INTERNATIONAL TELECOMMUNICATION UNION

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

# T.435

(08/95)

## TERMINALS FOR TELEMATIC SERVICES

# DOCUMENT TRANSFER AND MANIPULATION (DTAM) – SERVICES AND PROTOCOLS – ABSTRACT SERVICE DEFINITION AND PROCEDURES FOR CONFIRMED DOCUMENT MANIPULATION

## ITU-T Recommendation T.435

(Previously "CCITT Recommendation")

## FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T 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 World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation T.435 was prepared by ITU-T Study Group 8 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 11th of August 1995.

___________________

NOTE

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

# CONTENTS

*Page*

# SUMMARY

This Recommendation defines the abstract services for document remote access, presentation, editing, and transfer. It contains two new application service elements for interactive confirmed document manipulation, aligned with, and provided for, the integrated use with applications defined under the Distributed Office Application Model (DOAM). The two application service elements are the DTAM-TM and DTAM-TK.

DTAM-DM provides the facilities to allow all type of remote document manipulation of full documents and document fragments. It is especially designed to be combined with the ODA Abstract Interface for Document manipulation (Recommendation T.413). If document storage and access is requested, provision is made in DTAM-DM that full use of the document filing and retrieval functions defined in ISO/IEC 10166 (DFR) can be made within the DTAM based association. DTAM-TK provides an application token function so that two or more users can communicate in conferences.

**Recommendation T.435**

# DOCUMENT TRANSFER AND MANIPULATION (DTAM) – SERVICES AND PROTOCOLS – ABSTRACT SERVICE DEFINITION AND PROCEDURES FOR CONFIRMED DOCUMENT MANIPULATION

*(Geneva, 1995)*

## 1 Scope

ITU-T (former CCITT) specified the seven-layer reference model of Open Systems Interconnection (OSI), Recommendation X.200. The overall objective of the OSI reference model is to define standards allowing systems to communicate.

The OSI application layer is composed of the Application Service Elements (ASE) which are intended to provide specific services for applications.

One of these application service elements is Document Transfer And Manipulation (DTAM). DTAM is designed for document handling, that is, transfer of documents and access and manipulation of document fragments between application entities within open telematic systems. DTAM is defined in the T.430-Series of Recommendations. Recommendation T.431 gives an introduction and describes general principles of DTAM. Recommendations T.432 and T.433 define the application service element for document bulk transfer and unconfirmed document manipulation.

This Recommendation, together with the protocol specifications contained in Recommendation T.436, define two new application service elements for interactive confirmed document manipulation, aligned with (and provided for) the integrated use with applications defined under the Distributed Office Application Model (DOAM), ISO/IEC 10031.

The two application service elements defined by this Recommendation are:

1) DTAM-DM – The DTAM application service element for enhanced Document Manipulation; and

2) DTAM-TK – The DTAM application service element for application ToKen exchange.

The DTAM-DM abstract services provide facilities to allow remote operations on any kind of documents or document fragments. If the documents of interest are structured amount of information according to the Open Document Architecture (ODA), the document fragments represent the ODA constituents, the T.410 – and T.420 – Series of Recommendations. Documents of other types may provide other means to identify the document fragments uniquely. Documents and document fragments are considered to be manipulation objects within the scope of this Recommendation.

DTAM-DM can be combined with the Document Filing and Retrieval (DFR) application service element, ISO/IEC 10166 for document access and manipulation applications. Both DFR and DTAM-DM are applications that allow remote operations on any kind of documents. While DTAM-DM is concerned with the manipulation of document fragments, the filing and retrieval of complete documents is in the scope of DFR. When there is a need to manipulate documents altogether as well as document fragments that are kept within a document store, the combined use of DTAM-DM and DFR abstract services is applied.

The DTAM-TK application service element provides for the handling of an application token to structure a dialogue, if two or more users are involved in conference type of applications. DTAM-TK is intended to be used in conjunction with other application service elements, for example, together with DTAM-DM.

The DTAM-DM and DTAM-TK abstract services are provided in conjunction with the Association Control Service Element (ACSE), Recommendation X.217, the Remote Operation Service Element (ROSE), Recommendation X.219, and the Presentation Service, Recommendation X.216.

This Recommendation does not specify individual implementations or products, nor does it contain the implementation of entities or interfaces within a telematic system.

## 2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation T.411 (1993) ISO/IEC 8613-1:1994, *Information technology – Open Document Architecture (ODA) and interchange format – Introduction and general principles.*

- ITU-T Recommendation T.412 (1993) ISO/IEC 8613-2:1995, *Information technology – Open Document Architecture (ODA) and interchange format – Document structures*.

- ITU-T Recommendation T.413 (1994) ISO/IEC 8613-3 (1994), *Information technology Open Document Architecture (ODA) and interchange format – Abstract interface for the manipulation of ODA documents.*

- ITU-T Recommendation T.414 (1993) ISO/IEC 8613-4:1994, *Information technology – Open Document Architecture (ODA) and interchange format – Document profile.*

- ITU-T Recommendation T.415 (1993) ISO/IEC 8613-5:1994, *Information technology – Open Document Architecture (ODA) and interchange format – Open Document Interchange Format (ODIF).*

- ITU-T Recommendation T.416 (1993) ISO/IEC 8613-6:1994, *Information technology – Open Document Architecture (ODA) and interchange format – Character content architecture.*

- ITU-T Recommendation T.417 (1993) ISO/IEC 8613-7:1994, *Information technology – Open Document Architecture (ODA) and interchange formats – Raster graphics content architectures.*

- ITU-T Recommendation T.418 (1993) ISO/IEC 8613-8:1994, *Information technology – Open Document Architecture (ODA) and interchange format – Geometric graphics content architecture.*

- ITU-T Recommendation T.422 (1995) ISO/IEC 8613-12:1994, – *Open Document Architecture (ODA) and interchange format – Document pieces identification.*

- CCITT Recommendation T.431 (1992), *Document Transfer And Manipulation (DTAM) – Services and protocols – Introduction and general principles.*

- CCITT Recommendation T.432 (1992), *Document Transfer And Manipulation (DTAM) – Services and protocols – Service definition.*

- CCITT Recommendation T.433 (1992), *Document Transfer And Manipulation (DTAM) – Services and protocols – Protocol specification.*

- CCITT Recommendation T.434 (1992), *Binary file transfer format for the telematic services*.

- ITU-T Recommendation T.436 (1995), *Document Transfer And Manipulation (DTAM) – Services and protocols – Protocol specifications for confirmed document manipulation.*

- CCITT Recommendation X.200 (1988), *Reference model of Open Systems Interconnection for CCITT applications.*

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1).*

- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).*

- CCITT Recommendation X.215 (1988), *Session service definition for Open Systems Interconnection for CCITT applications.*

- CCITT Recommendation X.216 (1988), *Presentation service definition for Open Systems Interconnection for CCITT applications.*

- CCITT Recommendation X.217 (1992), *Association control service definition for Open Systems Interconnection for CCITT applications.*

- CCITT Recommendation X.219 (1988), *Remote operations: Model, notation and service definition.*

- ISO/IEC 10021-3: 1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 3: Abstract service definition conventions.*
- ISO/IEC 10031-1: 1991, *Information technology – Text and office systems – Distributed-office-applications Model – Part 1: General model.*
- ISO/IEC 10166-1:1991, *Information technology – Text and office systems – Document Filing and Retrieval (DFR) – Part 1: Abstract service definition and procedures.*
- ISO/IEC 10166-2: 1991, *Information technology – Text and Office systems – Document Filing and Retrieval (DFR) – Part 2: Protocol specification.*

## 3 Definitions

Definitions are given in Recommendation T.431.

Additionally, the following definitions are used in this Recommendation:

**3.1 DTAM-DM client**: The object in the DTAM-DM environment which consumes the DTAM-DM abstract services provided by a DTAM-DM server.

**3.2 DTAM-DM server**: The object in the DTAM-DM environment which supplies the DTAM-DM abstract services.

**3.3 DTAM-DM port**: The connection between DTAM-DM client and DTAM-DM server which provides the abstract services, that is, the means to communicate in the DTAM-DM environment.

**3.4 DTAM-DM protocol**: The collection of rules which support the realization of the DTAM-DM port abstract services. This set of rules is normally given in a separate protocol specification.

**3.5 DTAM-DM environment**: A constellation of communicating objects connected by a DTAM-DM port. The communication procedure follows the rules determined by the DTAM-DM protocol.

**3.6 DTAM-TK client/server**: The object in the DTAM-TK environment which performs the DTAM-TK abstract services.

**3.7 DTAM-DM/DTAM-TK client/server**: The object in the DTAM-DM/DTAM-TK environment which performs the DTAM-DM and DTAM-TK abstract services.

**3.8 DTAM-TK port**: The connection between application entities which provides the DTAM-TK abstract services.

**3.9 DTAM-TK protocol**: The collection of rules which support the realization of the DTAM-TK port abstract services. This set of rules is normally given in a separate protocol specification.

**3.10 DTAM-DM/DTAM-TK environment**: A communication application which allows the combined usage of DTAM-DM and DTAM-TK abstract services.

**3.11 DTAM-DM/DFR client**: The consumer of abstract services supplied by a DTAM-DM/DFR server.

**3.12 DTAM-DM/DFR server**: The part of a DTAM-DM/DFR environment that supplies the combined usage of DTAM-DM and DFR abstract services.

**3.13 DTAM-DM/DFR environment**: A communication application which allows the combined usage of DTAM-DM and DFR abstract services.

## 4 Abbreviations

Abbreviations are defined in Recommendation T.431.

Additionally, the following abbreviations are used in this Recommendation:

| | |
|---|---|
| DM | Document Manipulation |
| TK | ToKen exchange |
| DFR | Document Filing and Retrieval |
| MO | Manipulation Object |
| UPI | Unique-Permanent-Identifier |

# 5 Conventions

This Recommendation specifies the DTAM-DM and DTAM-TK abstract services using the abstract syntax notation one, Recommendations X.208 and X.209, in conjunction with the abstract service definition conventions, ISO/IEC 10021.

