE request primitives unless the window-size indicated has been reached. The window is advanced when an S-MINOR-SYNCHRONIZE confirm primitive is received by the requesting DTAM-PM.

6.6.3.2.3.3 When a complete Document Information has been transmitted, the requesting DTAM-PM issues an S-ACTIVITY-END request primitive.

6.6.3.2.4 *S-ACTIVITY-END indication primitive*

6.6.3.2.4.1 An S-ACTIVITY-END indication primitive indicates to the responding DTAM-PM that a complete Document Information has been transferred.

6.6.3.2.4.2 If the responding DTAM-PM has secured the complete Document Information, it issues a D-TRANSFER indication primitive to the responder, and issues an S-ACTIVITY-END response primitive.

6.6.3.2.5 *S-ACTIVITY-END confirm primitive*

6.6.3.2.5.1 An activity end is an implicit major synchronization point and once successfully confirmed by means of an S-ACTIVITY-END confirm primitive, it indicates to the requesting DTAM-PM that the Document Information has been secured by the responding DTAM-PM. The requesting DTAM-PM may then delete the transferred Document Information.

6.6.3.2.5.2 When the requesting DTAM-PM receives the S-ACTIVITY-END confirm primitive, it issues a D-TRANSFER confirm primitive with a Result parameter value of “document-information-transferred” to the requestor.

6.6.4 *Transfer-resume procedure*

6.6.4.1 *Purpose*

This procedure is used to resume transferring the part of the document which has not been transferred at the previous transmission.

Note – The Transfer Resume procedure is only supported by the Bulk Transfer in Transparent mode.

6.6.4.2 *APDUs used*

No APDUs are used in this procedure.

6.6.4.3 *Transfer-resume procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-TRANSFER request primitive from the requestor (sender of document);

b) an S-ACTIVITY-RESUME indication primitive, followed by one or more interchange-data-elements as user-data of S-DATA indication primitives each, except the last, followed by an S-MINOR-SYNCHRONIZE indication primitive;

c) an S-MINOR-SYNCHRONIZE confirm primitive;

d) an S-ACTIVITY-END indication primitive;

e) an S-ACTIVITY-END confirm primitive.

6.6.4.3.1 *D-TRANSFER request primitive*

6.6.4.3.1.1 If the requesting DTAM-PM possesses the Data Token and receives a D-TRANSFER request from the requestor, Document Information in the D-TRANSFER request primitive, which has an abstract form, is segmented by the group (segment) of interchange-data-elements. The segmenting unit (e.g. page, block) depends upon the characteristics of the DTAM-PM. The segmented abstract form is then transformed into the User Data in S-DATA.

6.6.4.3.1.2 The parameter "Document Information Type" contained in the D-TRANSFER request should indicate the "transfer of a document from a synchronization point", and the requesting DTAM-PM issues an S-ACTIVITY-RESUME request primitive and may continue the transfer procedure by issuing an S-DATA request primitive for the segment of interchange-data-elements following the last confirmed checkpoint. The checkpoint information is from the parameter "Synchronization Point" in the D-TRANSFER request primitive.

6.6.4.3.1.3 Another detailed procedure is followed by §§ 6.6.3.2.1.3, 6.6.3.2.1.4 and 6.6.3.2.1.5.

6.6.4.3.2 *S-ACTIVITY-RESUME indication primitive, S-DATA PDUs and S-MINOR-SYNCHRONIZE indication primitives*

6.6.4.3.2.1 The responding DTAM-PM receives an S-ACTIVITY-RESUME indication primitive, indicating the start of transfer of Document Information. The responding DTAM-PM receives an S-MINOR-SYNCHRONIZE indication primitive. If the responding DTAM-PM has secured the segment of interchange-data-elements, it issues an S-MINOR-SYNCHRONIZE response primitive.

6.6.4.3.3 *S-MINOR-SYNCHRONIZE confirm primitive*

6.6.4.3.3.1 The detailed procedure is followed by §§ 6.6.3.2.3.1, 6.6.3.2.3.2 and 6.6.3.2.3.3.

6.6.4.3.4 *S-ACTIVITY-END indication primitive*

6.6.4.3.4.1 The detailed procedure is followed by §§ 6.6.3.2.4.1 and 6.6.3.2.4.2.

6.6.4.3.5 *S-ACTIVITY-END confirm primitive*

6.6.4.3.5.1 The detailed procedure is followed by §§ 6.6.3.2.5.1 and 6.6.3.2.5.2.

6.6.5 *Transfer-interrupt*

6.6.5.1 *Purpose*

The transfer-interrupt procedure is used by the requesting DTAM-PM to handle a less severe (than those handled by the other error handling procedures) error situation during the transfer procedure, if at least one checkpoint was confirmed during the transfer procedure.

Note – The transfer-resume procedure is only supported by the bulk transfer in Transparent Mode.

6.6.5.2 *APDUs used*

No APDUs are used in this procedure.

6.6.5.3 *Transfer-interrupt procedure mapped on to Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a requesting DTAM-PM problem;

b) an S-ACTIVITY-INTERRUPT indication primitive;

c) an S-ACTIVITY-INTERRUPT confirm primitive.

6.6.5.3.1 *Requesting DTAM-PM problem*

6.6.5.3.1.1 If the requesting DTAM-PM detects a less severe problem and at least one checkpoint was confirmed during the transfer procedure, it issues an S-ACTIVITY-INTERRUPT request primitive with one of the following Reason parameter values:

a) "non-specific error", if the problem was indicated by an exception reporting procedure;

b) "local SS-User error", if the problem is a local requesting DTAM-PM problem.

6.6.5.3.2 *S-ACTIVITY-INTERRUPT indication primitive*

6.6.5.3.2.1 If the responding DTAM-PM receives an S-ACTIVITY-INTERRUPT indication primitive, it issues an S-ACTIVITY-INTERRUPT response primitive followed by D-TRANSFER indication.

6.6.5.3.3 *S-ACTIVITY-INTERRUPT confirm primitive*

6.6.5.3.3.1 If the requesting DTAM-PM receives an S-ACTIVITY-INTERRUPT confirm primitive, it issues D-TRANSFER confirm to the requesting DTAM user.

6.6.6 *Transfer-discard*

6.6.6.1 *Purpose*

The transfer-discard procedure is used by the requesting DTAM-PM to escape from a more severe (than those handled by the transfer-interrupt procedure) error situation, or a less severe error situation if no checkpoint was confirmed, during the transfer procedure.

Note – The transfer-resume procedure is only supported by the bulk transfer in Transparent Mode.

6.6.6.2 *APDUs used*

No APDUs are used in this procedure.

6.6.6.3 *Transfer-discard procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a requesting DTAM-PM problem;

b) an S-ACTIVITY-DISCARD indication primitive;

c) an S-ACTIVITY-DISCARD confirm primitive.

6.6.6.3.1 *Requesting DTAM-PM problem*

6.6.6.3.1.1 If the requesting DTAM-PM detects a more severe problem, or a less severe problem if no checkpoint was confirmed during the transfer procedure, it issues an S-ACTIVITY-DISCARD request primitive with one of the following Reason parameter values:

a) "non-specific error", if the problem was indicated by an error reporting procedure;

b) "local SS-User error", or "unrecoverable procedural error", if the problem is a local requesting DTAM-PM problem.

6.6.6.3.2 *S-ACTIVITY-DISCARD indication primitive*

6.6.6.3.2.1 If the responding DTAM-PM receives an S-ACTIVITY-DISCARD indication primitive, it issues an S-ACTIVITY-DISCARD response primitive followed by D-TRANSFER indication in Transparent Mode.

6.6.6.3.2.2 The responding DTAM-PM deletes all knowledge and contents of the associated DTAM user information (segments of Document Information) so far received.

6.6.6.3.2.3 If the responding DTAM-PM has already issued a D-TRANSFER indication primitive, it performs the session-abort procedure by issuing an S-U-ABORT request.

6.6.6.3.3 *S-ACTIVITY-DISCARD confirm primitive*

If the requesting DTAM-PM receives an S-ACTIVITY-DISCARD confirm primitive, it issues D-TRANSFER confirm to the requesting DTAM user in Transparent Mode.

6.7 *Document unconfirmed manipulation*

Document unconfirmed manipulation is used by the requestor to manipulate the constituents of ODA and Operational Structure which are commonly handled by both communication entities. Document unconfirmed manipulation consists of document create operation, document delete operation, document modify operation, document call operation and document rebuild operation.

6.7.1 *Document create operation*

6.7.1.1 *Purpose*

The document create operation procedure is used by the requestor of document manipulation to add the constituents of ODA and Operational Structure to a document without any confirmation of the create manipulation.

6.7.1.2 *APDUs used*

The document create operation procedure uses D-CREATE(DCR) APDU.

6.7.1.2.1 *DCR APDU*

The field of the DCR APDU is listed in Table 14/T.433.

TABLE 14/T.433

DCR APDU fields

Field name	Presence	Source	Sink
Create information	M	req	ind

6.7.1.3 *Document create operation procedure*

6.7.1.3.1 *Document create operation procedure mapped onto Presentation service (Normal Mode)*

Procedure is driven by the following events:

a) a D-CREATE request primitive from the requestor;

b) a DCR APDU as User Data of a P-DATA indication primitive.

6.7.1.3.1.1 *D-CREATE request primitive*

6.7.1.3.1.1.1 If the requesting DTAM-PM receives a D-CREATE request primitive, a DCR APDU is formed from the parameter values of the D-CREATE request primitive and transferred as User Data of A P-DATA request primitive. This may be done outside of an activity.

6.7.1.3.1.2 *DCR APDU*

6.7.1.3.1.2.1 If the responding DTAM-PM receives the DCR APDU as User Data of a P-DATA indication primitive, the responding DTAM-PM issues a D-CREATE indication primitive to the responder. The D-CREATE indication parameter is derived from the DCR APDU.

6.7.1.4 *Use of the DCR APDU fields*

The DCR APDU fields are used as specified below.

6.7.1.4.1 *Create Information*

This is the Create Information parameter value from the D-CREATE request primitive. It appears as the Create Information parameter value of the D-CREATE indication primitive.

6.7.2 *Document delete operation*

6.7.2.1 *Purpose*

The document delete operation procedure is used by the requestor of document manipulation to delete the constituents of ODA and Operational Structure of an existing document without any confirmation of the delete operation.

6.7.2.2 *APDUs used*

The document delete operation procedure uses D-DELETE(DDL) APDU.

6.7.2.2.1 *DDL APDU*

The field of the DDL APDU is listed in Table 15/T.433.

TABLE 15/T.433

DDL APDU fields

Field name	Presence	Source	Sink
Delete information	M	req	ind

6.7.2.3 *Document delete operation procedure*

6.7.2.3.1 *Document delete operation procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-DELETE request primitive from the requestor;

b) a DDL APDU as User Data of a P-DATA indication primitive.

6.7.2.3.1.1 *D-DELETE request primitive*

6.7.2.3.1.1.1 If the requesting DTAM-PM receives a D-DELETE request primitive, a DDL APDU is formed from the parameter values of D-DELETE request primitive and transferred as User Data of a P-DATA request primitive. This may be done outside of an activity.

6.7.2.3.1.2 *DDL APDU*

6.7.2.3.1.2.1 If the responding DTAM-PM receives the DDL APDU as User Data of a P-DATA indication primitive, the responding DTAM-PM issues a D-DELETE indication primitive to the responder. The D-DELETE indication primitive parameter is derived from the DDL APDU.

6.7.2.4 *Use of the DDL APDU fields*

The DDL APDU fields are used as specified below.

6.7.2.4.1 *Delete Information*

This is the Delete Information parameter value from the D-DELETE request primitive. It appears as the Delete Information parameter value of the D-DELETE indication primitive.

6.7.3 *Document modify operation*

6.7.3.1 *Purpose*

The document modify operation procedure is used by the requestor of document manipulation to modify the attributes of constituents of ODA and Operational Structure of an existing document without any confirmation of the modifying operation.

6.7.3.2 *APDUs used*

The document modify operation procedure uses D-MODIFY(DMD) APDU.

6.7.3.2.1 *DMD APDU*

The field of the DMD APDU is listed in Table 16/T.433.

TABLE 16/T.433

DMD APDU fields

Field name	Presence	Source	Sink
Modify information	M	req	ind

6.7.3.3 *Document modify operation procedure*

6.7.3.3.1 *Document modify operation procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-MODIFY request primitive from the requestor;

b) a DMD APDU as User Data of a P-DATA indication primitive.

6.7.3.3.1.1 *D-MODIFY request primitive*

6.7.3.3.1.1.1 If the requesting DTAM-PM receives a D-MODIFY request primitive, a DMD APDU is formed from the parameter values of D-MODIFY request primitive and transferred as User Data of a P-DATA request primitive. This may be done outside of an activity.

6.7.3.3.1.2 *DMD APDU*

6.7.3.3.1.2.1 If the responding DTAM-PM receives the DMD APDU as User Data of a P-DATA indication primitive, the responding DTAM-PM issues a D-MODIFY indication primitive to the responder. The D-MODIFY indication primitive parameter is derived from the DMD APDU.

6.7.3.4 *Use of the DMD APDU fields*

The DMD APDU fields are used as specified below.

6.7.3.4.1 *Modify Information*

This is the Modify Information parameter value from the D-MODIFY request primitive. It appears as the Modify Information parameter value of the D-MODIFY indication primitive.

6.7.4 *Document call operation*

6.7.4.1 *Purpose*

This document call operation procedure is used by the requestor of document manipulation to address or to read an object of Operational Structure which contains a sequence of DTAM protocol data units (with some restrictions, i.e. that only D-CREATE, D-DELETE and D-MODIFY can appear in this sequence). These protocol data units are applicable to the existing document.

6.7.4.2 *APDUs used*

The document call operation procedure uses D-CALL(DCL) APDU.

6.7.4.2.1 *DCL APDU*

The field of the DCL APDU is listed in Table 17/T.433.

TABLE 17/T.433

DCL APDU fields

Field name	Presence	Source	Sink
Call information	M	req	ind

6.7.4.3 *Document call operation procedure*

6.7.4.3.1 *Document call operation procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-CALL request primitive from the requestor;

b) a DCL APDU as User Data of a P-DATA indication primitive.

6.7.4.3.1.1 *D-CALL request primitive*

6.7.4.3.1.1.1 If the requesting DTAM-PM receives a D-CALL request primitive, a DCL APDU is formed from the parameter values of D-CALL request primitive and transferred as user-data of a P-DATA request primitive. This may be done outside of an activity.

6.7.4.3.1.2 *DCL APDU*

6.7.4.3.1.2.1 If the responding DTAM-PM receives the DCL APDU as User Data of a P-DATA indication primitive, the responding DTAM-PM issues a D-CALL indication primitive to the responder. The D-CALL indication primitive parameter is derived from the DCL APDU.