The applicability of services or service parameters is expressed by one of the following values:

| | |
|---|---|
| <Blank> | Not applicable. |
| M | Mandatory. The item shall be present in any case. |
| U | User option. The presence of the item depends on the user’s capabilities. The item may be absent. |
| C | Conditional. The presence of the item depends on conditions stated in this Recommendation. |
| FS | Further Study. Applicability is not clear at this time. |

NOTE – ASN.1 specifications in this Recommendation are written using “IMPLICIT TAGS” convention, which means systematic omission, at the time of ASN.1 encoding, of all unnecessary “nested” tags, especially those “recovered” by context-specific ones.

# 6 Abstract models

A client-server environment is an asymmetrical communication application which comprises two objects, the client and the server. Client and server are application entities which are connected by a port. The server is waiting to be contacted by the client so that the server can do something for the client.

The port provides the abstract services, that is, the means to communicate in the environment. The collection of rules which support the realization of the port abstract services is normally given in a separate protocol specification.

Using the notation of the client-server model, this Recommendation provides the following abstract communication environments:

- DTAM-DM environment;
- DTAM-DM/DTAM-TK combined environment;
- DTAM-DM/DFR combined environment.

The DTAM-DM environment provides abstract services for manipulations on inner parts of documents, the so called document fragments.

The DTAM-DM/DTAM-TK combined environment provides all abstract services of the DTAM-DM environment, and additionally allows to exchange an application token between two objects, which are client and server at the same time. This combined environment may be used in conference communication applications.

The DTAM-DM/DFR combined environment provides all abstract services of the DTAM-DM environment, and additionally allows to manage a document store. The abstract services provided by DFR allow to handle complete documents altogether instead of document fragments.

These abstract communication environments are described in the following subclauses.

The corresponding protocol specifications are given in Recommendation T.436.

## 6.1 The DTAM-DM environment

This subclause provides an abstract functional model of the DTAM-DM environment. The DTAM-DM environment provides for manipulations of document fragments.

The DTAM-DM environment comprises two objects, the DTAM-DM client and the DTAM-DM server. Both objects are connected by the DTAM-DM port.

Figure 1 shows the DTAM-DM environment.

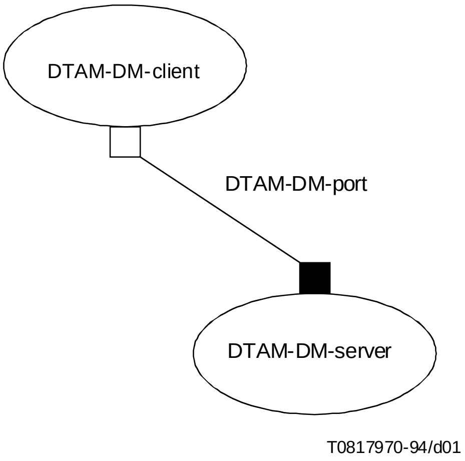


FIGURE 1/T.435

**The DTAM-DM environment**

### 6.1.1 The DTAM-DM client

The DTAM-DM client is modeled as an atomic object. The DTAM-DM client consumes the DTAM-DM port abstract services supplied by the DTAM-DM server.

The formal definition of the DTAM-DM client object is:

```
dtam-dm-client    OBJECT
                  PORTS { dtam-dm-port [ C ] }
                  ::= id-ot-dtam-dm-client
```

### 6.1.2 The DTAM-DM server

The DTAM-DM server is modeled as a separate object. It supplies the DTAM-DM port abstract services to the DTAM-DM client.

The formal definition of the DTAM-DM server object is:

```
dtam-dm-server    OBJECT
                  PORTS { dtam-dm-port [ S ] }
                  ::= id-ot-dtam-dm-server
```

### 6.1.3 The DTAM-DM port

The DTAM-DM port provides the means for the objects to communicate in the DTAM-DM environment.

The DTAM-DM port is defined as follows:

```
dtam-dm-port    PORT
                CONSUMER INVOKES
                {
                        DM-DOCUMENT-OPEN,
                        DM-DOCUMENT-SAVE,
                        DM-DOCUMENT-DISCARD,
                        DM-DOCUMENT-CLOSE,
                        DM-DOCUMENT-LIST,
                        DM-GET,
                        DM-SEARCH,
                        DM-CREATE,
                        DM-DELETE,
                        DM-MODIFY,
                        DM-COPY,
                        DM-MOVE,
            DM-REPLACE,
            DM-RESERVE,
            DM-UNRESERVE,
            DM-POINT,
            DM-MACRO-CALL,
            DM-GROUP-BEGIN,
            DM-GROUP-END
        }
        SUPPLIER INVOKES
        {
        }
        ::= id-pt-dtam-dm-port
```

### 6.1.4 The DTAM-DM information model

A DTAM-DM server offers the DTAM-DM client operations on documents and on the environment which determines the capabilities how the documents can be manipulated. Documents are substructured objects to be considered as Manipulation Objects (MO) which may be for example:

- ODA documents, substructured into several types of objects, classes, styles, contents, among others;
- ODA document specific constraints defined in Document Application Profiles (DAPs) which allow to handle different semantic features, such as segments, paragraphs, footnotes, tables of contents, etc.;
- SGML documents, substructured into regions determined by tags;
- any other types of data in a structured form (including proprietary formats and covering the handling of multimedia data).

The document manipulation environment consists of setting for the capabilities to manipulate a document (e.g. a profile of operations), and for the capabilities necessary to handle documents (e.g. specific profiles, character sets or non-basic characteristics).

#### 6.1.4.1 Identification of manipulation objects

DTAM document handling provides the operations to handle a previously identified document for different purposes, like manipulation or partial reading. The operations consist of three subsequently identified groups:

- one for handling of the complete document;
- one for handling of structured document fragments;
- and one for handling the structured document content information.

#### 6.1.4.2 Identification of complete documents

Document identification takes place at the beginning and end of document manipulation. For identification of the relevant document for subsequent manipulation or to save, discard or close a manipulation, the document open, document save, document discard and document close abstract services are offered. A simple list abstract service is provided for application which have no support of the document filing and retrieval capabilities.

More than one document may be opened at the same time. In this case, each of the operations for manipulation shall identify the document to which they apply.

Documents that may be identified by the open, save, discard, list or close abstract services are each type of standardized documents or identifiable data.

Identification mechanisms may be the UPI concept used in DFR, ISO/IEC 10166, among others.

#### 6.1.4.3 Identification of document fragments

Document fragments are identified during document manipulation operations. Therefore, the operations that identify the document fragments are the get-, search-, reserve-, unreserve-, create-, delete-, modify-, replace-, move-, macro-call- and point-operations.

Document fragments of a structured document may be identified by the specific identification mechanism used in the relevant type of structured information, for example using the location model of ODA documents, Recommendation T.413.

A second method for identification that is distinguished by a separated parameter is the identification of a document fragment via an expression that allow to derive which fragment(s) is (are) be manipulated.

#### 6.1.4.4 Identification of document contents information

The contents of documents may be identified by the search operation. The subsequent manipulation of unstructured parts of a document is for further study.

## 6.2 The DTAM-DM/DTAM-TK combined environment

For distributed document manipulation, the requirement is identified that an OSI system will often need to be the manipulating client for documents stored in other document servers, as well as being the document server for documents stored in its own OSI system. For this reason it was chosen to define the DTAM-DM/DTAM-TK abstract services as a symmetric service, allowing each entity to perform both the client and the server roles. Nevertheless there are dependencies which determine the role (client or server) of an entity for a specific moment.

The DTAM-DM/DTAM-TK environment comprises one object, the DTAM-DM/DTAM-TK client/server. This object is connected to two ports, the DTAM-DM-SYM port and the DTAM-TK port. See Figure 2.

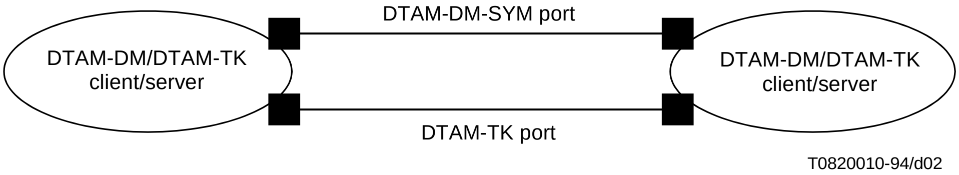


FIGURE 2/T.435

**The DTAM-DM/DTAM-TK combined environment**

### 6.2.1 The DTAM-DM/DTAM-TK client/server

The DTAM-DM/DTAM-TK client/server is modeled as an atomic object. The DTAM-DM/DTAM-TK client/server supports the abstract operations over the DTAM-DM-SYM port and the DTAM-TK port.

The formal definition of the DTAM-DM/DTAM-TK client/server object is:

```
dtam-dm-tk-client-server      OBJECT
                            PORTS { dtam-dm-sym-port, dtam-tk-port  }
                            ::= id-ot-dtam-dm-tk-client-server
```

### 6.2.2 The ports in the DTAM-DM/DTAM-TK environment

The ports in the DTAM-DM/DTAM-TK environment are the DTAM-DM-SYM port and the DTAM-TK port. The DTAM-DM-SYM supports the same abstract operations as the DTAM-DM port, however it is a symmetric port. The DTAM-DM-SYM and the DTAM-TK port are defined in the following subclauses.

#### 6.2.2.1 The DTAM-DM-SYM port

The DTAM-DM-SYM port enables an DTAM-DM/DTAM-TK client/server to manipulate document fragments on other document stores and to perform manipulation operations on its own document store.

The DTAM-DM-SYM port is defined as follows:

```
dtam-dm-sym-port          PORT
         ABSTRACT OPERATIONS {
                  DM-DOCUMENT-OPEN,
                  DM-DOCUMENT-SAVE,
                  DM-DOCUMENT-DISCARD,
                  DM-DOCUMENT-CLOSE,
                  DM-DOCUMENT-LIST,
                  DM-GET,
                  DM-SEARCH,
                  DM-CREATE,
                  DM-DELETE,
                  DM-MODIFY,
                  DM-COPY,
                  DM-MOVE,
                  DM-REPLACE,
                  DM-RESERVE,
                  DM-UNRESERVE,
                  DM-POINT,
                  DM-MACRO-CALL,
                  DM-GROUP-BEGIN,
                  DM-GROUP-END }
            ::= id-pt-dtam-dm-sym-port
```

#### 6.2.2.2 The DTAM-TK port

The DTAM-TK port enables the DTAM-DM/DTAM-TK client/server to exchange an application token which determines the role of the DTAM-DM/DTAM-TK client/server at any time. At one time, the DTAM/DM/DTAM-TK client/server can either act as a server or as a client. The use of an application token is specified in 6.2.3.

The DTAM-TK port is defined as follows:

```
dtam-tk-port          PORT
                  ABSTRACT OPERATIONS {
                          TK-TOKEN-GIVE,
                          TK-TOKEN-PLEASE }
                  ::= id-pt-dtam-tk-port
```

### 6.2.3 The use of an application token

The exchange of an application token between client and server implies that the roles of the application entities in the environment will change. This is illustrated in Figure 3:

User 1 and user 2 are two application entities which are able to operate as client or as server alternatively.

In the start configuration it is assumed that user 1 possesses the application token. User 1 operates as client, while user 2 operates as server. When user 1 performs the DTAM-TK abstract service TK-TOKEN-GIVE, the application token is given to user 2. User 1 switches to the "server-state", while the operation mode of user 2 is changed to "client-state".

Using the DTAM-TK abstract service TK-TOKEN-PLEASE, the actual server can request the application token exchange from the client.

## 6.3 The DTAM-DM/DFR combined environment

This subclause provides an abstract functional model of the DTAM-DM/DFR combined environment. The DTAM-DM/DFR combined environment provides all abstract services of the DTAM-DM environment, and additionally allows to manage a document store.

The DTAM-DM/DFR environment comprises two objects, the DTAM-DM/DFR client and the DTAM-DM/DFR server. Both objects are connected by two ports, the DTAM-DM port and the DFR port.

Figure 4 shows the DTAM-DM/DFR combined environment.

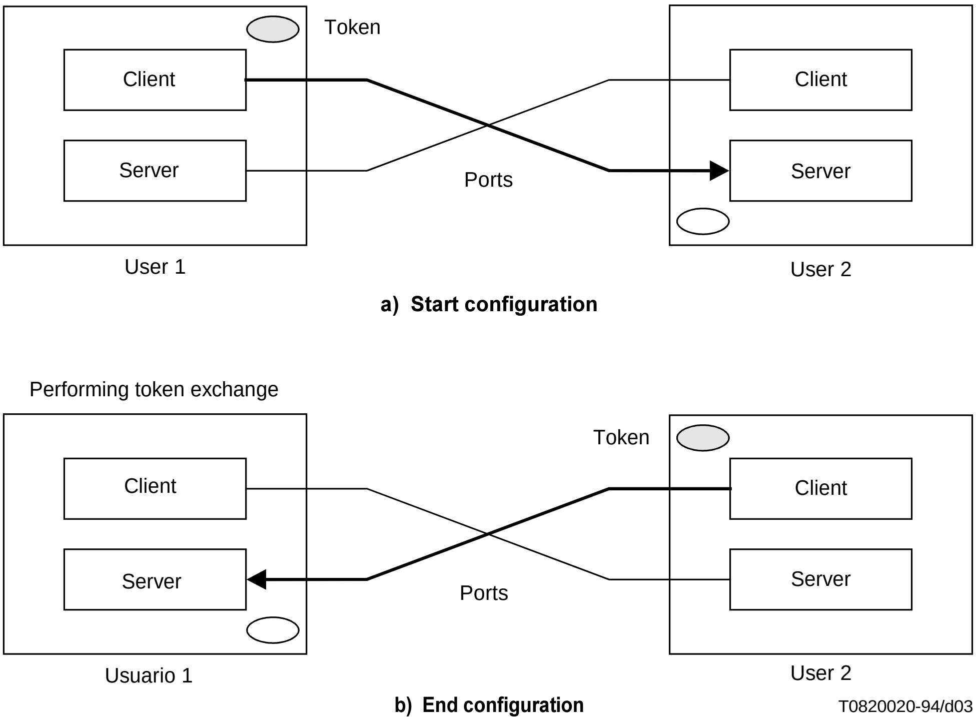


FIGURE 3/T.435

**The use of an application token**

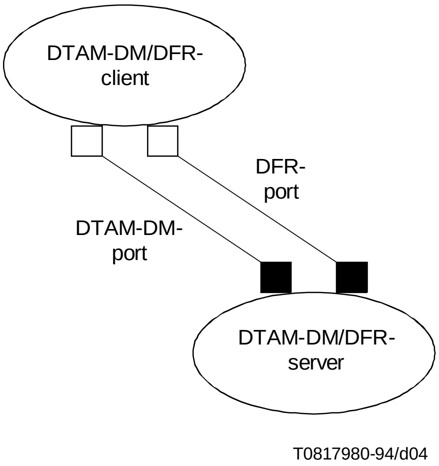


FIGURE 4/T.435

**The DTAM-DM/DFR combined environment**

### 6.3.1 The DTAM-DM/DFR client

The DTAM-DM/DFR client is modeled as an atomic object. The DTAM-DM/DFR client consumes the DTAM-DM port and the DFR port abstract services supplied by the DTAM-DM/DFR server.

The formal definition of the DTAM-DM/DFR client object is:

```
dtam-dm-dfr-client    OBJECT
                      PORTS { dtam-dm-port [ C ], dfr-port [ C ] }
                      ::= id-ot-dtam-dm-dfr-client
```

### 6.3.2 The DTAM-DM/DFR server

The DTAM-DM/DFR server is modeled as a separate object. It supplies the DTAM-DM port and the DFR port abstract services to the DTAM-DM/DFR client.

The formal definition of the DTAM-DM/DFR server object is:

```
dtam-dm-dfr-server    OBJECT
                      PORTS { dtam-dm-port [ S ], dfr-port [ S ] }
                      ::= id-ot-dtam-dm-dfr-server
```

### 6.3.3 The ports in the DTAM-DM/DFR environment

The ports in the DTAM-DM/DFR environment are the DTAM-DM port and the DFR port. The DTAM-DM port is already defined above. The DFR port is defined in ISO/IEC 10166 and imported by this Recommendation.

NOTE – The use of the DM-Document-LIST abstract service is not allowed in the DTAM-DM/DFR environment. In this case the corresponding DFR abstract services shall be used.

# 7 Overview of abstract services

This clause provides an introduction to the DTAM-DM and DTAM-TK abstract services. The formal definition of the abstract services is given in Annex B "Formal definition of abstract services".

## 7.1 Overview of DTAM-DM abstract services

This Recommendation defines the following DTAM-DM abstract services:

- DM-DOCUMENT-OPEN;
- DM-DOCUMENT-SAVE;
- DM-DOCUMENT-DISCARD;
- DM-DOCUMENT-CLOSE;
- DM-DOCUMENT-LIST;
- DM-GET;
- DM-SEARCH;
- DM-CREATE;
- DM-DELETE;
- DM-MODIFY;
- DM-COPY;
- DM-MOVE;
- DM-REPLACE;
- DM-RESERVE;
- DM-UNRESERVE;
- DM-POINT;
- DM-MACRO-CALL;
- DM-GROUP-BEGIN;
- DM-GROUP-END.

These DTAM-DM abstract services are introduced in the following subclauses.

### 7.1.1 DM-DOCUMENT-OPEN abstract service

The document open abstract service is used by the DTAM-DM client to select and open documents prior to document manipulations. No inner manipulations can take place on documents which were not opened before. Several documents may be opened at the same time.

### 7.1.2 DM-DOCUMENT-SAVE abstract service

The document save abstract service is used by the DTAM-DM client to save documents after document manipulations which were done since the use of the last DM-DOCUMENT-OPEN, DM-DOCUMENT-DISCARD or DM-DOCUMENT-SAVE abstract services.

### 7.1.3 DM-DOCUMENT-DISCARD abstract service

The document discard abstract service is used by the DTAM-DM client to discard document manipulations which were done since the use of the last DM-DOCUMENT-SAVE, DM-DOCUMENT-OPEN or DM-DOCUMENT-DISCARD abstract service.

### 7.1.4 DM-DOCUMENT-CLOSE abstract service

The document close abstract service is used by the DTAM-DM client to close and derelict documents after document manipulations. All document manipulations which were done since the use of the last DM-DOCUMENT-OPEN, DM-DOCUMENT-SAVE or DM-DOCUMENT-DISCARD abstract service are discarded. To save the document contents before the document will be closed, the DM-DOCUMENT-SAVE abstract service must be used.

### 7.1.5 DM-DOCUMENT-LIST abstract service

The document list abstract service is used by the DTAM-DM client to request a list of document identifications from the DTAM-DM server that are in-line with the indicated criteria. The use of the document list service is restricted to applications that have limited filing and retrieval capabilities. For applications that support both, filing and retrieval as well as manipulation, the combined usage of DFR and DTAM-DM is recommended.

### 7.1.6 DM-GET abstract service

The get abstract service is used by the DTAM-DM client to read document fragments.

### 7.1.7 DM-SEARCH abstract service

The search abstract service is used by the DTAM-DM client to search for specified document fragments. The DTAM-DM client specifies search domains (e.g. to search only in the first chapter) and search criteria. The search abstract service may be used to identify document fragments for invocation of subsequent abstract operations like modify or create, among others.