6.7.4.4 *Use of the DCL APDU fields*

The DCL APDU fields are used as specified below.

6.7.4.4.1 *Call Information*

This is the Call Information parameter value from the D-CALL request primitive. It appears as the Call Information parameter value of the D-CALL indication primitive.

6.7.5 *Document rebuild operation*

For further study.

6.8 *Document confirmed manipulation*

For further study.

6.9 *Typed Data Transfer*

6.9.1 *Purpose*

Typed data transmission is used independent of the data token and is issued from both DTAM users when required.

6.9.2 *APDUs used*

The Typed Data Transfer procedure uses D-TYPED-DATA(DTD) APDU.

6.9.2.1 *DTD APDU*

The field of the DTD APDU is listed in Table 18/T.433.

TABLE 18/T.433

DTD APDU fields

Field name	Presence	Source	Sink
Typed-Data Information	M	req	ind

6.9.3 *Typed Data Transfer procedure*

6.9.3.1 *Typed Data Transfer procedure mapped onto Presentation service*

This procedure is driven by the following events:

a) a D-TYPED-DATA request primitive from the requestor;

b) a DTD APDU as User Data of a P-TYPED-DATA indication primitive.

6.9.3.1.1 *D-TYPED-DATA request primitive*

6.9.3.1.1.1 If the requesting DTAM-PM receives a D-TYPED-DATA request primitive, a DTD APDU is formed from the parameter values of D-TYPED-DATA request primitive and transferred as User Data of a P-DATA request primitive.

6.9.3.1.2 *DTD APDU*

6.9.3.1.2.1 If the responding DTAM-PM receives the DTD APDU as User Data of a P-TYPED-DATA indication primitive, the responding DTAM-PM issues a D-TYPED-DATA indication primitive to the responder. The D-TYPED-DATA indication primitive parameter is derived from the DTD APDU.

6.9.4 *Use of the DTD APDU fields*

The DTD APDU fields are used as specified below.

6.9.4.1 *Typed-Data Information*

This is the Typed-Data String Information parameter value from the D-TYPED-DATA request primitive. It appears as the Typed-Data String Information parameter value of the D-TYPED-DATA indication primitive.

6.10 *Remote document access*

For further study.

6.11 *Remote document management*

For further study.

6.12 *Token control*

6.12.1 *Token please control*

6.12.1.1 *Purpose*

The token please procedure is used by a requestor (receiver of documents) to request the token from the responder (sender of documents).

6.12.1.2 *APDUs used*

The token please procedure uses the D-TOKEN-PLEASE (DTP) APDU in the Normal Mode without RTSE. In the case of bulk transfer in Normal Mode and Transparent Mode, no APDU is applied.

6.12.1.2.1 *DTP APDU*

The field of the DTP APDU is listed in Table 19/T.433.

TABLE 19/T.433

DTP APDU fields

Field name	Presence	Source	Sink
Priority	U	req	ind

6.12.1.3 *Token please procedure*

6.12.1.3.1 *Token please procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TOKEN-PLEASE request primitive from the requestor; and

b) a DTD APDU as User Data of a P-TOKEN-PLEASE indication primitive.

6.12.1.3.1.1 *D-TOKEN-PLEASE request primitive*

6.12.1.3.1.1.1 If the requesting DTAM-PM does not possess the data token and receives a D-TOKEN-PLEASE request from the requestor, a DTP APDU is formed from the parameter values of D-TOKEN-PLEASE request primitive and transferred as User Data of a P-TOKEN-PLEASE request primitive. This may be done either inside or outside an activity.

6.12.1.3.1.2 *DTP APDU*

6.12.1.3.1.2.1 If the responding DTAM-PM receives the DTP APDU as User Data of a P-TOKEN-PLEASE indication primitive, the responding DTAM-PM issues a D-TOKEN-PLEASE indication primitive to the responder. The D-TOKEN-PLEASE indication primitive parameter is derived from the DTP APDU.

6.12.1.3.2 *Token please procedure mapped onto RTSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TOKEN-PLEASE request primitive from the requestor; and

b) an RT-TURN-PLEASE indication primitive.

6.12.1.3.2.1 *D-TOKEN-PLEASE request primitive*

6.12.1.3.2.1.1 If the requesting DTAM-PM does not possess the data token and receives a D-TOKEN-PLEASE request from the requestor, it issues an RT-TURN-PLEASE request primitive with the Priority parameter value taken from the D-TOKEN-PLEASE request primitive. This may be done either inside or outside an activity of document transfer.

6.12.1.3.2.2 *DTP APDU*

6.12.1.3.2.2.1 If the responding DTAM-PM receives an RT-TURN-PLEASE indication primitive, the responding DTAM-PM issues a D-TOKEN-PLEASE indication primitive to the responder. The D-TOKEN-PLEASE indication primitive parameter is derived from the RT-TURN-PLEASE indication primitive.

6.12.1.3.3 *Token please procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-TOKEN-PLEASE request primitive from the requestor;

b) an S-TOKEN-PLEASE indication primitive.

6.12.1.3.3.1 *D-TOKEN-PLEASE request primitive*

6.12.1.3.3.1.1 If the requesting DTAM-PM does not possess the data token and receives a D-TOKEN-PLEASE request from the requestor, DTAM-PM issues an S-TOKEN-PLEASE request primitive. This may be done either inside or outside an activity.

6.12.1.3.3.2 *Implicit DTP APDU*

6.12.1.3.3.2.1 If the responding DTAM-PM receives an S-TOKEN-PLEASE indication primitive without any APDU on its User Data, the responding DTAM-PM issues a D-TOKEN-PLEASE indication primitive to the responder.

6.12.1.4 Use of the DTP APDU fields

The DTP APDU fields are used as specified below.

6.12.1.4.1 *Priority*

This parameter is the priority of the action, governed by the data token, that the requestor of the D-TOKEN-PLEASE service wishes to carry out. This parameter has to be supplied by the requestor of the D-TOKEN-PLEASE service.

This is the priority parameter value of the D-TOKEN-PLEASE request primitive. It appears as the priority parameter of the D-TOKEN-PLEASE indication primitive.

The value of this field is transparent to the DTAM-PM.

6.12.2 *Token give control*

6.12.2.1 *Purpose*

6.12.2.1.1 The token-give procedure is used by a requestor (sender of documents) to give the token to the responder (receiver or documents).

6.12.2.1.2 The requestor becomes the receiver and the responder becomes the sender.

6.12.2.2 *APDUs used*

No APDUs are used in this procedure.

6.12.2.3 *Token give procedure*

6.12.2.3.1 *Token give procedure mapped onto Presentation service (Normal Mode)*

This procedure is driven by the following events:

a) a D-TOKEN-GIVE request primitive;

b) a P-TOKEN-GIVE indication primitive.

6.12.2.3.1.1 *D-TOKEN-GIVE request primitive*

6.12.2.3.1.1.1 If the requesting DTAM-PM possesses the token and receives a D-TOKEN-GIVE request primitive from the requestor, it issues a P-TOKEN-GIVE request primitive and becomes the responding DTAM-PM.

This may be done only outside an activity.

6.12.2.3.1.2 *P-TOKEN-GIVE indication primitive*

6.12.2.3.1.2.1 If the responding DTAM-PM receives a P-TOKEN-GIVE indication primitive, the responding DTAM-PM issues a D-TOKEN-GIVE indication primitive to the responder. The responding DTAM-PM becomes the requesting DTAM-PM.

6.12.3 *CONTROL-GIVE*

6.12.3.1 *Purpose*

6.12.3.1.1 The CONTROL-GIVE procedure is used by a requestor to give all the tokens to the responder.

6.12.3.1.2 The requestor becomes the receiver and the responder becomes the sender.

6.12.3.2 *APDUs used*

No APDUs are used in this procedure.

6.12.3.3 *CONTROL-GIVE procedure*

6.12.3.3.1 *CONTROL-GIVE procedure mapped onto RTSE service (Normal Mode)*

This procedure is driven by the following events:

a) a D-CONTROL-GIVE request primitive;

b) an RT-TURN-GIVE indication primitive.

6.12.3.3.1.1 *D-CONTROL-GIVE request primitive*

6.12.3.3.1.1.1 If the requesting DTAM-PM possesses the tokens and receives a D-CONTROL-GIVE request primitive from the requestor, it issues an RT-TURN-GIVE request primitive and becomes the responding DTAM-PM. This may be done only outside an activity.

6.12.3.3.1.2 *RT-TURN-GIVE indication primitive*

6.12.3.3.1.2.1 If the responding DTAM-PM receives an RT-TURN-GIVE indication primitive, the responding DTAM-PM issues a D-CONTROL-GIVE indication primitive to the responder. The responding DTAM-PM becomes the requesting DTAM-PM.

6.12.3.3.2 *CONTROL-GIVE Procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a D-CONTROL-GIVE request primitive from the requestor;

b) an S-CONTROL-GIVE indication primitive.

6.12.3.3.2.1 *D-CONTROL-GIVE request primitive*

6.12.3.3.2.1.1 If the requesting DTAM-PM possess the tokens and receives a D-CONTROL-GIVE request primitive from the requestor, it issues an S-CONTROL-GIVE request primitive and becomes the responding DTAM-PM. This may be done only outside an activity.

6.12.3.3.2.2 *S-CONTROL-GIVE indication primitive*

6.12.3.3.2.2.1 If the responding DTAM-PM receives an S-CONTROL-GIVE indication primitive, the responding DTAM-PM issues a D-CONTROL-GIVE indication primitive to the responder. The responding DTAM-PM becomes the requesting DTAM-PM.

6.13 *Exception report*

6.13.1 *User-exception-report*

6.13.1.1 *Purpose*

The user-exception-report procedure is used by the responding DTAM-PM to report an error situation to the requesting DTAM-PM during document bulk transfer (Transparent Mode).

6.13.1.2 *APDUs used*

No APDUs are used in this procedure.

6.13.1.3 *User-exception-report procedure*

6.13.1.3.1 *User-exception-report procedure mapped onto Session service (Transparent Mode)*

This procedure is driven by the following events:

a) a responding DTAM-PM problem;

b) an S-U-EXCEPTION-REPORT indication primitive.

6.13.1.3.1.1 *Receiving DTAM-PM problem*

6.13.1.3.1.1.1 If the responding DTAM-PM detects a problem, it issues an S-U-EXCEPTION-REPORT request primitive. Depending on the severity of the detected error, the value of the Reason parameter of the S-U-EXCEPTION-REPORT request primitive is as follows:

a) in severe problem situations, the value “receiving ability jeopardized (unable to continue the association)” is used;

b) in exceptional circumstances, the responding DTAM-PM may have to delete a partially received document information, even though some minor synchronization points have been confirmed. In this case, the value “unrecoverable procedure error” is used;

c) if the responding DTAM-PM is not willing to complete a transfer procedure, the value “non-specific error” is used;

d) if the requesting DTAM-PM resumes a transfer procedure already finished by the responding DTAM-PM, the value "sequence error" is used;

e) for all other less severe error situations, the value “local SS-User error” is used.

6.13.1.3.1.2 *S-U-EXCEPTION-REPORT indication primitive*

6.13.1.3.1.2.1 If the requesting DTAM-PM receives an S-U-EXCEPTION-REPORT indication primitive, it performs one of the following procedures depending on the Reason parameter value of the S-U-EXCEPTION-REPORT indication primitive (in the Transparent Mode, only Reliable Transfer Mode without RTSE is available):

a) with a value “receiving ability jeopardized (unable to continue the association)”, the DTAM provider-abort procedure is performed;

b) with a value “unrecoverable procedure error”, the transfer-discard procedure followed by D-TRANSFER service (confirmation) is performed;

c) with a value “non-specific error”, the transfer-discard procedure followed by D-TRANSFER service (confirmation) is performed;

d) with a value “sequence error”, the transfer-discard procedure is performed and the requesting DTAM-PM issues a D-TRANSFER confirm primitive with a Result parameter value of “document-information-transferred” to the requestor and the transfer procedure is finished;

e) with a value “local SS-User error” and at least one confirmed checkpoint in the transfer procedure, the transfer-interrupt procedure followed by D-TRANSFER service (indication and confirmation) are performed. If no checkpoint was confirmed in the transfer procedure, the transfer-discard procedure followed by D-TRANSFER service (confirmation) is performed.

6.13.2 *Provider-Exception-Report*

6.13.2.1 *Purpose*

If the session service-provider (Transparent Mode) at the responding side detects an unexpected situation during an activity, not covered by other services, an S-P-EXCEPTION-REPORT indication primitive is respectively issued to both DTAM-PMs.

6.13.2.2 *APDUs used*

No APDUs are used in this procedure.

6.13.2.3 *Provider-Exception-Report Procedure*

6.13.2.3.1 *Provider-Exception-Report Procedure mapping onto Session service (Transparent Mode)*

This procedure is driven by the following event:

– an S-P-EXCEPTION-REPORT indication primitive.

6.13.2.3.1.1 *S-P-EXCEPTION-REPORT indication primitive*

6.13.2.3.1.1.1 The responding DTAM-PM ignores an S-P-EXCEPTION-REPORT indication primitive.

6.13.2.3.1.1.2 If the requesting DTAM-PM receives an S-P-EXCEPTION-REPORT indication primitive, it may perform one of the following procedures:

a) if at least one checkpoint was confirmed in the transfer procedure, the transfer-interrupt procedure followed by the D-TRANSFER service (confirmation); or

b) if no checkpoint was confirmed in the transfer procedure, the transfer-discard procedure followed by the D-TRANSFER service (confirmation); or

c) the DTAM provider-abort procedure.

6.14 *Rules of extensibility*

In addition to the procedures stated above, the following applies when processing the APDUs defined in this part of Recommendation T.433:

a) fields are ignored that are not defined in this part of Recommendation T.433 in DTAM association establishing phase PDUs (DINQ, DINR and DAB); and

b) for fields defined as having a maximum length in this part of Recommendation T.433, that portion of any value beyond the maximum length is ignored.

7 Mapping to the lower services

7.1 *Mapping to the presentation and ACSE services*

This section defines how a DTAM-PM transfers APDUs by means of:

a) the ACSE services; or

b) the presentation-services,

when RTSE is not used.

Table 21/T.433 lists the overview of ACSE or presentation-service mapping.

7.1.1 *Mapping on the ACSE services*

7.1.1.1 *Association-establishment procedure*

Association-establishment procedure takes place concurrently with the underlying ACSE association establishment.

7.1.1.1.1 *Directly mapped parameters*

The following parameters of D-INITIATE service primitives are mapped directly onto the corresponding parameters of the A-ASSOCIATE service primitives:

a) Application Context Name;

b) Calling AP Title;

c) Calling AP Invocation-Identifier;

d) Calling AE Qualifier;

e) Calling AE Invocation-Identifier;

f) Called AP Title;

g) Called AP Invocation-Identifier;

h) Called AE Qualifier;

i) Called AE Invocation-Identifier;

J) Responding AP Title;

k) Responding AP Invocation-Identifier;

l) Responding AE Qualifier;

m) Responding AE Invocation-Identifier;

n) Calling Presentation Address;

o) Called Presentation Address;

p) Responding Presentation Address;

q) Presentation Context Definition List;

r) Presentation Context Definition Result List;

s) Presentation Requirements;

t) Initial Assignment of Token;

u) Quality of Service;

v) Default Presentation Context Name (see note);

w) Default Presentation Context Result (see note);

Note – Use of this parameter is for further study.

7.1.1.1.2 *Parameters not used*

The following parameters of A-ASSOCIATE service primitives are not used:

a) Initial Synchronization Point Serial Number;

b) Diagnostic.

7.1.1.1.3 *Use of the other A-ASSOCIATE request and indication primitive parameters*

7.1.1.1.3.1 *Mode*

This parameter shall be supplied by the requestor of the association in the A-ASSOCIATE request primitive, and shall have the value “Normal Mode”.

7.1.1.1.3.2 *User Information*

For both the A-ASSOCIATE request and indication primitives, the User Information parameter is used to carry the D-INITIATE-REQ APDU.

7.1.1.1.3.3 *Session requirements*

This parameter is set by the association-initiating DTAM-PM to select the following functional units by means of the “Telematic requirements parameter” in D-INITIATE service primitives as shown in Table 20/T.433.

TABLE 20/T.433

Mapping into/out of the session requirements

"Telematic requirements"	Functional units
Kernel	Kernel functional unit
Token control	Half-duplex functional unit
Typed data transmission	Typed data functional unit
Capability management	Capability data exchange functional unit

Note – The use of duplex functional unit and negotiated release functional unit are for further study.

7.1.1.1.3.4 *Session Connection Identifier*

The association-initiating DTAM-PM will supply a Session Connection Identifier, which will be used to uniquely identify the session-connection. This identifier is formed of the following components:

a) SS-User Reference;

b) Common Reference;

c) Additional Reference Information (optionally).

The SS-User Reference is conveyed as the Calling SS-User Reference by the association-initiating DTAM-PM. Common Reference and Additional Reference Information are conveyed in similarly named parameters of the P-CONNECT primitive.

Each component, when present, will contain a data element of the appropriate type from the following definitions:

CallingSS-UserReference **::= PresentationAddress** *-- of the requestor*

CommonReference **::= UTCTime**

AdditionalReferenceInformation **::= T.61 String**

The PresentationAddress is represented as a string of octets.

7.1.1.1.4 *Use of the other A-ASSOCIATE response and confirm primitive parameters*

7.1.1.1.4.1 *User Information*

This parameter only has relevance if the application-association is accepted by the ACSE service-provider.

For both the A-ASSOCIATE response and confirmation primitives, the User Information parameter is used to carry the D-INITIATE-RESP APDU, whether the application-association is accepted or is rejected by the association-responding DTAM-PM.

7.1.1.1.4.2 *Result*

For the A-ASSOCIATE response primitive the Result parameter is set by the association responding DTAM-PM as follows:

a) if the association-responding DTAM-PM rejects the application-association, the value of this parameter is set to either "rejected by responder (transient)" or "rejected by responder (permanent)";

b) if the association-responding DTAM-PM accepts the request, the value of this parameter is derived from the Result parameter of the D-INITIATE response primitive.

7.1.1.1.4.3 *Result source*

To be provided by the editor.

7.1.1.1.4.4 *Session requirements*

This parameter has the same values as in the A-ASSOCIATE request and indication primitives.

7.1.1.1.4.5 *Session connection identifier*

This parameter has the same values as in the A-ASSOCIATE indication primitives. The Calling SS-User Reference value of the A-ASSOCIATE indication primitive is returned as a Called SS-User Reference by the association-responding DTAM-PM.

7.1.1.2 *Association-release procedure*

Association-release procedure takes place concurrently with the underlying ACSE association release.

7.1.1.2.1 *Directly mapped parameters*

The following parameter of D-TERMINATE service primitives is mapped directly onto the corresponding parameters of the A-RELEASE service primitives:

– User Data (on User Information).

7.1.1.2.2 *Parameters not used*

The following parameter of the A-RELEASE service is not used:

– Reason.

7.1.1.2.3 *Use of the other A-RELEASE response and confirm primitive parameters*

7.1.1.2.3.1 *Result*

The value of this parameter is “Affirmative”.

7.1.1.3 *Association-provider-abort procedure*

The use of the A-P-ABORT indication primitive parameters are defined in Recommendation X.217.

7.1.1.4 *Association-user-abort procedure*

Association-user-abort procedure takes place concurrently with the underlying ACSE association abort.

7.1.1.4.1 *Use of A-ABORT request and indication primitive parameters*

7.1.1.4.1.1 *Abort source*

This parameter value is “requestor”.

7.1.1.4.1.2 *User information*

This parameter value is the DAB APDU.

7.1.2 *Mapping on the Presentation services*

7.1.2.1 *Use of the P-DATA request and indication primitive parameters*

7.1.2.1.1 *User Data*

The following DTAM APDUs are conveyed by this parameter:

a) D-CREATE APDU

b) D-DELETE APDU

c) D-MODIFY APDU

d) D-CALL APDU.

7.1.2.2 *Use of the P-TYPED-DATA request and indication primitive parameters*

7.1.2.2.1 *User Data*

The D-TYPED-DATA APDU is conveyed by this parameter.

7.1.2.3 *Token please procedure*

When the DTAM user issues a D-TOKEN-PLEASE service primitive, this results into a P-TOKEN-PLEASE.

7.1.2.3.1 *Use of the P-TOKEN-PLEASE request and indication primitive parameters*

This parameter takes a value which corresponds to requesting the data token.

7.1.2.3.1.1 *Tokens*

The responding DTAM-PM will only request the data token.

7.1.2.3.1.2 *User Data*

The D-TOKEN-PLS APDU is conveyed by this parameter.

7.1.2.4 *Token give procedure*

When the DTAM-user issues a D-TOKEN-GIVE service primitive, this results into a P-TOKEN-GIVE.

7.1.2.4.1 *Use of the P-TOKEN-GIVE request and indication primitive parameters*

7.1.2.4.1.1 *Tokens*

This parameter takes a value which corresponds to giving the data token.

7.1.2.5 *Capability data exchange procedure*

7.1.2.5.1 *Use of the P-CAPAB-DATA service parameters*

7.1.4.5.1.1 *User Data*

The following DTAM APDUs are conveyed by this parameter:

a) D-CAPABILITY-REQ APDU;

b) D-CAPABILITY-RESP APDU.

7.2 *Mapping to the presentation and RTSE services*

7.2.1 *Mapping on the RTSE services*

Table 22/T.433 lists the overview of the mapping to the presentation and RTSE services.

7.2.1.1 *Association-establishment procedure (mapping onto RT-OPEN)*

Association-establishment procedure takes place concurrently with the underlying RTSE RT-OPEN procedure.

TABLE 21/T.433

Overview of the mapping to the presentation and ACSE services without RTSE

Functional units	Service primitive		Protocol elements (PDU)	Mapping DTAM PDU to ACSE service/ Presentation service	
Association use control	D-INITIATE	req/ind	D-INITIATE-REQ PDU	A-ASSOCIATE	req/ind
		rsp/cnf	D-INITIATE-RESP PDU	A-ASSOCIATE	rsp/cnf
	D-TERMINATE	req/ind	D-TERMINATE-REQ PDU	A-RELEASE	req/ind
		rsp/cnf	D-TERMINATE-RESP PDU	A-RELEASE	rsp/cnf
	D-U-ABORT	req/ind	D-ABORT PDU	A-ABORT	req/ind
	D-P-ABORT	ind	None	A-P-ABORT	ind
Capability	D-CAPABILITY	req/ind	D-CAPABILITY-REQ PDU	P-CAPAB-DATA	req/ind
		rsp/cnf	D-CAPABILITY-RES PDUP	P-CAPAB-DATA	rsp/cnf
Document uncomfirmed manipulation	D-CREATE	req/ind	D-CREATE PDU	P-DATA	req/ind
	D-DELETE	req/ind	D-DELETE PDU	P-DATA	req/ind
	D-MODIFY	req/ind	D-MODIFY PDU	P-DATA	req/ind
	D-CALL	req/ind	D-CALL PDU	P-DATA	req/ind
	D-REBUILD (See Note)	req/ind	D-REBUILD PDU (See Note)	P-DATA	req/ind
Token control	D-TOKEN-GIVE	req/ind	None	P-TOKEN-GIVE	req/ind
	D-TOKEN-PLS	req/ind	D-TOKEN-PLS PDU	P-TOKEN-PLS	req/ind
	D-CONTROL-GIVE	req/ind	None	P-CONTROL-GIVE	req/ind
Typed data transmission	D-TYPED-DATA	req/ind	D-TYPED-DATA PDU	P-TYPED-DATA	req/ind

Note – This DTAM service or PDU is for further study.

TABLE 22/T.433

Overview of the mapping to the presentation and RTSE services

Functional units	Service primitive		Protocol elements (PDU)	Mapping DTAM PDU to RTSE service/ Presentation service	
Association use control	D-INITIATE	req/ind	D-INITIATE-REQ PDU	RT-OPEN	req/ind
		rsp/cnf	D-INITIATE-RESP PDU	RT-OPEN	rsp/cnf
	D-TERMINATE	req/ind	D-TERMINATE-REQ PDU	RT-CLOSE	req/ind
		rsp/cnf	D-TERMINATE-RESP PDU	RT-CLOSE	rsp/cnf
	D-U-ABORT	req/ind	None	RT-U-ABORT	req/ind
	D-P-ABORT	ind	None	RT-P-ABORT	ind
Capability	D-CAPABILITY	req/ind	D-CAPABILITY-REQ PDU	P-CAPAB-DATA	req/ind
		rsp/cnf	D-CAPABILITY-RESP PDU	P-CAPAB-DATA	rsp/cnf
Document bulk transfer	D-TRANSFER	req	None	RT-TRANSFER	req
		ind	None		ind
		cnf	None		cnf
Token control	D-TOKEN-PLS	req/ind	None	RT-TURN-PLEASE	req/ind
	D-CONTROL-GIVE	req/ind	None	RT-TURN-GIVE	req/ind

7.2.1.1.1 *Directly mapped parameters*

The following parameters of D-INITIATE service primitives are mapped directly onto the corresponding parameters of the RT-OPEN service primitives:

a) Application Context Name;

b) Calling AP Title;

c) Calling AP Invocation-Identifier;

d) Calling AE Qualifier;

e) Calling AE Invocation-Identifier;

f) Called AP Title;

g) Called AP Invocation-Identifier;

h) Called AE Qualifier;

i) Called AE Invocation-Identifier;

j) Responding AP Title;

k) Responding AP Invocation-Identifier;

l) Responding AE Qualifier;

m) Responding AE Invocation-Identifier;

n) Calling Presentation Address;

o) Called Presentation Address;

p) Responding Presentation Address;

q) Presentation Context Definition List;

r) Presentation Context Definition Result List;

s) Default Presentation Context Name (see note);

t) Default Presentation Context Result (see note).

Note – Use of this parameter is for further study.

7.2.1.1.2 *Use of the other RT-OPEN request and indication primitive parameters*

7.2.1.1.2.1 *Mode*

This parameter shall be supplied by the requestor of the association in the RT-OPEN request primitive, and shall have the value “Normal Mode”.

7.2.1.1.2.2 *User Information*

For both the RT-OPEN request and indication primitives, the User Information parameter is used to carry the D-INITIATE-REQ APDU.

7.2.1.1.2.3 *Dialogue-mode*

This parameter should be supplied by the requestor of the association in the RT-OPEN request primitive, and shall have the value “two-way-alternative interaction”.

7.2.1.1.2.4 *Initial turn*

This parameter should be supplied by the requestor of the association in the RT-OPEN request primitive to set the initial token (turn) to association-initiator or association-responder. The value of this parameter shall be transferred from that of the parameter “Initial Assignment of Token” in D-INITIATE request primitive.

7.2.1.1.2.5 *Other parameters*

Note – It is necessary to select the use of Session Capability Data functional unit in the parameter “Session requirements” but this parameter is set by RTSE and not visible within DTAM. It is the responsibility of the DTAM implementers to make sure that the “Session requirements” parameter in the A-ASSOCIATE request primitive will contain the appropriate value to select Session Capability Data FU.

7.2.1.1.3 *Use of the other RT-OPEN response and confirm primitive parameters*

7.2.1.1.3.1 *User Information*

This parameter only has relevance if the application-association is accepted by the RTSE service-provider.

For both the RT-OPEN response and confirmation primitives, the User Information parameter is used to carry the D-INITIATE-RESP APDU, whether the application-association is accepted or is rejected by the association-responding DTAM-PM.

7.2.1.1.3.2 *Result*

To be provided.

7.2.1.1.3.3 *Result source*

To be provided.

7.2.1.2 *Association-release procedure (mapping onto RT-CLOSE)*

Association-release procedure takes place concurrently with the underlying RT-CLOSE procedure. Only the initiating DTAM-PM may issue an RT-CLOSE request.

7.2.1.2.1 *Directly mapped parameters*

The following parameter of D-TERMINATE service primitives is mapped directly onto the corresponding parameters of the RT-CLOSE service primitives:

– User Data (on User Information).

7.2.1.2.2 *User of the other RT-CLOSE response and confirm primitive parameters*

No other parameter is used.

7.2.1.3 *Association-provider-abort procedure*

The use of the RT-P-ABORT indication primitive parameters are defined in Recommendation X.218.

7.2.1.4 *Association-user-abort procedure*

Association-user-abort procedure takes place concurrently with the underlying RTSE association-user-abort. All D-U-ABORT service parameters are directly mapped to the RT-U-ABORT service.

7.2.1.5 *Transfer procedure (mapping onto RT-TRANSFER)*

Transfer procedure takes place concurrently with the underlying RT-TRANSFER procedure. A DTAM-PM may issue an RT-TRANSFER request primitive only if it possesses the turn (token) and if there is no outstanding RT-TRANSFER confirm primitive.

7.2.1.5.1 *Use of the RT-TRANSFER service primitive parameters*

The following parameters of D-TRANSFER service primitives are mapped onto the corresponding parameters of the RT-TRANSFER service primitives.

7.2.1.5.1.1 *Document information*

The value of Document information shall be mapped onto the APDU parameter of RT-TRANSFER request primitive.