### 7.1.8 DM-CREATE abstract service

The create abstract service is used by the DTAM-DM client to add document fragments to an existing document. The create abstract service may carry the document fragment, including the values applicable to the created document fragment. The DTAM-DM client must specify the position of the additional document fragment inside the document.

### 7.1.9 DM-DELETE abstract service

The delete abstract service is used by the DTAM-DM client to delete document fragments.

### 7.1.10 DM-MODIFY abstract service

The modify abstract service is used by the DTAM-DM client to modify already existing document fragments. The modify abstract service assigns new values to attributes of the identified document fragment.

### 7.1.11 DM-COPY abstract service

The copy abstract service is used by the DTAM-DM client to duplicate document fragments to another position of the same or a different document.

#### 7.1.12 DM-MOVE abstract service

The move abstract service is used by the DTAM-DM client to change the position of document fragments inside a document, or from one document to another.

#### 7.1.13 DM-REPLACE abstract service

The replace abstract service is used by the DTAM-DM client to replace document fragments. The replace abstract service effects the deletion of the identified document fragment and the replacement by the new document fragment.

#### 7.1.14 DM-RESERVE abstract service

The reserve abstract service is used by the DTAM-DM client to establish the reservation of the specified document fragment. The reserve abstract operation does not apply to ownership or access rights of document fragments (e.g. the access rights in the document profile of an ODA document).

#### 7.1.15 DM-UNRESERVE abstract service

The unreserve abstract service is used by a DTAM-DM client to release a previous reservation of a document fragment done by using the reserve abstract service.

#### 7.1.16 DM-POINT abstract service

The point abstract service is used by DTAM-DM clients in multi-user applications like joint-editing or joint-viewing.

#### 7.1.17 DM-MACRO-CALL abstract service

This is for further study.

#### 7.1.18 DM-GROUP-BEGIN abstract service

The group-begin abstract service is used by the DTAM-DM client to indicate the start of a set of DTAM-DM abstract services which are to be processed and responded to as a group.

This is for further study.

#### 7.1.19 DM-GROUP-END abstract service

The group-end abstract service is used by the DTAM-DM client to indicate the end of a set of DTAM-DM abstract services which are to be processed and responded to as a group.

This is for further study.

### 7.2 Overview of DTAM-TK abstract services

This Recommendation defines the following DTAM-TK abstract services:

- TK-TOKEN-GIVE;
- TK-TOKEN-PLEASE.

These DTAM-TK abstract services are introduced in the following subclauses.

#### 7.2.1 TK-TOKEN-GIVE abstract service

The token-give abstract service surrenders the application token from the current DTAM-TK client to the DTAM-TK server and changes the roles of the application entities in the communication process (from client to server and vice versa). The application token makes use of the ROSE services, Recommendation X.219, and hence has no relationship to communication tokens used in other OSI-layers, for example, the session tokens.

#### 7.2.2 TK-TOKEN-PLEASE abstract service

The token-please abstract service is used by the DTAM-TK server to request the application token from a DTAM-TK client, that is, a request changes the role of the application entities in the communication process.

# 8 Establishment of an application association

The DTAM-DM and DTAM-TK application service elements are used in conjunction with ACSE services and the formal notation of the abstract bind-operation and the unbind-operation. The specification of the abstract bind and unbind operation is specified in Recommendation X.407. The mapping of the bind-operation and unbind-operation on the ACSE services is described in Recommendation X.219.

In the following explanations the terms “DTAM-DM client” and “DTAM-DM server” are used which have only a meaning in the DTAM-DM and DTAM-DM/DFR environment. In the DTAM-DM/DTAM-TK environment these terms must be interpreted as “requesting client/server” and “responding client/server” respectively. In the DTAM-DM/DFR environment these terms must be interpreted as “DTAM-DM/DFR client” and “DTAM-DM/DFR server” respectively.

## 8.1 Abstract-bind and Abstract-unbind Parameters

### 8.1.1 ABSTRACT-BIND parameters

The DTAM-DMBind abstract operation binds the DTAM-DM port of the DTAM-DM client (consumer) to the DTAM-DM server (supplier). In the DTAM-DM/DTAM-TK environment, the DTAM-DMBind abstract operation binds the DTAM-DM-SYM and DTAM-TK ports of the DTAM-DM/DTAM-TK client/server. In the DTAM-DM/DFR environment, the DTAM-DMBind abstract operation binds the DTAM/DM and DFR ports of the DTAM-DM/DFR client (consumer) to the DTAM-DM/DFR server (supplier).

```
DTAM-DMBind ::= ABSTRACT-BIND
            TO {dtam-dm-port[S], dfr-port[S], dtam-dm-sym-port, dtam-tk-port}
            BIND
                 ARGUMENT      DtamDmBindArgument
                 RESULT        DtamDmBindResult
                 BIND-ERROR    DtamDmBindError
```

### 8.1.1.1 Bind-argument parameters

The DtamDmBindArgument parameter allows to agree on the capabilities used during the association.

```
DtamDmBindArgument ::= [0] SEQUENCE {
           dtamProtocolVersion              [0] BIT STRING
                                            { version-1 (0) } DEFAULT {version-1},
           dtamManipulationCapabilities     [1] ManipulationCapabilities,
           odaApplicationCapabilities       [2] SEQUENCE OF OdaApplicationCapabilities
                                            OPTIONAL,
           dfrCapablities                   [3] DfrCapabilities OPTIONAL,
           bilateralInformation             [4] SEQUENCE OF
                                            BilateralInformation OPTIONAL,
           applicationRequirements          [5] ApplicationRequirements OPTIONAL }
```

a) **dtamProtocolVersion**: This parameter indicates the DTAM-DM protocol versions which the DTAM-DM client requests. It is a variable length bit string where each bit set to one indicates the version of the DTAM-DM protocol which is requested. Multiple bits may be set indicating the support of multiple versions. No trailing bits higher than the highest version of this Recommendation which the client supports are included. That is, the last bit of the string is set to one. When this parameter is absent, the default value “version-1” is applied.

   If the DTAM-DM server supports a requested protocol version, DTAM-DM Abstract-Bind continues normally. Otherwise, the DTAM-DM server will not accept the association with a corresponding error message (see 8.1.1.3).

   Version-1 is defined by Recommendation T.436.

b) **dtamManipulationCapabilities**: This parameter indicates the DTAM-DM capabilities the DTAM/DM client wants to use during the association. The capabilities are negotiated with the DTAM-DM server. If the DTAM-DM server can support these capabilities, the bind operation continues normally. If the DTAM-DM server cannot support these capabilities, the DTAM-DM server will not accept the association with a corresponding error message (see 8.1.1.3).

   It is possible to select a standardized DTAM-DM profile or to agree on known bilateral conventions. Alternatively, the manipulation level can be indicated.

   See Table 1 for an overview of the relationship between DTAM-DM abstract services and the possible values of manipulationLevelSelection hold in the ManipulationCapabilities parameter.

```
ManipulationCapabilities ::= CHOICE {
                profileSelection                [0] OBJECT IDENTIFIER,
                manipulationLevelSelection      [1] ENUMERATED {
                                                basicReadOnlyLevel       (0),
                                                basicManipulationLevel   (1),
                                                extendedLevel            (2) },
                bilateralAgreement              [2] IMPLICIT OCTET STRING }
```

c) **odaApplicationCapabilities**: In the bind-operation, this parameter is optional and shall only be used in the conference type of applications, where each partner has symmetrical capabilities (consists of a client and a server). It specifies the document application profile and, optionally, two other parameters described hereafter as the receiving capabilities of the DTAM-DM/DTAM-TK client/server. The continued progress of the service is only guaranteed if the communication partner acts within the receiving capabilities of the DTAM-DM/DTAM-TK client/server.

The NonBasicOdaDocumentCharacteristics specify the non-basic document characteristics available to the server as the receiving capabilities. The attributes and values that may be specified by this parameter are restricted to those that are permitted by the corresponding document application profile. The format of this parameter is described in Recommendation T.414.

The NonBasicOdaStructuralCharacteristics specify the non-basic structural characteristics available to the server as the receiving capabilities. The attributes and values that may be specified by this parameter are restricted to those that are permitted by the corresponding document application profile. The format of this parameter is described in Recommendation T.414.

```
OdaApplicationCapabilities ::= SEQUENCE {
                odaDocumentApplicationProfile           [0] OBJECT IDENTIFIER,
                nonBasicOdaDocumentCharacteristics      [1] Non-Basic-Doc-Characteristics
                                                        OPTIONAL,
                nonBasicOdaStructuralCharacteristics    [2] Non-Basic-Struc-Characteristics
                                                        OPTIONAL }
```

d) **dfrCapabilities**: In the bind-operation, this parameter is optional and shall only be used in the DTAM-DM/DFR environment. This parameter indicates the DFR capabilities which the DTAM-DM/DFR client requests in the the DTAM-DM/DFR environment. This is a DFR protocol version and, optionally, a DFR profile.

The DFR protocol version is a variable length bit string where each bit set to one indicates the version of the DFR protocol which is requested. Multiple bits may be set indicating the support of multiple versions. No trailing bits higher than the highest version of this Recommendation which the client supports are included. That is, the last bit of the string is set to one.