7.2.1.5.1.2 *Transfer-time*

The value of Transfer-time shall be directly mapped onto the Transfer-time parameter of RT-TRANSFER request primitive.

7.2.1.5.1.3 *Result*

The value of Result shall be directly mapped onto the Result parameter of RT-TRANSFER request primitive.

7.2.1.6 *Token Please procedure*

When a DTAM user issues a D-TOKEN-PLEASE service primitive, this results into an RT-TURN-PLEASE.

7.2.1.6.1 *Use of the RT-TURN-PLEASE request and indication primitive parameters*

This parameter takes a value which corresponds to requesting the data token.

7.2.1.6.1.1 *Priority*

The value of Priority shall be mapped onto the Priority parameter of RT-TURN-PLEASE service primitive.

7.2.1.7 *Control Give procedure*

When the DTAM-user issues a D-CONTROL-GIVE service primitive, this results into an RT-TURN-GIVE. This will transfer all the tokens from the requestor to the responder.

7.2.1.7.1 *Use of the RT-TURN-GIVE request and indication primitive parameters*

The RT-TURN-GIVE service primitives have no parameters. All the tokens are automatically passed to the other DTAM-PM.

7.2.2 *Mapping on the Presentation services*

7.2.2.1 *Capability data exchange procedure*

7.2.2.1.1 *Use of the P-CAPAB-DATA service parameters*

7.2.2.1.1.1 *User data*

The following DTAM APDUs are conveyed by this parameter:

a) D-CAPABILITY-REQ APDU;

b) D-CAPABILITY-RESP APDU.

7.3 *Mapping to the Session service (Recommendation X.215) in Transparent Mode*

This section defines how a DTAM-PM transfers APDUs by means of the Session service. Table 23/T.433 lists the overview of Session mapping.

7.3.1 *DTAM association-establishment procedure*

The association-establishment procedure takes place concurrently with the underlying establishment of Session connection.

7.3.1.1 *Directly mapped parameters*

No parameters of D-INITIATE service primitives are mapped directly onto the corresponding parameters of the S-CONNECT service primitives.

7.3.1.2 *Use of the other S-CONNECT request and indication primitive parameters*

7.3.1.2.1 *User Information*

For both the S-CONNECT request and indication primitives, the User Information parameter is used to carry to D-INITIATE-REQ APDU.

7.3.1.2.2 *Session requirements*

This parameter is set by the initiating DTAM-PM to select the following functional units by means of the "Telematic requirements parameter" in the D-INITIATE service primitive as shown in Table 24/T.433.

TABLE 23/T.433

DTAM service primitive/Protocol mapping to Session services

Functional units	Service primitive		Protocol elements (PDU)	Mapping DTAM PDU to Session service	
Association use control	D-INITIATE	req/ind	D-INITIATE-REQ PDU	S-CONNECT	req/ind
		rsp/cnf	D-INITIATE-RESP PDU	S-CONNECT	rsp/cnf
	D-TERMINATE	req/ind	None	S-RELEASE	req/ind
		rsp/cnf	None	S-RELEASE	rsp/cnf
	D-U-ABORT	req/ind	None	S-U-ABORT	req/ind
	D-P-ABORT	ind	None	S-P-ABORT	ind
Capability	D-CAPABILITY	req/ind	D-CAPABILITY-REQ PDU	S-CAPAB-DATA	req/ind
		rsp/cnf	D-CAPABILITY-RESP PDU	S-CAPAB-DATA	rsp/cnf
Document bulk transfer	D-TRANSFER	req	None	S-ACT-START/RESUME	req/ind
		ind	None	S-DATA	req/ind
				S-ACT-END/DCD/INT	req/ind
		cnf		S-ACT-END/DCD/INT	rsp/cnf
			None	S-MINOR-SYNC	req/ind
				S-MINOR-SYNC	rsp/cnf
			– User-exception-report	S-U-EXCEPTION-REPORT	req/ind
			– Provider-exception-report	S-P-EXCEPTION-REPORT	ind
Token control	D-CONTROL-GIVE	req/ind	None	S-CONTROL-GIVE	req/ind
	D-TOKEN-PLS	req/ind	None	S-TOKEN-PLS	req/ind

TABLE 24/T.433

Mapping into/out of the Session requirements

"Telematic requirements"	Functional units
Kernel	Kernel functional unit
Token control	Half-duplex functional unit
Capability	Capability data exchange functional unit
Bulk transfer	Minor synchronize functional unit Activity management functional unit Exceptions functional unit

7.3.1.2.3 *Session reference*

The initiating DTAM-PM will supply a Session Connection Identifier, which will be used to uniquely identify the session-connection. This identifier is formed of the following components:

a) Terminal Identifier of the calling terminal;

b) Date and Time;

c) Additional Session Reference Number (optionally).

The Terminal Identifier of the calling terminal is conveyed as the Calling SS-User Reference by the initiating DTAM-PM. Date and Time and Additional Session Reference Number are conveyed in parameters of the S-CONNECT primitive.

Each component, when present, will contain a data element of the appropriate type based on Recommendation F.200.

7.3.1.2.4 *Service Identifier*

The initiating DTAM-PM must supply a Service Identifier which has the value "1" to specify the Telematic services.

7.3.1.2.5 *Non-Basic Session Capabilities*

The initiating DTAM-PM may supply Non-Basic Session Capabilities, which will be used to specify the non-basic session capabilities available as receiving capabilities of the sender of this primitive. This parameter is formed of the following components:

a) Miscellaneous Session Capabilities;

b) Window Size

The requested checkpoint window parameter indicates, for each direction of transmission, the maximum number of checkpoints which may remain unacknowledged. Checkpoints are only inserted by the sender of a document. Values of this parameter may be the reason for subsequent termination. The continued progress of the service is only guaranteed if the entity acting as receiver gives acknowledgements within this limit. The window size is stated independently by each entity as the maximum value for when that entity is the receiving entity. There is no negotiation. The values for each direction of data transfer are not necessarily the same. The parameter value is an integer.

7.3.1.2.6 *Inactivity Timer*

The initiating DTAM-PM may use this parameter to negotiate the value of an Inactivity Timer. The default value is 60 seconds.

7.3.1.3 *Use of the other S-CONNECT response and confirm primitive parameters*

7.3.1.3.1 *User Information*

This parameter only has relevance if the application-association is accepted by the Session service-provider.

For both the S-CONNECT response and confirm primitives, the User Information parameter is used to carry the D-INITIATE-RESP APDU if the application-association is accepted or is rejected by the responding DTAM-PM.

7.3.1.3.2 *Result*

For the S-CONNECT response primitive the Result parameter is set by the association responding DTAM-PM as follows:

a) When the association-responding DTAM-PM detects errors in the S-CONNECT indication primitive (e.g., no Session User Data), the association-responding DTAM-PM rejects the application-association. The value of this parameter is set to "refuse".

b) If the association-responding DTAM-PM accepts the request, the value of this parameter is derived from the Result parameter of the D-INITIATE response primitive as shown in Table 25/T.433.

For the S-CONNECT confirm primitive the Result parameter is set by the association responding DTAM-PM as follows:

a) When the association-requesting (initiating) DTAM-PM receives the S-CONNECT confirm primitive with the Result parameter of "refuse", the association-requesting DTAM-PM issues the D-INITIATE confirm primitive with the Result parameter of "rejected by responding DTAM-PM" to the association-requesting DTAM user.

b) When the association-requesting DTAM-PM receives the S-CONNECT confirm primitive with the Result parameter of "accept", the association-requesting DTAM-PM issues the D-INITIATE confirm primitive with the Result parameter of "accepted" to the association-requesting DTAM user.

TABLE 25/T.433

Result parameter mapping

D-INITIATE response	S-CONNECT response/confirm	D-INITIATE confirm
Accepted	Accept	Accepted
Rejected by responder with some reasons	Refuse	Rejected by responding DTAM-PM
Rejected by responding DTAM-PM	Refuse	Rejected by responding DTAM-PM

7.3.1.3.3 *Session requirements*

This parameter has the same values as in the S-CONNECT request and indication primitives.

7.3.1.3.4 *Session reference*

This parameter has the same values as in the S-CONNECT indication primitives. The Terminal Identifier of the calling terminal of the S-CONNECT indication primitive is returned as the Terminal Identifier of the called terminal by the responding DTAM-PM.

7.3.1.3.5 *Service identifier*

This parameter has the same values as in the S-CONNECT request and indication primitives.

7.3.1.3.6 *Non-Basic Session Capabilities*

The responding DTAM-PM may supply Non-Basic Session Capabilities, which will be used to specify the non-basic session capabilities available as receiving capabilities of the sender of this primitive. This parameter is formed of the same components as those in S-CONNECT request and indication primitives.

7.3.1.3.7 *Inactivity Timer*

The responding DTAM-PM may use this parameter to negotiate the value of an Inactivity Timer.

7.3.2 *Association release procedure*

The association release procedure takes place concurrently with the underlying release of Session connection.

There are no D-TERMINATE service parameters to map onto Session connection release service parameters.

7.3.3 *Association-provider-abort procedure*

The uses of the S-P-ABORT indication primitive parameters are defined in Recommendation X.215.

7.3.4 *Association-user-abort procedure*

Association-user-abort procedure takes place concurrently with the underlying Session abort. All D-U-ABORT service parameters are directly mapped to the S-U-ABORT service.

7.3.5 *Transfer procedure*

7.3.5.1 *Use of the S-ACTIVITY-START request and indication primitive parameters*

7.3.5.1.1 *Document Reference Number (Activity Identifier)*

The requesting DTAM-PM must allocate and supply the next Document Reference Number (Activity Identifier) for the current session.

The DTAM-PMs should manage the mapping between the parameter “Document Reference Information” in D-TRANSFER service and the parameter “Document Reference Number (Activity Identifier)” in the S-ACTIVITY-START request and indication primitives.

7.3.5.1.2 *Document type identifier*

This parameter is user option and a detailed use of this parameter is for further study.

7.3.5.1.3 *Service interworking identifier*

This parameter is user option and a detailed use of this parameter is for further study.

7.3.5.1.4 *User Data*

This parameter is only used to invoke the DTAM capability. The information, which is generated by the DTAM-PM based on the parameter of “Document Characteristics” in Document Profile contained in the Document Information, is conveyed as shown in Figure 3/T.433.

7.3.5.2 *Use of the S-DATA request and indication primitive parameters*

The document information is divided into segments such that the segment boundaries coincide with the minor synchronization points. Each segment consists of an integral number of interchange-data-elements. The interchange-data-elements of each segment are encoded using the Basic Encoding Rules defined in Recommendation X.209. The encoded interchange-data-elements of each segment are concatenated, forming an encoded segment.

```
S-ACTIVITY-START-user-data      ::= CHOICE
                    {   [4]  IMPLICIT DocumentCharacteristics          }

DocumentCharacteristics       ::=SET {
  DocumentApplicationProfile      ::= CHOICE {
                                    [0] IMPLICIT OCTET STRING
                                    -- '01'H Non-DocumentApplicationProfile
                                    -- '02'H DocumentApplicationProfile T.503
                                    [4] IMPLICIT SET OF OBJECT IDENTIFIER }

DocumentArchitectureClass         [1] IMPLICIT OCTET STRING OPTIONAL,
                                    -- '00'H means FDA

nonBasicDocumentCharacteristics [2]     IMPLICIT NonBasicDocumentCharacteristics
                                                          OPTIONAL,

nonBasicStructuralCharacteristics [3]   IMPLICIT NonBasicStructuralCharacteristics
                                                          OPTIONAL   }

NonBasicDocumentCharacteristics     ::= SET {
  commentsCharacterSets [1]       IMPLICIT OCTET STRING OPTIONAL,
                                      -- string of escape sequences
  pageDimensions                  [2]  IMPLICIT SET OF Dimension-Pair OPTIONAL,
  ra-gr-coding-attributes         [3]  IMPLICIT SET OF Ra-Gr-Coding-Attribute
                                       OPTIONAL,
                                       -- Ra-Gr-Coding-Attribute is defined in
                                       -- Recommendation T.415
  ra-gr-presentation-features     [4]  IMPLICIT SET OF
                                           Ra-Gr-Presentation-Features OPTIONAL
                                       -- Ra-Gr-Presentation-Features is defined in
                                       -- Recommendation T.415          }

NonBasicStructuralCharacteristics       ::= SET {
  numberOfObjectsPerPage          [0]  IMPLICIT INTEGER OPTIONAL   }
```

FIGURE 3/T.433

User data in S-ACTIVITY-START/RESUME

7.3.5.2.1 *User Data*

A segment of interchange-data-elements is conveyed by the User Data.

Note – Some DTAM-PMs may take one of the following actions:

– when sending the document information, the requesting DTAM-PM may suppress the Document Profile located at the top of the document information;

– when receiving the document information, the responding DTAM-PM may regenerate the Document Profile and may attach it to the top of the Document Information based on the User Data of S-ACTIVITY-START indication.

7.3.5.3 *Use of the S-MINOR-SYNCHRONIZE service parameters*

7.3.5.3.1 *Type*

The DTAM-PM uses only the “explicit confirmation expected” type of minor synchronization.

7.3.5.3.2 *Synchronization point serial number (Checkpoint reference number)*

The session service-provider allocates checkpoint serial numbers and passes them to the requesting and the responding DTAM-PMs to associate with the transmitted data.

7.3.5.4 *Use of the S-ACTIVITY-END service parameters*

7.3.5.4.1 *Synchronization point serial number (Checkpoint reference number)*

The serial number of the implied major synchronization point is allocated by the session service-provider and passed up to both DTAM-PMs. This parameter will be mapped into/out of the parameter "Synchronization Point" in D-TRANSFER service.

7.3.6 *Token Please procedure*

When the DTAM-user issues a D-TOKEN-PLEASE service primitive, this results into an S-TOKEN-PLEASE.

7.3.6.1 *Use of the S-TOKEN-PLEASE request and indication primitive parameters*

7.3.6.1.1 *Tokens*

The responding DTAM-PM (receiver of document) will only request the data token.

7.3.7 *Control Give procedure*

When the DTAM-user issues a D-CONTROL-GIVE service primitive, this results into an S-CONTROL-GIVE. This will transfer all the tokens from the requestor to the responder.

7.3.7.1 *Use of the S-CONTROL-GIVE request and indication primitive parameters*

The S-CONTROL-GIVE service primitives have no parameters.

7.3.8 *Capability data exchange procedure*

7.3.8.1 *Use of the S-CAPAB-DATA service parameters*

7.3.8.1.1 *User Data*

The following DTAM APDUs are conveyed by this parameter:

a) D-CAPABILITY-REQ APDU;

b) D-CAPABILITY-RESP APDU.

7.3.8.1.2 *Inactivity Timer*

The initiating/responding DTAM-PMs may use this parameter to negotiate the value of an Inactivity Timer.

7.3.8.1.3 *Storage capacity*

The initiating/responding DTAM-PMs may supply a storage capacity to negotiate the memory size for the communication. However, for some applications under a Transport Mode, this parameter is used by the sending DTAM-PM to indicate a "required storage capacity" to the peer machine. The receiving DTAM-PM uses this parameter to respond whether it is able to provide this storage capacity or not, so as to maintain compatibility with the old implementation based on Recommendation T.73.

7.3.9 *User-exception-report procedure*

7.3.9.1 *Use of the S-U-EXCEPTION-REPORT service parameters*

7.3.9.1.1 *Reason*

This parameter may specify one of the following reasons:

a) non-specific error;

b) temporarily unable to enter into, or to continue a session (receiving ability jeopardized);

c) sequence error;

d) unrecoverable procedure error;

e) local SS-User error.

7.3.10 *Provider-exception-report procedure*

7.3.10.1 *Use of the S-P-EXCEPTION-REPORT service parameters*

7.3.10.1.1 *Reason*

This parameter may specify the following reasons:

– protocol error.

7.3.11 *Transfer-interrupt procedure*

7.3.11.1 *Use of the S-ACTIVITY-INTERRUPT service parameters*

7.3.11.1.1 *Reason*

This parameter may specify one of the reasons as described in Reason for S-U-EXCEPTION-REPORT service primitive.

7.3.12 *Transfer-discard procedure*

7.3.12.1 *Use of the S-ACTIVITY-DISCARD service parameters*

7.3.12.1.1 *Reason*

This parameter may specify one of the reasons as described in Reason for S-U-EXCEPTION-REPORT service primitive.

7.3.13 *Transfer-user-resumption procedure*

7.3.13.1 *Use of the S-ACTIVITY-RESUME service parameters*

7.3.13.1.1 *Document Reference Number (Activity Identifier)*

The requesting DTAM-PM must allocate and supply the next Document Reference Number (Activity Identifier) for the current session.

7.3.13.1.2 *Old Document Reference Number*

The requesting DTAM-PM must supply the Old Document Reference Number (the original Activity Identifier) assigned to the previously interrupted activity in the S-ACTIVITY-START request primitive.

7.3.13.1.3 *Checkpoint point serial number*

The requesting DTAM-PM will specify the Serial Number of the last confirmed checkpoint in the interrupted activity. The session service-provider will also set the current session serial number to this value. If there was no previously confirmed checkpoint, the activity cannot be continued. The requesting DTAM-PM must then send an S-ACTIVITY-RESUME request primitive (with the Synchronization Point Serial Number set to zero), followed by an S-ACTIVITY-DISCARD request primitive.

This parameter will be mapped into/out of the parameter “Synchronization Point” in D-TRANSFER service.

7.3.13.1.4 *Old Session Reference*

The requesting DTAM-PM must supply the Session Reference of the session-connection during which the Activity was started. The Session Reference of the previous session-connection is conveyed in the Calling and Called Terminal Identifier, Common Reference and optionally, Additional Reference Information components of this parameter.

7.3.13.1.5 *Document type identifier*

This parameter is user option and a detailed use of this parameter is for further study.

7.3.13.1.6 *Service interworking identifier*

This parameter is user option and a detailed use of this parameter is for further study.

7.3.13.1.7 *User Data*

This parameter has the same format of User Data as for the S-ACTIVITY-START service parameters.

8 Abstract syntax definition of APDUs

8.1 *Abstract syntax definition of APDUs in Normal Mode*

This abstract syntax is described by the notation of ASN.1 defined in Recommendation X.208.

```
DTAM-APDU { ccitt recommendation tseries(20) dtam(433) apdus(0) }
DEFINITIONS ::=
BEGIN
-- PROLOGUE --
-- EXPORTS everything --

IMPORTS
-- ODA --

DocumentProfileDescriptor, LayoutClassDescriptor, LayoutObjectDescriptor, TextUnit,
LogicalClassDescriptor, LogicalObjectDescriptor, LayoutStyleDescriptor, ODADocumentApplicationProfile,
NonBasicODADocumentCharacteristics, NonBasicODAStructuralCharacteristics

        FROM Interchange-Data-Elements { joint-iso-ccitt 8 1 5 5 }

-- OPERATIONAL STRUCTURE --

OperationalDescriptor, OperationalInformationIdentifier

-- FROM Recommendations T.441 and T.541 --

dTAM APPLICATION-SERVICE-ELEMENT ::= { ccitt recommendation tseries(20)
                                       dtam(433) aseID(1) }
-- APDU DEFINITIONS --

[1]  D-INITIATE-REQ ::= [APPLICATION 10] IMPLICIT SEQUENCE

    {   serviceClasses              [0] IMPLICIT INTEGER OPTIONAL,
                                        -- the use of this parameter
                                        -- is for further study.

        telematicRequirements       [1] IMPLICIT BITSTRING
                                        {  kernel                          (0),
                                           capabilityManagement            (1),
                                           documentBulkTransfer (2),
                                           typedDataTransmission           (3),
                                           documentUnconfirmedManipulation (4),
                                           documentConfirmedManipulation   (5),
                                           remoteDocumentAccess            (6),
                                           remoteDocumentManagement        (7),
                                           tokenControl                    (8),
                                           exceptionReport                 (9)   },
        applicationCapabilities     [2] IMPLICIT SET OF Application Capabilities,
        protocolVersion             [3] IMPLICIT BITSTRING
                                                {          version-1  (0) }  OPTIONAL,
        dTAMQOS                     [4] IMPLICIT BITSTRING                   OPTIONAL,
        account                     [5] IMPLICIT Account                     OPTIONAL,
        userInformation             [8] OCTETSTRING                          OPTIONAL
        }
    --  Registration of OBJECT IDENTIFIER for Recommendation T.503 is required.
    --  dTAMQOS and account parameters are for further study.
```

[2] D-INITIATE-RESP ::= [APPLICATION 11] IMPLICIT SEQUENCE

```
{   telematicRequirements          [0] IMPLICIT BITSTRING OPTIONAL,
    applicationCapabilities        [1] IMPLICIT SET OF Application Capabilities OPTIONAL,
    protocolVersion                [2] IMPLICIT BITSTRING
                                       {  version-1  (0)}  OPTIONAL,
    dTAMQOS                        [3] IMPLICIT BITSTRING  OPTIONAL,
    result                         [4] INTEGER
        {   accepted                                                  (0),
            rejected by responder(reason-not-specified)               (1),
            rejected by responder(protocolVersion-not-supported)      (2),
            rejected by responder(DTAMQOS-not-supported)              (3),
            rejected by responder(application-context-not-supported)  (4),
            rejected by responding DTAM-PM                            (5) },
        userInformation            [7] OCTETSTRING                    OPTIONAL
}
```

[3] D-TERMINATE-REQ ::= [APPLICATION 14] IMPLICIT SEQUENCE

```
{   userInformation                [0] OCTETSTRING                    OPTIONAL
    }
```

[4] D-TERMINATE-RESP ::= [APPLICATION 15] IMPLICIT SEQUENCE

```
{   charging                       [0] IMPLICIT Charging              OPTIONAL,
    userInformation                [1] OCTETSTRING                    OPTIONAL
```

[5] D-ABORT-REQ ::= [APPLICATION 13] IMPLICIT SEQUENCE

```
{   aBORTSource                    [0] INTEGER
                                   {   requestor                     (0),
                                       DTAMserviceProvider           (1)   },
    aBORTReason                    [1] INTEGER
                                   {   local-system-problem          (0),
                                       invalid-parameter             (1),
                                       unrecognized-activity         (2),
                                       temporary-problem             (3),
                                       protocol-error                (4),
                                       permanent-error               (5),
                                       transfer-completed            (6)   },

    Reflected-parameter            [2] IMPLICIT BITSTRING             OPTIONAL,
                                       -- 8 bits maximum, only if abortReason
                                       -- is invalid Parameter

    userInformation                [3] OCTETSTRING                    OPTIONAL

    }
```

Note – Not used in the DTAM bulk transfer. In the DTAM bulk transfer (Normal Mode) all D-ABORT service parameters are directly mapped to the RT-ABORT service.

[6] D-CAPABILITY-REQ ::= [23] IMPLICIT SEQUENCE

```
{   applicationCapabilities        [0] Application Capabilities       OPTIONAL,
    userInformation                [2] OCTETSTRING                    OPTIONAL
    }
```

[7] D-CAPABILITY-RESP ::= [24] IMPLICIT SEQUENCE

```
{   applicationCapabilities        [0] Application Capabilities       OPTIONAL,
    capabilityResult               [2] IMPLICIT Capability Result,
    userInformation                [3] OCTETSTRING                    OPTIONAL
    }
```

CapabilityResult ::= INTEGER

```
{   confirmation-of-all-the-requestedCapabilites                (0),
    a-list-of-the-requestedCapabilities                         (1),
    a-complete-list-of-non-basicReceivingCapabilities           (2),
    none-of-the-capabilities-requested-by-the-initiator         (3)  }
```

[8] D-TYPED-DATA ::= [APPLICATION 16] CHOICE

```
{   NumericString,
    PrintableString,
    TeletexString,
    VideotexString,
    VisibleString,
    OctetString,
    IA5String,
    GraphicString       }
```

[9] D-CREATE ::= [APPLICATION 17] IMPLICIT SEQUENCE OF CreateInformation

CreateInformation ::= SEQUENCE

```
{   [0]  IMPLICIT ParentObjectOrClassIdentifier OPTIONAL,
    [1]  Object         }
```

ParentObjectOrClassIdentifier ::= ObjectOrClassIdentifier

-- ObjectOrClassIdentifier ::= { Refer to Recommendation T.415 }

Object ::= CHOICE

```
{   [0]  IMPLICIT DocumentProfileDescriptor,
    [1]  IMPLICIT LayoutClassDescriptor,
    [2]  IMPLICIT LayoutObjectDescriptor,
    [3]  IMPLICIT TextUnit,
    [5]  IMPLICIT LogicalClassDescriptor,
    [6]  IMPLICIT LogicalObjectDescriptor,
    [7]  IMPLICIT PresentationStyleDescriptor,
    [8]  IMPLICIT LayoutStyleDescriptor,
         -- The above descriptors and text unit are defined
         -- in Recommendation T.415
    [9]  IMPLICIT OperationalDescriptor
         -- The above descriptors are defined
         -- in Recommendation T.441 and details are for further
         -- study }
```

[10] D-DELETE ::= [APPLICATION 18] IMPLICIT DeleteInformation

DeleteInformation ::= SEQUENCE OF CHOICE

```
{   [0]  IMPLICIT ObjectOrClassIdentifier,
    [1]  IMPLICIT ContentPortionIdentifier,
         -- The above Identifiers are defined
         -- in Recommendation T.415
    [2]  IMPLICIT OperationalInformationIdentifier
         -- The above Identifiers are defined
         -- in Recommendation T.441 and details are for
         -- further study }
```

[11] D-MODIFY ::= [APPLICATION 19] IMPLICIT SEQUENCE OF ModifyInformation

ModifyInformation ::= SEQUENCE

```
{   [0]  IMPLICIT CurrentObjectOrClassIdentifier     OPTIONAL,
    [1]  Object         }
```

CurrentObjectOrClassIdentifier ::= ObjectOrClassIdentifier

-- ObjectOrClassIdentifier ::= { Refer to Recommendation T.415 }

```
Object ::= CHOICE

        {   [0]   IMPLICIT DocumentProfileDescriptor,
            [1]   IMPLICIT LayoutClassDescriptor,
            [2]   IMPLICIT LayoutObjectDescriptor,
            [3]   IMPLICIT TextUnit,
            [5]   IMPLICIT LogicalClassDescriptor,
            [6]   IMPLICIT LogicalObjectDescriptor,
            [7]   IMPLICIT PresentationStyleDescriptor,
            [8]   IMPLICIT LayoutStyleDescriptor,
                  -- The above descriptors and text unit are defined
                  -- in Recommendation T.415
            [9]   IMPLICIT OperationalDescriptorDescriptor,
                  -- The above descriptors are defined
                  -- in Recommendation T.441 and details are for
                  -- further study }

[12]  D-CALL ::= [APPLICATION 20] IMPLICIT CALLInformation

CALLInformation ::= SEQUENCE OF CHOICE

        {   [0]   IMPLICIT OperationalInformationIdentifier
                  -- The above identifiers are defined in
                  -- Recommendations T.441 and T.541 }

[13]  D-REBUILD ::= [APPLICATION 21] IMPLICIT REBUILDInformation
[ for further study ]

[14]  D-TOKEN-PLEASE ::= [APPLICATION 22] IMPLICIT Priority
```

Note – Not used in the DTAM bulk transfer. In the case of DTAM bulk transfer (Normal Mode) the D-TOKEN-PLEASE service parameter is directly mapped to the RT-TURN-PLEASE service.

```
Priority ::= INTEGER

ApplicationCapabilities                 ::= CHOICE {

oDAApplicationCapabilities              [0] IMPLICIT SET {

oDADocumentApplicationProfile  [0]      IMPLICIT OBJECT IDENTIFIER,

nonBasicODADocumentCharacteristics      [1]                 IMPLICIT
                                        NonBasicODADocumentCharacteristics
                                        OPTIONAL,

nonBasicODAStructuralCharacteristics    [2]                 IMPLICIT
                                        NonBasicODAStructuralCharacteristics
                                        OPTIONAL,

operationalApplicationProfile           [3] IMPLICIT SET OF OBJECT IDENTIFIER
                                        OPTIONAL },

filetransferCapabilities                [1] IMPLICIT INTEGER    {
                                        bftCapabilities                 (0),
                                        transparentDataCapability (1)   }

                                        }

END -- of DTAM Protocol in Normal Mode
```

8.2 *Abstract syntax definition of APDUs for use of the session service in Transparent Mode*

This abstract syntax is described by the notation of ASN.1 defined in the Recommendation X.208.

```
[1]  D-INITIATE-REQ ::= CHOICE

        {   [4]   IMPLICIT ApplicationCapabilities      }

    ApplicationCapabilities             ::= SET {
        documentApplicationProfileT73   [0] IMPLICIT OCTET STRING OPTIONAL,
                                        -- '02'H document application profile (T.503)
        documentArchitectureClass       [1] IMPLICIT OCTET STRING OPTIONAL
                                        -- '00'H means FDA }
```

[2] D-INITIATE-RESP ::= CHOICE

```
{       [4]     IMPLICIT ApplicationCapabilities        }

ApplicationCapabilities                 ::= SET {
    documentApplicationProfileT73   [0] IMPLICIT OCTET STRING OPTIONAL,
                                        -- '02'H document application profile (T.503)
    documentArchitectureClass       [1] IMPLICIT OCTET STRING OPTIONAL
                                        -- '00'H means FDA }
```

[3] D-CAPABILITY-REQ ::= CHOICE

```
{       [4]     IMPLICIT ApplicationCapabilities        }

ApplicationCapabilities                 ::= SET {
    documentApplicationProfileT73   [0] IMPLICIT OCTET STRING OPTIONAL,
    documentArchitectureClass       [1] IMPLICIT OCTET STRING OPTIONAL,
    nonBasicDocCharacteristics      [2] IMPLICIT NonBasicDocCharacteristics OPTIONAL,
    nonBasicStrucCharacteristics    [3] IMPLICIT NonBasicStrucCharacteristics OPTIONAL    }

        -- "NonBasicDocCharacteristics" and "NonBasicStrucCharacteristics" are defined
        -- in Recommendation T.415.
```

[4] D-CAPABILITY-RESP ::= CHOICE

```
{       [4]     IMPLICIT ApplicationCapabilities        }

ApplicationCapabilities                 ::= SET {
    documentApplicationProfileT73   [0] IMPLICIT OCTET STRING OPTIONAL,
    documentArchitectureClass       [1] IMPLICIT OCTET STRING OPTIONAL,
    nonBasicDocCharacteristics      [2] IMPLICIT NonBasicDocCharacteristics OPTIONAL,
    nonBasicStrucCharacteristics    [3] IMPLICIT NonBasicStrucCharacteristics OPTIONAL }
```

9 Conformance

For further study.