If the DTAM-DM/DFR server supports the requested DFR capabilities, DTAM-DM Abstract-Bind continues normally. Otherwise, the DTAM-DM/DFR server will not accept the association with a corresponding error message (see 8.1.1.3).

```
DfrCapabilities ::= SEQUENCE {
                dfrProtocolVersion          [0] DfrProtocolVersion DEFAULT {version-1},
                dfrProfile                  [1] OBJECT IDENTIFIER OPTIONAL}

DfrProtocolVersion ::= BIT STRING {
                version-1 (1),
                version-2 (2) }
```

- Version-1 is defined by ISO/IEC 10166-1:1991
- Version-2 is defined by ISO/IEC 10166-1:1991 together with ISO/IEC 10166-1:1991/Amd 1.

e) **bilateralInformation**: This parameter includes bilateral information between the DTAM-DM client and the DTAM-DM server. Format of this bilateral information is as follows:

```
BilateralInformation ::= SEQUENCE {
                serverName              [0]         ServerName,
                information             [1]         EXTERNAL }
ServerName ::= DistinguishedName
```

f) **applicationRequirements**: This parameter indicates additional application requirements which are negotiated between the DTAM server and the client. The meaning of this parameter is application dependent. If the DTAM server can support these requirements, the bind operation continues normally. Otherwise, the DTAM server will not accept the association with a corresponding error message (see 8.1.1.3).

```
ApplicationRequirement ::= CHOICE {
                            [0] OCTET STRING,
                            [1] OBJECT IDENTIFIER }
```

TABLE 1/T.435

**Relation between abstract services and the manipulationLevelSelection values**

| | manipulationLevelSelection | | |
|---|---|---|---|
| DTAM-DM abstract services | basicReadOnlyLevel | basicManipulationLevel | extendedLevel |
| DM-DOCUMENT-OPEN (Note 1) | M | M | M |
| DM-DOCUMENT-SAVE (Note 1) | | M | M |
| DM-DOCUMENT-DISCARD (Note 1) | | M | M |
| DM-DOCUMENT-CLOSE (Note 1) | M | M | M |
| DM-POINT (Note 1) | U | U | U |
| DM-DOCUMENT-LIST (Note 2) | C | C | C |
| DM-GET (Note 3) | M | M | M |
| DM-SEARCH (Note 3) | M | M | M |
| DM-CREATE (Note 4) | | M | M |
| DM-DELETE (Note 4) | | M | M |
| DM-MODIFY (Note 4) | | M | M |
| DM-COPY (Note 4) | | M | M |
| DM-MOVE (Note 5) | | | M |
| DM-REPLACE (Note 5) | | | M |
| DM-RESERVE (Note 5) | | | M |
| DM-UNRESERVE (Note 5) | | | M |
| DM-MACRO-CALL (Note 6) | FS | FS | FS |
| DM-GROUP-BEGIN (Note 6) | FS | FS | FS |
| DM-GROUP-END (Note 6) | FS | FS | FS |

NOTES

1 The abstract services DM-DOCUMENT-OPEN and DM-DOCUMENT-CLOSE have to be supported by all values of manipulationLevelSelection. The use of the DM-POINT service is optional (U).

2 The use of the DM-DOCUMENT-LIST abstract service is not allowed in combined DTAM-DM/DFR applications. In such a case the DFR list abstract service shall be used instead (C).

3 Additionally to the abstract services listed in Note 1, the abstract services DM-GET and DM-SEARCH have to be supported if the value of manipulationLevelSelection is basicReadOnlyLevel (M), other abstract services are not applicable ( <blank>).

4 Additionally to the abstract services listed in Note 3, the abstract services DM-DOCUMENT-SAVE, DM-DOCUMENT-DISCARD, DM-CREATE, DM-DELETE, DM-MODIFY and DM-COPY have to be supported if the value of manipulationLevelSelection is basicManipulationLevel (M), other abstract services are not applicable ( <blank>).

5 Additionally to the abstract services listed in Note 4, the abstract services DM-MOVE, DM-REPLACE, DM-RESERVE, DM-UNRESERVE and DM-POINT have to be supported if the value of manipulationLevelSelection is extendedLevel (M).

6 The support of the abstract services DM-MACRO-CALL, DM-GROUP-BEGIN and DM-GROUP-END is for further study for all values of manipulationLevelSelection (FS).

#### 8.1.1.2 Bind-result parameters

The DtamDmBindResult parameter indicates the capabilities supported by the DTAM-DM server.

```
DtamDmBindResult ::= [1] SEQUENCE {
            serverSelectedDtamProtocolVersion    [0] BIT STRING
                                                  { version-1 (0) } DEFAULT {version-1},
            odaApplicationCapabilities           [1] SEQUENCE OF OdaApplicationCapabilities,
            serverSelectedDfrVersion             [2] BIT STRING
                                                  { version-1 (1),
                                                    version-2 (2) } OPTIONAL,
            bilateralInformation                 [3] SEQUENCE OF
                                                  BilateralInformation OPTIONAL }
```

a) **serverSelectedDtamProtocolVersion**: The parameter indicates the DTAM-DM protocol version which the DTAM-DM server selected from those proposed by the DTAM-DM client. It is a variable length bit string where only the version selected is set to one. The value of this parameter defines the protocol version to be used for this association.

b) **odaApplicationCapabilities**: This parameter specifies the document application profile and, optionally, two other parameters as the receiving capabilities of the DTAM-DM server. The continued progress of the service is only guaranteed if the DTAM-DM client acts within the receiving capabilities of the DTAM-DM server.

c) **serverSelectedDfrProtocolVersion**: This parameter is optional and shall only be used in the DTAM-DM/DFR environment. The parameter indicates the DFR protocol version which the DTAM-DM/DFR server selected from those proposed by the DTAM-DM/DFR client. It is a variable length bit string where only the version selected is set to one. The value of this parameter defines the protocol version to be used for this association.

d) **bilateralInformation**: This parameter includes bilateral information between DTAM-DM server and DTAM-DM client.

#### 8.1.1.3 Bind-error parameters

The DtamDMBindError parameter reports the errors that may occur during association establishment.

DmServiceProblem reports that the DTAM-DM server cannot establish the association due to some operational error.

The parameter ProtocolProblem indicates that the DTAM-DM server does not support the requested protocol versions. If the DTAM-DM server does not support the requested DTAM-DM or DFR protocol version proposed by the DTAM-DM client, it may optionally indicate the protocol versions which could be supported by the DTAM-DM server.

The parameter ProfileProblem indicates that the DTAM-DM server does not support the requested profiles of the protocol. If the DTAM-DM server does not support the requested DTAM-DM or DFR profiles by the DTAM-DM client, it may optionally indicate the profile which could be supported by the DTAM-DM server.

The DmServiceProblem paramter is also used in the DTAM-DM operations and defined in 9.4.8.

The parameter ApplicationProblem indicates that the DTAM-DM server does not support the requested application requirements. If the DTAM-DM server does not support the requested application requirements, it may optionally indicate the application requirements which could be supported by the DTAM-DM server.

```
DtamDmBindError ::= [2] CHOICE {
            serviceError           [0] DmServiceProblem,
            protocolError          [1] ProtocolProblem,
            profileError           [2] ProfileProblem,
            applicationError       [3] ApplicationProblem }

ProtocolProblem ::= SEQUENCE {
           protocol-version-not-supported     [0] SEQUENCE OF ENUMERATED {
                                              dtam-version-not-supported (0),
                                              dfr-version-not-supported    (1) },
           supportedDtamProtocolVersions      [1] BIT STRING
                                               { version-1 (0) } OPTIONAL,
           supportedDfrProtocolVersions       [2] BIT STRING
                                              { version-1 (0),
                                               version-2 (1) } OPTIONAL }
```

```
ProfileProblem ::= SEQUENCE {
        profile-not-supported               [0] SEQUENCE OF ENUMERATED {
                                            dtam-manipulation-level-not-supported   (0),
                                            dfr-profile-not-supported               (1) },
        supportedDtamManipulationLevels     [1] SEQUENCE OF ManipulationCapabilities
                                            OPTIONAL,
        supportedDfrProfiles                [2] SEQUENCE OF OBJECT IDENTIFIER
                                            OPTIONAL }

ApplicationProblem ::= SEQUENCE OF ApplicationRequirements OPTIONAL
```

### 8.1.2 ABSTRACT-UNBIND parameters

This subclause defines and describes the unbind-operation parameters for the DTAM-DM, DTAM-DM-SYM, DFR and DTAM-TK port depending on the environment. There are no arguments or errors associated with the unbind-operation.

```
DTAM-DMUnBind ::= ABSTRACT-UNBIND
          FROM {dtam-dm-port[S], dfr-port[S], dtam-dm-sym-port, dtam-tk-port}
```

## 8.2 Use of the A-ABORT service

The A-ABORT service is initiated by either the client or server to dissolve unconditionally an association. The encoding of the DTAM-DMAbortInfo, which is mapped to the A-ABORT "user information" parameter, is defined by using the ASN.1 EXTERNAL data type.

The DtamDmAbortSource parameter indicates the source of the association abort, if it is the DTAM-DM service provider or service user. The DtamDmAbortReason parameter indicates the reason for the abort. The value "temporary-problem" indicates, that no attempt at association-recovery should be made for a period of time determined by a local rule.

```
DTAM-DMAbortInfo ::= [3] SEQUENCE {
          abortSource             [0] DtamDmAbortSource,
          abortReason             [1] DtamDmAbortReason OPTIONAL,
          userInformation         [2] EXTERNAL OPTIONAL }

DtamDmAbortSource ::= ENUMERATED {
          dtamDmServiceUser       (0),
          dtamDmServiceProvider   (1) }

DtamDmAbortReason ::= ENUMERATED {
          local-system-problem    (0),
          temporary-problem       (1),
          protocol-error          (2) }
```

# 9 Definition of abstract services

This clause provides the definition of the DTAM-DM and DTAM-TK abstract services. The complete formal definition is contained in Annex B.

Some data types used in these definitions are defined in other Recommendations or international standards. These data types are imported by this Recommendation.

An argument of any DTAM-DM or DTAM-TK abstract service is a sequence of parameters, some of them being mandatory for the given abstract service, while others are optional. Some of the parameters may be common to all or most of the services. These parameters are defined as common data types.

The definitions of the DTAM-DM and the DTAM-TK abstract services are given in separate subclauses.