ANNEX A

(to Recommendacion T.433)

Examples of protocol sequence for document bulk transfer

Figures A-1/T.433 to A-3/T.433 show the examples of protocol sequence for the document bulk transfer.

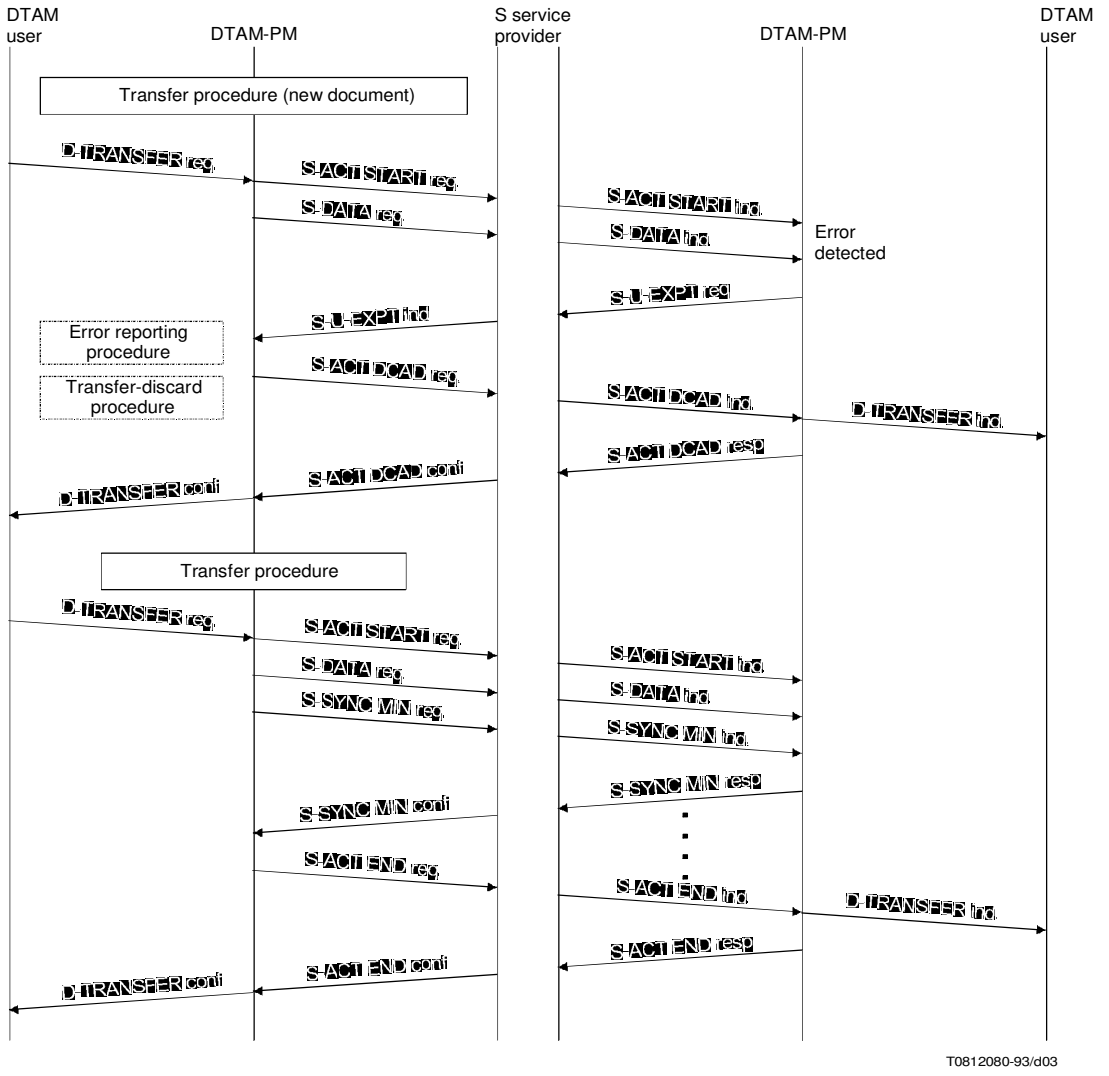


FIGURE A-1/T.433
Transfer procedure (transfer-discard procedure)
(Transparent mode)

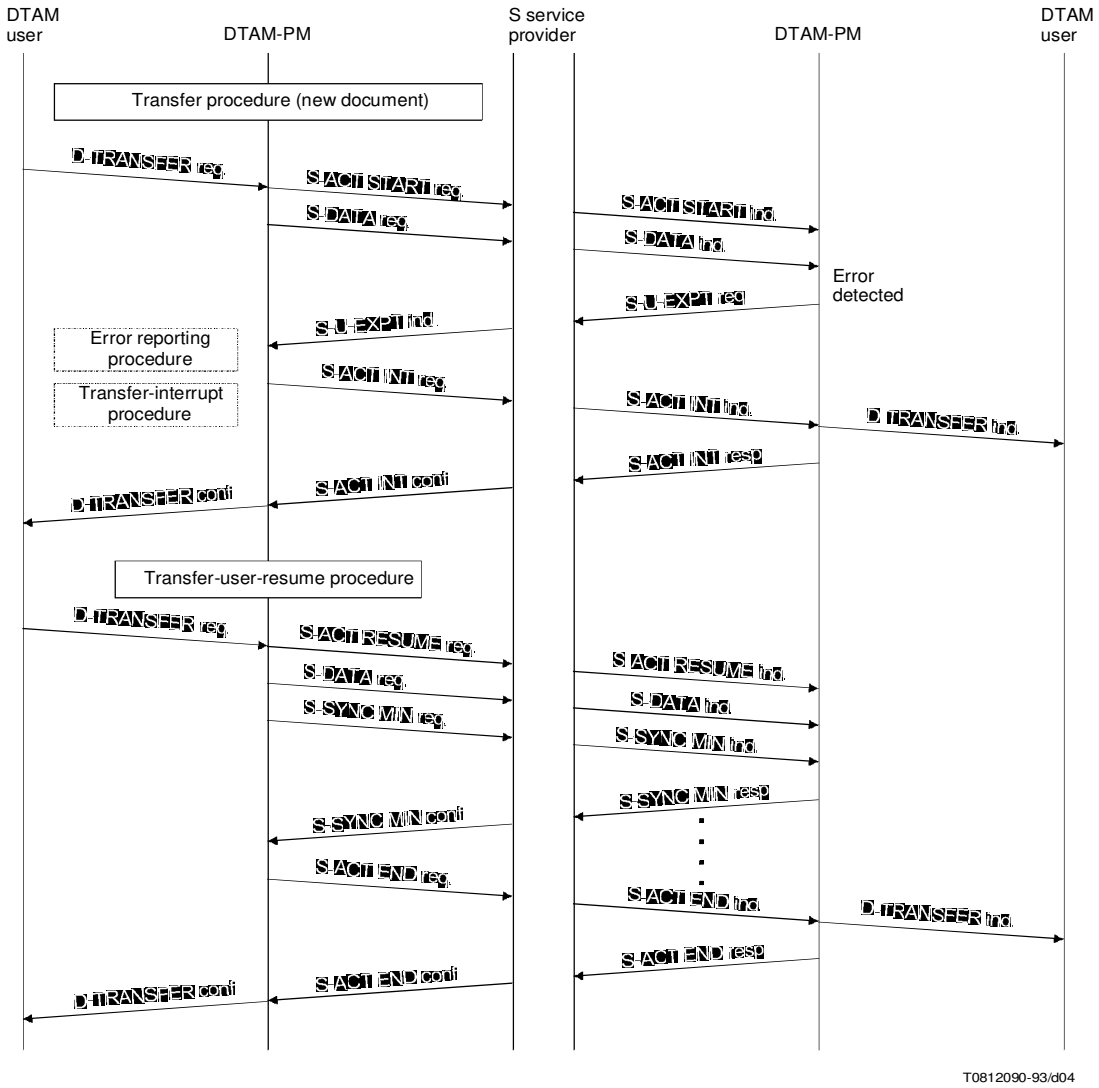


FIGURE A-2/T.433

Transfer procedure (transfer-interrupt procedure) and transfer-user-resume procedure
(Transparent mode)

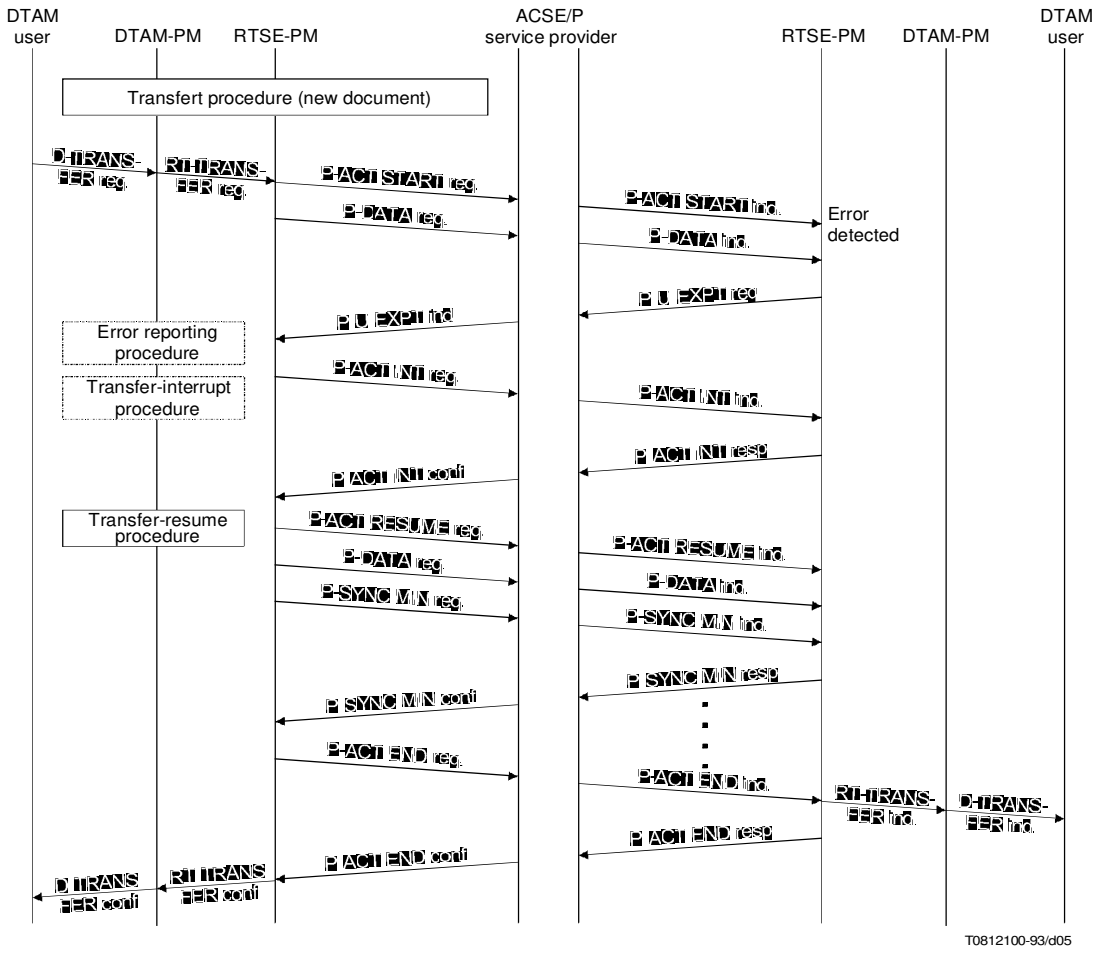


FIGURE A-3/T.433

Transfer procedure when RTSE is used

(Normal mode)

ANNEX B

(to Recommendation T.433)

DTAM-PM state tables

(Document bulk transfer/Transparent mode)

B.1 *General*

The DTAM-PM state table described in this annex is applicable to Document Bulk Transfer in Transparent Mode.

This annex defines a single DTAM Protocol Machine (DTAM-PM) in terms of a state table. The state table shows the interrelationship between the state of an application-association, the incoming events that occur in the protocol, the actions taken, and finally, the resultant state of the application-association.

The DTAM-PM state table does not constitute a formal definition of a DTAM-PM. It is included to provide a more precise specification of the elements of procedure defined in § 6.

This annex contains the following tables:

a) Table B-1/T.433 specifies the abbreviated name, source, and name/description of each incoming event. The sources are:

   1) DTAM-SE-user (DTAM-SE-user);

   2) peer DTAM-PM (DTAM-PM-peer);

   3) session service provider (SS-provider);

   4) DTAM-PM (DTAM-PM).

b) Table B-2/T.433 specifies the abbreviated name of each state of the DTAM-PM.

c) Table B-3/T.433 specifies the abbreviated name, target, and name/description of each outgoing event. The targets are:

   1) DTAM-SE-user (DTAM-SE-user);

   2) peer DTAM-PM (DTAM-PM-peer);

   3) session service provider (SS-provider);

   4) DTAM-PM (DTAM-PM).

d) Table B-4/T.433 specifies the predicates;

e) Table B-5/T.433 specifies the specific actions;

f) Tables B-6/T.433 to B-11/T.433 inclusive specify the DTAM-PM state table using the abbreviations of the above tables.