## 9.1 Imported data types

From ODA (Recommendation T.413):

```
                                ConstituentType,
                                ConstituentIdentifier,
                                AttributeValuesType,
                                QueryType
```

From ODA (Recommendation T.415):

**Non-Basic-Doc-Characteristics,**
**Non-Basic-Struc-Characteristics.**
**Document-Reference**

From ODA (Recommendation T.422):

**Location-expression,**
**Basic-location-expression,**
**Constituent-Locator**

From DFR (ISO/IEC 10166):

**DfrEntryName,**

From MHS (Recommendation X.407):

**ABSTRACT-BIND,**
**ABSTRACT-UNBIND,**
**ABSTRACT-ERROR,**
**ABSTRACT-OPERATION,**
**OBJECT,**
**PORT**

From DS (Recommendation X.500):

**DistinguishedName**

## 9.2 Defined data types common for abstract services

The following data types are defined as common data types in this Recommendation:

**DocumentId**
**ManipulationObject**
**ObjectValue**
**ObjectPosition**
**ObjectContent**
**AssignedId**
**NonPermanentIdentifier**

### 9.2.1 DocumentIdentification

```
DocumentId ::= CHOICE {
        dfrName                   [0] DfrEntryName,
        documentReference         [1] Document-Reference,
        documentName              [2] OCTET STRING,
        nonPermanentIdentifier    [3] NonPermanentIdentifier }
```

### 9.2.2 ManipulationObject

```
ManipulationObject ::= SEQUENCE {
        documentIdentifier        [0] DocumentId OPTIONAL,
        objectValue               [1] ObjectValue OPTIONAL }
```

### 9.2.3 ObjectValue

```
ObjectValue ::= CHOICE {
        odaExpression             [0] Oda-expression,
        otherExpression           [1] OCTET STRING,
        anyExpression             [2] ANY }
```

### 9.2.4 Oda-Expression

```
Oda-Expression ::= CHOICE {
        locationExpression        [0] Location-Expression,
        basicLocationExpression   [1] basic-Location-Expression,
        constituentLocator        [2] Constituent-Locator,
        ConstituentIdentifier     [3] Constituent-Identifier }
```

### 9.2.5 ObjectPosition

```
ObjectPosition ::= ENUMERATED {
        before                  (0),
        afterLastChild          (1) }
```

### 9.2.6 ObjectContent

```
ObjectContent ::= CHOICE {
        odaContent              [0] ConstituentType,
        octetAligned            [1] OCTET STRING,
        anyContent              [2] ANY }
```

### 9.2.7 AssignedId

```
AssignedId ::= CHOICE {
        odaIdentifier           [0] ConstituentIdentifier,
        nonOdaIdentifier        [1] INTEGER,
        any                     [2] ANY }
```

### 9.2.8 NonPermanentIdentifier

```
NonPermanentIdentifier ::= CHOICE {
                                [0] INTEGER,
                                [1] OCTET STRING}
```

## 9.3 Definition of DTAM-DM abstract services

This Recommendation defines the following DTAM-DM abstract services:

- DM-DOCUMENT-OPEN;
- DM-DOCUMENT-SAVE;
- DM-DOCUMENT-DISCARD;
- DM-DOCUMENT-CLOSE;
- DM-DOCUMENT-LIST;
- DM-GET;
- DM-SEARCH;
- DM-CREATE;
- DM-DELETE;
- DM-MODIFY;
- DM-COPY;
- DM-MOVE;
- DM-REPLACE;
- DM-RESERVE;
- DM-UNRESERVE;
- DM-POINT;
- DM-MACRO-CALL;
- DM-GROUP-BEGIN;
- DM-GROUP-END.

The DTAM-DM abstract services are described according to the definition of the abstract operations macro in Recommendation X.407. In the DTAM-DM or DTAM-DM/DFR environment, all abstract operations are invoked by the consumer (DTAM-DM or DTAM-DM/DFR client) . In the DTAM-DM/DTAM-TK environment all abstract operations are invoked by the actual client (The application token decides whether the DTAM-DM/DTAM-TK client/server is server or client for a specific moment). Each abstract operation always reports either success or an error. The abstract operations are mapped on the ROSE services defined in Recommendation X.219.

These DTAM-DM abstract services are defined in the following subclauses.

### 9.3.1 DM-DOCUMENT-OPEN abstract service

The document open abstract service is used by the DTAM-DM client to select and open documents prior to document manipulations. No inner manipulations can take place on documents which were not opened before. Several documents may be opened at the same time.

```
DM-DOCUMENT-OPEN ::= ABSTRACT-OPERATION
        ARGUMENT                    DM-DOCUMENT-OPEN-Arguments
        RESULT                      DM-DOCUMENT-OPEN-Result
        ERRORS
        {
                DM-DocumentError,
                DM-AccessError,
                DM-ServiceError
        }

DM-DOCUMENT-OPEN-Arguments ::= SEQUENCE
{
        documentId                  [0] DocumentId,
        mode                        [1] ModeType OPTIONAL
}

ModeType ::= ENUMERATED{ read (0), modify (1), create (2) }

DM-DOCUMENT-OPEN-Result ::= CHOICE
{
        noValue                     [0] NULL,
        nonPermanentIdentifier      [1] NonPermanentIdentifier
}
```

#### 9.3.1.1 Arguments

The documentId parameter is mandatory and identifies the selected document for further manipulation. The mode parameter is optional and may have three values: 'read', 'modify' and 'create'. If a document is opened with “mode” equal to 'read', it will only be possible to perform read operations on it. In case it is opened with “mode” equal to 'modify', it will be possible to apply all manipulation operations. Finally, the 'create' value means that the document is to be created. In this latter case, the permanent identifier is the identification to be given to the document.

#### 9.3.1.2 Result

The DM-DOCUMENT-OPEN-Result will be returned in case of success. It either carries a nonPermanentIdentifier, which will be used up to the related DM-DOCUMENT-CLOSE operation, where a new permanent identifier will be assigned, or carries no parameter, which implies that the relevant permanent identifier is used throughout the manipulation phase.

#### 9.3.1.3 Errors

The reasons for giving an error in the open service are:

- DM-DocumentError: The documentId parameter does not indicate an existing document.
- DM-AccessError: The document is already opened by the user, is reserved or the user has no valid access rights.
- DM-ServiceError: The document cannot be opened due to local reasons.

### 9.3.2 DM-DOCUMENT-SAVE abstract service

The document save abstract service is used by the DTAM-DM client to save documents after document manipulations. This operation is of dedicated importance when used before the DM-DOCUMENT-CLOSE.

```
DM-DOCUMENT-SAVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-DOCUMENT-SAVE-Arguments
        RESULT                  DM-DOCUMENT-SAVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-DOCUMENT-SAVE-Arguments ::= SEQUENCE
{
        documentId              [0] DocumentId
}

DM-DOCUMENT-SAVE-Result ::= NULL
```

#### 9.3.2.1 Arguments

The documentId parameter is mandatory and identifies the selected document for the save service.

#### 9.3.2.2 Result

If the request succeeds, the DM-DOCUMENT-SAVE-Result will be returned. There are no parameters.

In this case, all manipulations that have been invoked since the last DM-DOCUMENT-OPEN, DM DOCUMENT-SAVE or DM-DOCUMENT-DISCARD are permanently stored.

#### 9.3.2.3 Errors

The reasons for giving an error in the save service are:

- DM-AccessError: The document is not opened or is reserved by some other user or the user has no valid access rights.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The document cannot be saved due to local reasons.

### 9.3.3 DM-DOCUMENT-DISCARD abstract service

The document discard abstract service is used by the DTAM-DM client to discard document manipulations which were done since the use of the last document save, open or discard abstract service.

```
DM-DOCUMENT-DISCARD ::= ABSTRACT-OPERATION
        ARGUMENT                DM-DOCUMENT-DISCARD-Arguments
        RESULT          DM-DOCUMENT-DISCARD-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-DOCUMENT-DISCARD-Arguments ::= SEQUENCE
{
        documentId              [0] DocumentId
}

DM-DOCUMENT-DISCARD-Result ::= NULL
```

#### 9.3.3.1 Arguments

The documentId parameter is mandatory and identifies the selected document for the discard service. The modifications done on the document since the last save, open or discard operation are discarded.

#### 9.3.3.2 Result

If the request succeeds, the DM-DOCUMENT-DISCARD-Result will be returned. There are no parameters.

#### 9.3.3.3 Errors

The reasons for giving an error in the discard service are:

- DM-AccessError: The document is not opened or is reserved by another user or the user has no valid access rights.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The document cannot be discarded due to local reasons.

### 9.3.4 DM-DOCUMENT-CLOSE abstract service

The document close abstract service is used by the DTAM-DM client to close and derelict documents after document manipulations. The modifications which were done since the use of the last DM-DOCUMENT-OPEN, DM-DOCUMENT-SAVE or DM-DOCUMENT-DISCARD abstract service are discarded. To save the document contents before the document will be closed, the DM-DOCUMENT-SAVE abstract service must be used.

```
DM-DOCUMENT-CLOSE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-DOCUMENT-CLOSE-Arguments
        RESULT                  DM-DOCUMENT-CLOSE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-DOCUMENT-CLOSE-Arguments ::= SEQUENCE
{
        documentId              [0] DocumentId
}

DM-DOCUMENT-CLOSE-Result ::= CHOICE
{
        noValue                 NULL,
        permanentId             [0] DocumentId
}
```

#### 9.3.4.1 Arguments

The documentId parameter is mandatory and identifies the selected document for the close service.

#### 9.3.4.2 Result

If the request succeeds, the DM-DOCUMENT-CLOSE-Result carries a permanent identifier, which the server has assigned to the manipulated document. This is only the case, if a nonPermanentIdentifier was used during manipulation (since the DM-DOCUMENT-OPEN). If the permanent identifier was used, no value is returned.