For some events the source and the target is the DTAM-PM (internal event). If the DTAM-PM issues an internal event as part of an action taken, the DTAM-PM awaits that internal event in the resultant state.

TABLE B-1/T.433

Incoming events list

Abbreviated name	Source	Name and description
D-CAPreq	DTAM-SE-user	D-CAPABILITY request primitive
D-CAPres	DTAM-SE-user	D-CAPABILITY response primitive
D-INTreq	DTAM-SE-user	D-INITIATE request primitive
D-INTres+	DTAM-SE-user	D-INITIATE response primitive (accepted)
D-INTres–	DTAM-SE-user	D-INITIATE response primitive (rejected)
D-TERreq	DTAM-SE-user	D-TERMINATE request primitive
D-TERres	DTAM-SE-user	D-TERMINATE response primitive
D-TRreq	DTAM-SE-user	D-TRANSFER request primitive
D-TRreq	DTAM-SE-user	D-TRANSFER request primitive (resume)
D-TPreq	DTAM-SE-user	D-TOKEN-PLEASE request primitive
D-CGreq	DTAM-SE-user	D-CONTROL-GIVE request primitive
D-UAreq	DTAM-SE-user	D-U-ABORT request primitive
DINQ	Session-provider	D-INITIATE-REQ APDU as user data of an S-CONNECT indication
DINR+	Session-provider	D-INITIATE-RESP APDU as user data of an S-CONNECT confirm or an A-ASSOCIATE confirm primitive (accepted)
S-CONcnf	Session-provider	S-CONNECT confirm primitive (rejected)
SEG	Session-provider	Segment of document information
S-CAPind	Session-provider	S-CAPABILITY-DATA indication primitive
S-CAPcnf	Session-provider	S-CAPABILITY-DATA confirm primitive
S-RELind	Session-provider	S-RELEASE indication primitive
S-RELcnf	Session-provider	S-RELEASE comfirm primitive
S-ASind	Session-provider	S-ACTIVITY-START indication primitive
S-ARind	Session-provider	S-ACTIVITY-RESUME indication primitive
S-AEind	Session-provider	S-ACTIVITY-END indication primitive
S-AEcnf	Session-provider	S-ACTIVITY-END confirm primitive
S-Alind	Session-provider	S-ACTIVITY-INTERRUPT indication primitive
S-Alcnf	Session-provider	S-ACTIVITY-INTERRUPT confirm primitive
S-ADind	Session-provider	S-ACTIVITY-DISCARD indication primitive
S-ADcnf	Session-provider	S-ACTIVITY-DISCARD confirm primitive
S-TPind	Session-provider	S-TOKEN-PLEASE indication primitive
S-CGind	Session-provider	S-CONTROL-GIVE indication primitive
S-MSind	Session-provider	S-MINOR-SYNC indication primitive
S-MScnf	Session-provider	S-MINOR-SYNC confirm primitive
S-UEind	Session-provider	U-EXCEPTION-REPORT indication primitive
S-PEind	Session-provider	P-EXCEPTION-REPORT indication primitive
S-UABind	Session-provider	U-ABORT indication primitive
S-PABind	Session-provider	P-ABORT indication primitive
next	DTAM-PM	Transfer of the next segment
tr-discard	DTAM-PM	Start of transfer-discard procedure
tr-interr	DTAM-PM	Start of transfer-interrupt procedure
tr-p-ab	DTAM-PM	Start of procedures transfer-abort followed by Provider-abort
transfer	DTAM-PM	Start of transfer procedure
resume	DTAM-PM	Start of resume procedure

TABLE B-2/T.433

DTAM-PM states

Abbreviated name	Name and description
STA0	Idle: unassociated
STA01	Awaiting DINR
STA02	Awaiting D-INTres+ or D-INTres–
STA11	Associated: sending DTAM-PM
STA111	Associated: awaiting S-CAPcnf
STA22	Associated: receivingDTAM-PM
STA221	Associated: awaiting D-CAPres
STA30	Transfer: sending DTAM-PM
STA31	Suspended transfer: sending DTAM-PM
STA32	Awaiting S-AEcnf: DTAM-PM
STA34*	Awaiting start of discard or interrupt procedure
STA341	Awaiting S-ADcnf or S-Alcnf (discard/interrupt procedure)
STA40	Awaiting SEG: receiving DTAM-PM
STA41	Awaiting S-MSind or S-AEind: receiving DTAM-PM
STA70*	Awaiting start of abort procedure (local abort)
STA71*	Awaiting start of abort procedure (remote abort)
STA91	Awaiting D-TERres
STA92	Awaiting S-RELcnf

TABLE B-3/T.433

Outgoing events list

Abbreviated name	Target	Name and description
D-CAPind	DTAM-SE user	D-CAPABILITY indication primitive
D-CAPcnf	DTAM-SE user	D-CAPABILITY confirm primitive
D-INTind	DTAM-SE user	D-INITIATE indication primitive
D-INTcnf+	DTAM-SE user	D-INITIATE confirm primitive (accepted)
D-INTcnf-	DTAM-SE user	D-INITIATE confirm primitive (rejected)
D-TERind	DTAM-SE user	D-TERMINATE indication primitive
D-TERcnf	DTAM-SE user	D-TERMINATE confirm primitive
D-TRind	DTAM-SE user	D-TRANSFER indication primitive
D-TRcnf+	DTAM-SE user	D-TRANSFER confirm primitive (transferred)
D-TRcnf–	DTAM-SE user	D-TRANSFER confirm primitive (not transferred)
D-TPind	DTAM-SE user	D-TOKEN-PLEASE indication primitive
D-CGind	DTAM-SE user	D-CONTROL-GIVE indication primitive
D-UAind	DTAM-SE user	D-U-ABORT indication primitive
D-PAind	DTAM-SE user	D-P-ABORT indication primitive
DINQ	Session-provider	D-INITIATE-REQ APDU as user data of an S-CONNECT request
DINR+	Session-provider	D-INITIATE-RESP APDU as user data of an S-CONNECT response primitive (accepted)
S-CONresp–	Session-provider	S-CONNECT response primitive (rejected)
SEG	Session-provider	Segment of document information
S-CAPreq	Session-provider	S-CAPABILITY-DATA request primitive
S-CAPres	Session-provider	S-CAPABILITY-DATA response primitive
S-RELreq	Session-provider	S-RELEASE request primitive
S-RELres	Session-provider	S-RELEASE response primitive
S-ASreq	Session-provider	S-ACTIVITY-START request primitive
S-ARreq	Session-provider	S-ACTIVITY-RESUME request primitive
S-AEreq	Session-provider	S-ACTIVITY-END request primitive
S-AEres	Session-provider	S-ACTIVITY-END response primitive
S-Alreq	Session-provider	S-ACTIVITY-INTERRUPT request primitive
S-Alres	Session-provider	S-ACTIVITY-INTERRUPT response primitive
S-ADreq	Session-provider	S-ACTIVITY-DISCARD request primitive
S-ADres	Session-provider	S-ACTIVITY-DISCARD response primitive
S-TPreq	Session-provider	S-TOKEN-PLEASE request primitive
S-CGreq	Session-provider	S-CONTROL-GIVE request primitive
S-MSreq	Session-provider	S-MINOR-SYNC request primitive
S-MSres	Session-provider	S-MINOR-SYNC response primitive
S-UEreq	Session-provider	U-EXCEPTION-REPORT request primitive
S-UABreq	Session-provider	U-ABORT request primitive
next	DTAM-PM	Transfer of the next segment
tr-discard	DTAM-PM	Start of transfer-discard procedure
tr-interr	DTAM-PM	Start of transfer-interrupt procedure
tr-p-ab	DTAM-PM	Start of procedures transfer-abort followed by Provider-abort
transfer	DTAM-PM	Start of transfer procedure
resume	DTAM-PM	Start of resume procedure

TABLE B-4/T.433

Predicates

Code	Name and description
p1	DTAM-PM can support the requested application-association (connection)
p11	Association-initiating DTAM-PM = TRUE
p31	Segment is the last one necessary to transfer the complete APDU
p32	Outstanding-minor-syncs < window size
p33	Outstanding-minor-syncs = 0
p35	Checkpoint confirmed = TRUE (at least one S-Mscnf received)
p36	Local choice
p361	Reason parameter value of S-UEind is “receiving ability jeopardized”
p37	Minor sync = last acknowledged minor sync + 1
p41	Received segment (SEG) has been secured
p42	Complete DTAM-SE-user APDU has been secured

TABLE B-5/T.433

Actions

Code	Name and description
a1	Association-initiating DTAM-PM = TRUE
a2	Association-initiating DTAM-PM = FALSE
a10	Store the segment received
a30	Outstanding-minor-syncs = 0 Checkpoint-confirmed = FALSE
a31	Outstanding-minor-syncs = outstanding minor syncs + 1
a32	Outstanding-minor-syncs = outstanding minor syncs – 1 Checkpoint-confirmed = TRUE

TABLE B-6/T.433

DTAM-PM state table – Association/connection establishment

	STA0	STA01	STA02
D-INTreq	p1 DINQ [a1] STA01		
DINQ	p1 D-INTind [a2] STA02 ¬ p1 S-CONcnf– STA0		
D-INTres+			DINR+ STA22
D-INTres–			S-CONcnf– STA0
DINR+		D-INTcnf+ STA11	
S-CONcnf		D-INTcnf– STA0	
S-PABind		D-PAind STA0	D-PAind STA0
D-UAreq		S-UABreq STA0	S-UABreq STA0
S-UABind		D-UAind STA0	D-UAind STA0

TABLE B-7/T.433

DTAM-PM state table – Association established, outside transfer

	STA11	STA22	STA111	STA221
D-TRreq	transfer STA30			
D-TRreq*	resume STA30			
S-ASind		STA40		
S-ARind		STA40		
D-TPreq		S-PTreq STA22		S-PTreq STA221
S-TPind	D-TPind STA11		D-TPind STA111	
D-CGreq	S-CGreq STA22			
S-CGind		D-CGind STA11		
D-CAPreq	S-CAPreq STA111			
S-CAPind		D-CAPind STA221		
D-CAPres				S-CAPres STA22
S-CAPcnf			D-CAPcnf STA11	
S-PABind	D-PAind STA0	D-PAind STA0	D-PAind STA0	D-PAind STA0
D-UAreq	S-UABreq STA0	S-UABreq STA0	S-UABreq STA0	S-UABreq STA0
S-UABind	D-UAind STA0	D-UAind STA0	D-UAind STA0	D-UAind STA0

TABLE B-8/T.433

DTAM-PM state table – Sending DTAM-PM, transfer

	STA30	STA31	STA32
transfer	[a30] S-ASreq next STA30		
resume	[a30] S-ARreq next STA30		
next	p32&p31 SEG S-MSreq [a31] next STA30 p32&p31 SEG S-AEreq STA32 ¬ p32 STA31		
S-MScnf	p37 [a32] STA30 ¬ p37 tr-p-ab STA71*	p37 [a32] next STA30 ¬ p37 tr-p-ab STA71*	p37 [a32] STA32 ¬ p37 tr-p-ab STA71*
S-AEcnf			p33 D-TRcnf+ STA11 ¬ p33 tr-p-ab STA71*
S-PTind	D-TPind STA30	D-TPind STA30	D-TPind STA30
S-UEind	p361 tr-p-ab STA71 ¬ p361&p35 tr-interr STA34 ¬ p361&¬ p35 tr-discard STA34	p361 tr-p-ab STA71 ¬ p361&p35 tr-interr STA34 ¬ p361&¬ p35 tr-discard STA34	p361 tr-p-ab STA71 ¬ p361&p35 tr-interr STA34 ¬ p361&¬ p35 tr-discard STA34
S-PEind	p35 tr-interr STA34 ¬ p35 tr-discard STA34	p35 tr-interr STA34 ¬ p35 tr-discard STA34	p35 tr-interr STA34 ¬ p35 tr-discard STA34
S-PABind	tr-p-ab STA70	tr-p-ab STA70	tr-p-ab STA70
D-UAreq	S-UABreq STA0	S-UABreq STA0	S-UABreq STA0
S-UABind	tr-p-ab STA70	tr-p-ab STA70	tr-p-ab STA70

TABLE B-9/T.433

DTAM-PM state table – Sending DTAM-PM, error handling

	STA34	STA341
tr-discard	S-ADreq STA341	
tr-interr	S-Alreq STA341	
S-ADcnf		D-TRcnf– STA11
S-Alcnf		D-TRcnf– STA11
S-PABind	tr-p-ab STA70	tr-p-ab STA70
D-UABreq	S-UABreq STA0	S-UABreq STA0
S-UABind	tr-p-ab STA70	tr-p-ab STA70
S-PTind	STA34*	STA341

TABLE B-10/T.433

DTAM-PM state table – receiving DTAM-PM

	STA40	STA41
SEG	[a10] STA41	
S-MSind		p41 S-MSres STA40
S-AEind		p42 S-AEres D-TRind STA22
S-Alind	S-Alres STA22	S-Alres STA22
S-ADind	S-ADres STA22	S-ADres STA22
S-PEind	STA40	STA41
D-TPreq	S-PTreq STA40	S-PTreq STA41
S-PABind	tr-p-ab STA70	tr-p-ab STA70
D-UABreq	S-UABreq STA0	S-UABreq STA0
S-UABind	tr-p-ab STA70	tr-p-ab STA70

TABLE B-11/T.433

DTAM-PM state table – Association release

	STA70*	STA71*	STA11	STA22	STA91	STA92
D-TERreq			p11 S-RELreq STA92			
S-RELind				¬ p11 D-TERind STA91		
D-TERres					S-RELres STA0	
S-RELcnf						D-TERcnf STA0
tr-p-ab	D-TRcnf– D-PAind STA0	D-TRcnf– S-UABreq D-PAind STA0				
S-PABind		STA70	D-PAind STA0	D-PAind STA0	D-PAind STA0	D-PAind STA0
D-UAreq	STA70	STA71	S-UABreq STA0	S-UABreq STA0	S-UABreq STA0	
S-UABind		STA70	D-UAind STA0	D-UAind STA0	D-UAind STA0	

B.2 *Conventions*

The intersection of an incoming event (row) and a state (column) forms a cell.

In the state table, a blank cell represents the combination of an incoming event and a state that is not defined for the DTAM-PM (see § B.3.1). Some states await solely some incoming events from the source DTAM-PM (internal events). These states are marked by * and no other incoming events are considered.

A non-blank cell represents an incoming event and a state that is defined for the DTAM-PM. Such a cell contains one or more action lists. An action list may be either mandatory or conditional. If a cell contains a mandatory action list, it is the only action list in the cell.

A mandatory action list contains:

a) optionally one or more outgoing events;

b) optionally one or more specific actions;

c) an resultant state.

A conditional action list contains:

a) a predicate expression comprising predicates and Boolean operators (– represents the Boolean NOT, & represents the Boolean AND);

b) a mandatory action list. (This mandatory list is used only if the predicate expression is true.)

B.3 *Actions to be taken by the DTAM-PM*

The DTAM-PM table defines the actions to be taken by the DTAM-PM in terms of an optional outgoing event, optional specific actions, and the resultant state of the application-association.

B.3.1 *Invalid intersections*

Blank cells indicate an invalid intersection of an incoming event and state. If such an intersections occurs, one of the following actions is taken:

a) if the incoming event comes from the DTAM-SE user, or is an internal event, any action taken by the DTAM-PM is a local matter;

b) if the incoming event is related to a received APDU, or SS-provider, either the DTAM-PM issues an appropriate internal event, or the DTAM-PM issues a DTAMind outgoing event (to its DTAM-SE-user) and an abort outgoing event (to its peer DTAM-PM).

B.3.2 *Valid intersections*

If the intersection of the state and incoming event is valid, one of the following actions is taken:

a) If the cell contains a mandatory action list, the DTAM-PM takes the actions specified.

b) If a cell contains one or more conditional action lists, for each predicate expression that is true, the DTAM-PM takes the actions specified. If none of the predicate expressions are true, the DTAM-PM takes one of the actions defined in § B.3.1.

B.4 *Definition of variables*

The following variables are specified.

B.4.1 *Association-initiating DTAM-PM*

This Boolean variable is set TRUE if the DTAM-PM is the association-initiating DTAM-PM (specific action [a1]), otherwise it is set FALSE (specific action [a2]).

This Boolean variable is tested in the predicate p11.

B.4.2 *Checkpoint-confirmed*

This Boolean variable is TRUE, if at least one checkpoint was confirmed during the transfer procedure. It is set FALSE at the beginning of the transfer procedure (specific action [a30]). It is set TRUE, if an S-MINOR-SYNCHRONIZE confirm positive is issued to the sending DTAM-PM (specific action [a32]).

B.4.3 *Transfer-completed*

This Boolean variable is TRUE, if the receiving DTAM-PM aborted the association because it could not discard an already completed transfer. It is set by the specific actions [a93] and [a94].

This Boolean variable is tested in the predicate p37.

B.4.4 *Outstanding-minor-syncs*

This integer variable indicates the number of outstanding checkpoint confirmations during the transfer procedure. It is set to zero at the beginning of the transfer procedure (specific action [a30]). It is incremented by one, a S-MINOR-SYNCHRONIZE request primitive is issued by the sending DTAM-PM (specific action [a31]).

The value of this variable is compared with the value of the window-size field of the S-CONFcnf in the predicate p32. The value of this variable is compared with the value zero in the predicate p33.

ANNEX C

(to Recommendation T.433)

DTAM-PM state tables

(Document bulk transfer/Normal mode)

C.1 *General*

The DTAM-PM state table described in this annex is applicable to Document Bulk Transfer based on the use of RTSE.

This annex defines a single DTAM Protocol Machine (DTAM-PM) in terms of a state table. The state table shows the interrelationship between the state of an application association, the incoming events that occur (protocol data units) and the actions taken.

This annex contains the following tables:

a) Table C-1/T.433 specifies the abbreviated name, source and name/description of each incoming event. The sources are:

   1) DTAM-SE-user (DTAM-SE-user);

   2) peer DTAM-PM (DTAM-PM-peer);

   3) Presentation Service Provider (Present. SP);

   4) RTSE Service Provider (RTSE SP).

b) Table C-2/T.433 specifies the abbreviated name of each state of the DTAM-PM.

c) Table C-3/T.433 specifies the abbreviated name, target, and name/description of each outgoing event. The targets are:

   1) DTAM-SE-user (DTAM-SE-user);

   2) peer DTAM-PM (DTAM-PM-peer);

   3) Presentation Service Provider (Present. SP);

   4) RTSE Service Provider (RTSE SP).

d) Table C-4/T.433 specifies the predicates;

e) Tables C-5/T.433 through C-8/T.433 inclusive specify the DTAM-PM state table using the abbreviations of the above tables.

TABLE C-1/T.433

Incoming events list

Abbreviated name	Source	Name and description
D-CAPreq	DTAM-SE-user	D-CAPABILITY request primitive
D-CAPres+	DTAM-SE-user	D-CAPABILITY response primitive (Result = "accepted")
D-CAPres–	DTAM-SE-user	D-CAPABILITY response primitive (Result = "rejected")
D-INTreq	DTAM-SE-user	D-INITIATE request primitive
D-INTres+	DTAM-SE-user	D-INITIATE response primitive (Result = "accepted")
D-INTres–	DTAM-SE-user	D-INITIATE response primitive (Result = "rejected")
D-TERreq	DTAM-SE-user	D-TERMINATE request primitive
D-TERres	DTAM-SE-user	D-TERMINATE response primitive
D-TRreq	DTAM-SE-user	D-TRANSFER request primitive
D-TPreq	DTAM-SE-user	D-TOKEN-PLEASE request primitive
D-CGreq	DTAM-SE-user	D-CONTROL-GIVE request primitive
D-UAreq	DTAM-SE-user	D-U-ABORT request primitive
DCPQ	DTAM-PM-PEER	D-CAPABILITY-REQ APDU as user data of an P-CAPABILITY-DATA indication primitive
DCPR+	DTAM-PM-PEER	D-CAPABILITY-RESP APDU as user data of an P-CAPABILITY-DATA confirm primitive (Result = "accepted")
DCPR–	DTAM-PM-PEER	D-CAPABILITY-RESP APDU as user data of an P-CAPABILITY-DATA confirm primitive (Result = "rejected by ...")
DINQ	DTAM-PM-PEER	D-INITIATE-REQ APDU as user data of an RT-OPEN indication primitive
DINR+	DTAM-PM-PEER	D-INITIATE-RESP APDU as user data of an RT-OPEN confirm primitive (Result = "accepted")
DINR–	DTAM-PM-PEER	D-INITIATE-RESP APDU as user data of an RT-OPEN confirm primitive (Result = "rejected by ...")
DTEQ	DTAM-PM-PEER	D-TERMINATE-REQ APDU as user data of an RT-CLOSE indication primitive
DTER	DTAM-PM-PEER	D-TERMINATE-RESP APDU as user data of an RT-CLOSE confirm primitive
P-CDind	Present.SP	P-CAPABILITY-DATA indication
P-CDconf	Present.SP	P-CAPABILITY-DATA confirmation
RT-OPENind	RTSE SP	RT-OPEN indication primitive
RT-OPENcnf	RTSE SP	RT-OPEN confirmation primitive
RT-P-ABind	RTSE SP	RT-Provider abort indication
RT-U-ABind	RTSE SP	RT-User abort indication
RT-TRind	RTSE SP	RT-TRANSFER indication primitive
RT-TRcnf+	RTSE SP	positive RT-TRANSFER conf. primitive
RT-TRcnf–	RTSE SP	negative RT-TRANSFER conf. primitive
RT-TGind	RTSE SP	RT-TURN-GIVE indication primitive
RT-TPind	RTSE SP	RT-TURN-PLEASE indication primitive

TABLE C-2/T.433

DTAM-PM states

Abbreviated name	Name and description
STA0	Idle: unassociated
STA01	Awaiting DINR+ , DINR– (initiator) as user data of an RT-OPEN confirm primitive
STA02	Awaiting D-INTres+, D-INTres– (responder)
STA11	Associated: DTAM-PM is the sending DTAM-PM (association-initiating or responding DTAM-PM)
STA22	Associated: DTAM-PM is the receiving DTAM-PM (association-initiating or responding DTAM-PM)
STA111	Associated: DTAM-PM is the sending DTAM-PM and awaiting DCPR+ or DCPR– (association-initiating or responding DTAM-PM)
STA221	Associated: DTAM-PM is the receiving DTAM-PM and awaiting D-CAPres+ or D-CAPres– (association-initiating or responding DTAM-PM)
STA30	Associated: DTAM-PM is the sending DTAM-PM and awaiting RT-TRcnf+ or RT-TRcnf– (association-initiating or responding DTAM-PM)
STA91	DTAM-PM is association-responder and awaiting D-TERresp
STA92	DTAM-PM is association-initiator and awaiting D-TER

TABLE C-3/T.433

Outgoing events list

Abbreviated name	Target	Name and description
D-CAPind	DTAM-SE-user	D-CAPABILITY indication primitive
D-CAPcnf+	DTAM-SE-user	D-CAPABILITY confirm primitive (Result = "accepted")
D-CAPcnf–	DTAM-SE-user	D-CAPABILITY confirm primitive (Result = "rejected")
D-INTind	DTAM-SE-user	D-INITIATE indication primitive
D-INTcnf+	DTAM-SE-user	D-INITIATE confirm primitive (Result = "accepted")
D-INTcnf–	DTAM-SE-user	D-INITIATE confirm primitive (Result = "rejected")
D-TERind	DTAM-SE-user	D-TERMINATE indication primitive
D-TERcnf	DTAM-SE-user	D-TERMINATE confirm primitive
D-TRind	DTAM-SE-user	D-TRANSFER indication primitive
D-TRcnf+	DTAM-SE-user	D-TRANSFER confirm primitive (Result = "Document transferred")
D-TRcnf–	DTAM-SE-user	D-TRANSFER confirm primitive (Result = "Document not transferred")
D-CGind	DTAM-SE-user	D-CONTROL-GIVE indication primitive
D-TPind	DTAM-SE-user	D-TOKEN-PLEASE indication primitive
D-UAind	DTAM-SE-user	D-U-ABORT indication primitive
D-PAind	DTAM-SE-user	D-P-ABORT indication primitive
DCPQ	DTAM-PM-PEER	D-CAPABILITY-REQ APDU as user data of an P-CAPABILITY-DATA request primitive
DCPR+	DTAM-PM-PEER	D-CAPABILITY-RESP APDU as user data of an P-CAPABILITY-DATA response primitive (Result = "accepted")
DCPR–	DTAM-PM-PEER	D-CAPABILITY-RESP APDU as user data of an P-CAPABILITY-DATA response primitive (Result = "rejected by...")
DINQ	DTAM-PM-PEER	D-INITIATE-REQ APDU as user data of an RT-OPEN request primitive
DINR+	DTAM-PM-PEER	D-INITIATE-RESP APDU as user data of an RT-OPEN response primitive (Result = "accepted")
DINR–	DTAM-PM-PEER	D-INITIATE-RESP APDU as user data of an RT-OPEN response primitive (Result = "rejected by...")
DTEQ	DTAM-PM-PEER	D-TERMINATE-REQ APDU as user data of an RT-CLOSE request primitive
DTER	DTAM-PM-PEER	D-TERMINATE-RESP APDU as user data of an RT-CLOSE response primitive
P-CDreq	Present. SP	P-CAPABILITY-DATA request
P-CDresp	Present. SP	P-CAPABILITY-DATA response
RT-CLOSEresp	RTSE SP	RT-CLOSE response primitive
RT-CLOSEreq	RTSE SP	RT-CLOSE request primitive
RT-OPENresp	RTSE SP	RT-OPEN response primitive
RT-OPENreq	RTSE SP	RT-OPEN request primitive
RT-TRreq	RTSE SP	RT-TRANSFER request primitive
RT-TGreq	RTSE SP	RT-TURN-GIVE request primitive
RT-TPreq	RTSE SP	RT-TURN-PLEASE request primitive
RT-U-ABreq	RTSE SP	RT-User-Abort request primitive

TABLE C-4/T.433

Predicates

Code	Name and description
p1	DTAM-PM can support the request application-association (connection)
p11	Association-initiating DTAM-PM
p12	Association-initiating DTAM-PM and the DTAM-user is agreed to give the token

TABLE C-5/T.433

DTAM-PM state table – Association establishment

	STA0	STA01	STA02
D-INTreq	p1: RT-OPENreq [DINQ] STA01		
D-INTres+			RT-OPENresp [DINR+] STA22
D-INTres–			RT-OPENresp [DINR–] STA0
RT-OPENind [DINQ]	p1: D-INTind STA02 ¬ p1: RT-OPENresp [DINR–] STA0		
RT-OPENcnf [DINR+]		D-INTcnf+ STA11	
RT-OPENcnf [DINR–]		D-INTcnf– STA0	
D-UAreq		RT-U-ABreq STA0	RT-U-ABreq STA0
RT-U-ABind		D-UAind STA0	D-UAind STA0
RT-P-ABind		D-PAind STA0	D-PAind STA0

TABLE C-6/T.433

DTAM-PM table – Association-establishment, transfer

	STA11	STA22
D-TRreq	RT-TRreq STA30	
RT-TRind		D-TRind STA22
D-CAPreq	P-CDreq [DCPQ] STA111	
P-CDind [DCPQ]		D-CAPind STA221
D-TERreq	p11: RT-CLOSEreq [DTEQ] STA92	
RT-CLOSEind [DTEQ]		–p11: D-TERind STA91 p11: RT-U-ABreq D-PAind STA0
RT-TPind	D-TPind STA11	
D-TPreq		RT-TPreq STA22
D-CGreq	RT-TGreq STA22	
RT-TGind		D-CGind STA11
D-UAreq	RT-U-ABreq STA0	RT-U-ABreq STA0
RT-U-ABind	D-UAind STA0	D-UAind STA0
RT-P-ABind	D-PAind STA0	D-PAind STA0
D-CAPres+		P-CDresp [DCPR+] STA22
D-CAPres–		P-CDresp [DCPR–] STA22
P-CDcnf [DCPR+]	D-CAPcnf+ STA11	
P-CDcnf [DCPR–]	D-CAPcnf– STA11	
RT-TPind	D-TPind STA111	
D-UAreq	RT-U-ABreq STA0	RT-U-ABreq STA0
RT-U-ABind	D-UAind STA0	D-UAind STA0
RT-PAind	D-PAind STA0	D-PAind STA0

DTAM-PM, transfer – Transfer by RTSE

	STA30
RT-TRcnf+	D-TRconf+ STA11
RT-TRcnf–	D-TRconf– STA11
RT-TPind	D-TPind STA30
D-UAreq	RT-U-ABreq STA0
RT-U-ABind	D-UAind STA0
RT-PAind	D-PAind STA0

TABLE C-8/T.433

DTAM-PM state table – Abort and association-release

	STA91	STA92
D-TER resp	RT-CLOSEresp [DTER] STA0	
RT-CLOSEcnf [DTER]		D-TERcnf STA0
D-UAreq	RT-U-ABreq STA0	RT-U-ABreq STA0
RT-U-ABind	D-UAind STA0	D-UAind STA0
RT-PAind	D-PAind STA0	D-PAind STA0