#### 9.3.4.3 Errors

The reasons for giving an error in the close service are:

- DM-AccessError: The document is not opened by the user or is reserved or the user has no valid access rights.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The document cannot be closed due to local reasons.

### 9.3.5 DM-DOCUMENT-LIST abstract service

The document list abstract service is used by the DTAM-DM client to request a list of document identifications from the DTAM-DM server that are in-line with the indicated criteria. The use of the document list service is restricted to applications that have limited filing and retrieval capabilities. For applications that support both, filing and retrieval as well as manipulation, the combined usage of DFR and DTAM-DM is recommended.

```
DM-DOCUMENT-LIST ::= ABSTRACT-OPERATION
        ARGUMENT              DM-DOCUMENT-LIST-Arguments
        RESULT                DM-DOCUMENT-LIST-Result
        ERRORS
        {
                DM-ListError,
                DM-ServiceError
        }

DM-DOCUMENT-LIST-Arguments ::= CHOICE
                {
                odaCriteria                 [0] QueryType,
                otherCriteria               [1] OtherCriteria,
                anyCriteria                 [2] ANY

                }

OtherCriteria ::= CHOICE {
                all                         [0] BOOLEAN,
                prefix                      [1] OCTET STRING,
                suffix                      [2] OCTET STRING }

DM-DOCUMENT-LIST-Result ::= SEQUENCE OF DocumentId
```

#### 9.3.5.1 Arguments

The parameter odaCriteria is a query that consists of a logical expression comprising attributes and their values of an ODA document profile. It is defined in Recommendation T.413.

The parameter OtherCriteria can be applied to any type of document. It can consist of three different selection criteria. The all parameter requires to list all documents. The prefix parameter requires to list all documents with document names started with the same string as the prefix. The suffix parameter requires to list all documents with document names ending with the same string as the suffix.

The parameter anyCriteria requires to list all documents with document names matching the given criteria.

#### 9.3.5.2 Result

As a result, a list of documentIds is returned. The document names of these documentIds fulfill the specified list criteria.

#### 9.3.5.3 Errors

The reasons for giving an error in the list service are:

- DM-ListError: The specified list criteria cannot be accepted.
- DM-ServiceError: The list operation cannot be performed due to local reasons.

### 9.3.6 DM-GET abstract service

The get abstract service is used by the DTAM-DM client to read document fragments.

```
DM-GET ::= ABSTRACT-OPERATION
        ARGUMENT              DM-GET-Arguments
        RESULT                DM-GET-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-GET-Arguments ::= SEQUENCE
{
        object             [0] ManipulationObject
}

DM-GET-Result ::= SEQUENCE OF SEQUENCE OF ObjectContent
```

#### 9.3.6.1 Arguments

The object parameter is mandatory and carries a 'Location-expression' to identify the selected document and document fragment for the get service.

#### 9.3.6.2 Result

Upon successful completion of the get operation, the result parameter returns the content of the object.

#### 9.3.6.3 Errors

The reasons for giving an error in the get service are:

- DM-AccessError: The document is not opened or is reserved by another user.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The get operation cannot be performed due to local reasons.

### 9.3.7 DM-SEARCH abstract service

The search abstract service is used by the DTAM-DM client to search for specified document fragments. The DTAM-DM client specifies search domains (e.g. to search only in the first chapter) and search criteria. The search abstract service may be used to identify document fragments for invocation of subsequent abstract operations like modify or create, among others.

```
DM-SEARCH ::= ABSTRACT-OPERATION
        ARGUMENT            DM-SEARCH-Arguments
        RESULT              DM-SEARCH-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-SEARCH-Arguments ::= SEQUENCE
{       object      [0] ManipulationObject,
        limit       [1] INTEGER OPTIONAL
}

DM-SEARCH-Result ::= SEQUENCE
{
        objectList  [0] SEQUENCE OF DocFragment-Id,
        number      [1] INTEGER
}

DocFragment-Id ::= ConstituentIdentifier
```

#### 9.3.7.1 Arguments

The object parameter is mandatory and carries a 'Location-expression' to identify the selected document and document fragment for the search service. The limit parameter is optional and indicates the maximum number of objects to be returned by the search service.

#### 9.3.7.2 Result

The objectList parameter contains all object identifications found by the search service. The number parameter indicates the number of found objects. If limit was given, number is not greater than limit. If there are more objects which fulfill the search critera, the operation stops.

#### 9.3.7.3 Errors

The reasons for giving an error in the search service are:

- DM-AccessError: The document is not opened or is reserved by another user.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The search operation cannot be performed due to local reasons.

### 9.3.8 DM-CREATE abstract service

The create abstract service is used by the DTAM-DM client to add document fragments to an existing document. The create abstract service may carry the document fragment, including the values applicable to the created document fragment. The DTAM-DM client must specify the position of the additional document fragment inside the document.

```
DM-CREATE ::= ABSTRACT-OPERATION
        ARGUMENT                        DM-CREATE-Arguments
        RESULT                          DM-CREATE-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-CREATE-Arguments ::= SEQUENCE
{
                destination     [0] ManipulationObject,
                position        [1] ObjectPosition,
                content         [2] SEQUENCE OF ObjectContent OPTIONAL
}

DM-CREATE-Result ::= CHOICE
{
        noValue                 [0] NULL,
        assignedID              [1] SEQUENCE OF AssignedId
}
```

#### 9.3.8.1 Arguments

The destination parameter identifies the selected document and document fragment for the create service. It carries a ’Constituent-Locator’. The position parameter identifies the position of the document fragment in the selected document. The content parameter contains the information of the document fragment to be included in the selected document.

In normal cases, all parameters defined in the DM-CREATE-argument are specified. However, if this abstract service is used to create a new paragraph, not inserting document fragment from client to server, but instantiating a subordinate structure by referring to generic structures or defaulting rules specified in ODA document stored in server, the parameter “content” may not be used.

#### 9.3.8.2 Result

The DM-DOCUMENT-CREATE-Result may carry no parameters or the sequence of new identifiers of the fragments.

#### 9.3.8.3 Errors

The reasons for giving an error in the create service are:

- DM-AccessError: The documentId does not indicate an opened document or the user has no valid access rights or the document is reserved by another user.
- DM-FragmentError: The destination parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.

- DM-AttributeError: It is not possible to create the attribute value.
- DM-UpdateError: It is not possible to create a document fragment.
- DM-ServiceError: The document fragment cannot be created due to local reasons.

### 9.3.9 DM-DELETE abstract service

The delete abstract service is used by the DTAM-DM client to delete document fragments[1].

```
DM-DELETE ::= ABSTRACT-OPERATION
        ARGUMENT        DM-DELETE-Arguments
        RESULT          DM-DELETE-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-DELETE-Arguments ::= SEQUENCE
{
        object          [0] ManipulationObject
}

DM-DELETE-Result ::= NULL
```

#### 9.3.9.1 Arguments

The object parameter is mandatory and carries a ’Location-expression’ to identy the selected document and document fragment for the delete service.

#### 9.3.9.2 Result

If the request succeeds, the DM-DELETE-Result will be returned. There are no parameters.

#### 9.3.9.3 Errors

The reasons for giving an error in the delete service are:

- DM-AccessError: The document is not opened or is reserved by another user or the user has no valid access rights.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-AttributeError: It is not possible to delete the attribute value.
- DM-UpdateError: It is not possible to delete the document fragment.
- DM-ServiceError: The document fragment cannot be deleted due to local reasons.

### 9.3.10 DM-MODIFY abstract service

The modify abstract service is used by the DTAM-DM client to modify already existing document fragments. The modify abstract service assigns new values to attributes of the identified document fragment.

[1] In some cases, the DM-DELETE abstract service may require the server to store the deleted document fragments in order to reuse these document fragments. However, it is out of the scope of this Recommendation whether the deleted document fragments are stored or not.

```
DM-MODIFY ::= ABSTRACT-OPERATION
        ARGUMENT                DM-MODIFY-Arguments
        RESULT                  DM-MODIFY-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-MODIFY-Arguments ::= SEQUENCE
{
                object                  [0] ManipulationObject,
                modifications           [1] ModificationsType
}

DM-MODIFY-Result ::= NULL

ModificationsType ::= CHOICE
{
        odaModifications        [0] OdaModifications,
        otherModifications      [1] OCTET STRING,
        anyModifications        [2] ANY
}

OdaModifications ::= SEQUENCE
{
        attributeValue          [0] AttributeValuesType,
        deleting                [1] BOOLEAN DEFAULT FALSE
}
```

#### 9.3.10.1 Arguments

The object parameter is mandatory and carries a 'Location-expression' to identify the selected document and document fragment for the modify service. The modifications parameter contains the new values of the attributes of the identified document fragment.

#### 9.3.10.2 Result

If the request succeeds, the DM-MODIFY-Result will be returned. There are no parameters.

#### 9.3.10.3 Errors

The reasons for giving an error in the modify service are:

- DM-AccessError: The document is not opened or is reserved by another user or the user has no valid access rights.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-AttributeError: It is not possible to modify the attribute value.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- Dm-UpdateError: It is not possible to modify a document fragment.
- DM-ServiceError: The document fragment cannot be modified due to local reasons.

### 9.3.11 DM-COPY abstract service

The copy abstract service is used by the DTAM-DM client to duplicate document fragments to another position of the same or a different document.

```
DM-COPY ::= ABSTRACT-OPERATION
        ARGUMENT            DM-COPY-Arguments
        RESULT              DM-COPY-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-COPY-Arguments ::= SEQUENCE
{
        source              [0] ManipulationObject OPTIONAL,
        destination         [1] ManipulationObject OPTIONAL,
        position            [2] ObjectPosition OPTIONAL
}

DM-COPY-Result ::= SEQUENCE { newConstituent    [0] ManipulationObject }
```

#### 9.3.11.1 Arguments

The source parameter carries a ’Basic-location-expression’ to identify the selected document and document fragment to be copied. The destination parameter identifies the document and document fragment which is the destination of the copy service. It carries a ’Constituent-Locator’. The position parameter identifies the position of the document fragment in the destination document which may be the same or a different document.

In normal cases, all the parameters defined in DM-COPY-arguments should be used. In exceptional cases, the parameters defined in DM-COPY-arguments may be omitted as follows:

– If the DM-COPY-argument is used to extract fragments from the buffer and to insert these document fragments into the target document, the parameter “source” may not be used.

– If the DM-COPY-argument is used to store a document fragment into the buffer with no modifications of the target document, both parameters “destination” and “position” may not be used.

#### 9.3.11.2 Result

If the request succeeds, the DM-COPY-Result will be returned. The parameter “newConstituent” identifies the new object, by use of a 'constituent-identifier'.

#### 9.3.11.3 Errors

The reasons for giving an error in the copy service are:

- DM-AccessError: The source and/or destination documentId is not opened or is reserved by another user or the user has no valid access rights.
- DM-FragmentError: The source and/or destination parameters do not indicate existing document fragments.
- DM-DocumentError: The document-ID does not indicate an exisiting source and/or destination document.
- DM-UpdateError: It is not possible to copy a document fragment.
- DM-AttributeError: It is not possible to copy attribute values.
- DM-ServiceError: The copy operation cannot be performed due to local reasons.

### 9.3.12 DM-MOVE abstract service

The move abstract service is used by the DTAM-DM client to change the position of document fragments inside a document, or from one document to another.

```
DM-MOVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-MOVE-Arguments
        RESULT                  DM-MOVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-MOVE-Arguments ::= SEQUENCE
{
        source                  [0] ManipulationObject,
        destination             [1] ManipulationObject,
        position                [2] ObjectPosition
}

DM-MOVE-Result ::= SEQUENCE
{
                        newConstituent    [0] ManipulationObject
}
```

#### 9.3.12.1 Arguments

The source parameter carries a 'Basic-location-expression' to identify the selected document and document fragment to be moved. The destination parameter identifies the document and document fragment which is the destination of the move service. It carries a 'Constituent-Locator'. The position parameter identifies the position of the document fragment in the destination document which may be the same or a different document.

#### 9.3.12.2 Result

If the request succeeds, the DM-MOVE-Result will be returned. The parameter "newConstituent" identifies the new object by use of a 'Constituent-identifier'.

#### 9.3.12.3 Errors

The reasons for giving an error in the move service are:

- DM-AccessError: The source and/or destination documentId is not opened or is reserved by another user or the user has no valid access rights.
- DM-FragmentError: The source and/or destination parameters do not indicate existing document fragments.
- DM-DocumentError: The document-ID does not indicate an exisiting source and/or destination document.
- DM-AttributeError: It is not possible to move attribute values.
- DM-UpdateError: It is not possible to move a document fragment.
- DM-ServiceError: The move operation cannot be performed due to local reasons.

### 9.3.13 DM-REPLACE abstract service

The replace abstract service is used by the DTAM-DM client to replace document fragments. The replace abstract service effects the deletion of the identified document fragment and the replacement by the new document fragment.

```
DM-REPLACE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-REPLACE-Arguments
        RESULT                  DM-REPLACE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-REPLACE-Arguments ::= SEQUENCE
{
        object          [0] ManipulationObject,
        content         [1] ObjectContent
}

DM-REPLACE-Result ::= NULL
```

### 9.3.13.1 Arguments

The object parameter is mandatory and carries a 'Constituent-Locator' to identify the selected document and document fragment for the replace service. The content parameter is mandatory and contains the new document fragment to be inserted.

### 9.3.13.2 Result

If the request succeeds, the DM-REPLACE-Result will be returned. There are no parameters.

### 9.3.13.3 Errors

The reasons for giving an error in the replace service are:

- DM-AccessError: The document is not opened or is reserved by another user or the user has no valid access rights.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID does not indicate an exisiting document.
- DM-UpdateError: It is not possible to replace a document fragment.
- DM-AttributeError: It is not possible to replace attribute values.
- DM-ServiceError: The replace operation cannot be done due to local reasons.

### 9.3.14 DM-RESERVE abstract service

The reserve abstract service is used by the DTAM-DM client to establish the reservation of the specified document fragment. The reservation service specifies the document fragment as exclusive-access, which means that other users shall not use the DM-GET, DM-DELETE, DM-MODIFY, DM-COPY, DM-MOVE, DM-REPLACE, DM-RESERVE, DM-UNRESERVE, DM-SEARCH and DM-POINT abstract services that refer to the reserved document fragment.

The document fragment is implicitly unreserved by the DM-CLOSE and explicitly by the DM-UNRESERVE operations.

The reservation process does not apply to ownership or access rights of document parts, which may already be defined for document parts. For ODA documents this is handled by the ODA document profile or subprofile (see Recommendation T.414).

```
DM-RESERVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-RESERVE-Arguments
        RESULT                  DM-RESERVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ReservationError,
                DM-ServiceError
        }

DM-RESERVE-Arguments ::= SEQUENCE
{
        object          [0] ManipulationObject
}

DM-RESERVE-Result ::= NULL
```

#### 9.3.14.1 Arguments

The object parameter is mandatory and carries a ’Location-expression’ to identify the selected document and document fragment for the reserve service.

#### 9.3.14.2 Result

Should the request succeed, the DM-RESERVE-Result will be returned. There are no parameters.

#### 9.3.14.3 Errors

The reasons for giving an error in the reserve service are:

- DM-AccessError: The document is not opened by the user or the user has no valid access rights.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ReservationError: The document fragment is reserved.
- DM-ServiceError: The reserve operation cannot be performed due to local reasons.

### 9.3.15 DM-UNRESERVE abstract service

The unreserve abstract service is used by a DTAM-DM client to release a previous reservation of a document fragment done by using the reserve abstract service.

```
DM-UNRESERVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-UNRESERVE-Arguments
        RESULT                  DM-UNRESERVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ReservationError,
                DM-ServiceError
        }

DM-UNRESERVE-Arguments ::= SEQUENCE
{
        object          [0] ManipulationObject
}

DM-UNRESERVE-Result ::= NULL
```

#### 9.3.15.1 Arguments

The object parameter is mandatory and carries a ’Location-expression’ to identify the selected document and document fragment for the unreserve service.

#### 9.3.15.2 Result

If the request succeeds, the DM-UNRESERVE-Result will be returned. There are no parameters.

#### 9.3.15.3 Errors

The reasons for giving an error in the unreserve service are:

- DM-AccessError: The document is not opened by the user or the user has no valid access rights.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ReservationError: The document fragment has not been reserved.
- DM-ServiceError: The unreserve operation cannot be performed due to local reasons.

### 9.3.16 DM-POINT abstract service

The point abstract service is used by DTAM-DM clients in multi-user applications like joint-editing or joint-viewing.

```
DM-POINT ::= ABSTRACT-OPERATION
        ARGUMENT            DM-POINT-Arguments
        RESULT              DM-POINT-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ServiceError
        }

DM-POINT-Arguments ::= SEQUENCE
{
                        object          [0] ManipulationObject
}

DM-POINT-Result ::= NULL
```

#### 9.3.16.1 Arguments

The object parameter is mandatory and identifies the selected document and document fragment for the point service.

#### 9.3.16.2 Result

If the request succeeds, the DM-POINT-Result will be returned. There are no parameters.

#### 9.3.16.3 Errors

The reasons for giving an error in the point service are:

- DM-AccessError: The document is not opened or is reserved by another user.
- DM-FragmentError: The object parameter does not indicate an existing document fragment.
- DM-DocumentError: The document-ID parameter does not indicate an existing document.
- DM-ServiceError: The point operation cannot be performed due to local reasons.

### 9.3.17 DM-MACRO-CALL abstract service

This is for further study.

```
DM-MACRO-CALL ::= ABSTRACT-OPERATION
        ARGUMENT                DM-MACRO-CALL-Arguments
        RESULT                  DM-MACRO-CALL-Result
        ERRORS
        {
                DM-ServiceError
        }

DM-MACRO-CALL-Arguments ::= NULL

DM-MACRO-CALL-Result ::= NULL
```

#### 9.3.17.1 Arguments

None.

#### 9.3.17.2 Result

None.

#### 9.3.17.3 Errors

The reasons for giving an error in the abstract service are:

- DM-ServiceError: The operation cannot be performed due to local reasons.

### 9.3.18 DM-GROUP-BEGIN abstract service

The group-begin abstract service is used by the DTAM-DM client to indicate the start of a set of DTAM-DM abstract services which are to be processed and responded to as a group.

This is for further study.

```
DM-GROUP-BEGIN ::= ABSTRACT-OPERATION
        ARGUMENT                DM-GROUP-BEGIN-Arguments
        RESULT                  DM-GROUP-BEGIN-Result
        ERRORS
        {
                DM-ServiceError
        }

DM-GROUP-BEGIN-Arguments ::=NULL

DM-GROUP-BEGIN-Result ::= NULL
```

#### 9.3.18.1 Arguments

None.

#### 9.3.18.2 Result

None.

#### 9.3.18.3 Errors

The reasons for giving an error in the abstract service are:

- DM-ServiceError: The group-begin operation cannot be performed due to local reasons.

### 9.3.19 DM-GROUP-END abstract service

The group-end abstract service is used by the DTAM-DM client to indicate the end of a set of DTAM-DM abstract services which are to be processed and responded to as a group.

This is for further study.

```
DM-GROUP-END ::= ABSTRACT-OPERATION
          ARGUMENT                  DM-GROUP-END-Arguments
          RESULT                    DM-GROUP-END-Result
          ERRORS
          {
                    DM-ServiceError
          }
```

```
DM-GROUP-END-Arguments ::=NULL
```

```
DM-GROUP-END-Result ::= NULL
```

#### 9.3.19.1 Arguments

None.

#### 9.3.19.2 Result

None.

#### 9.3.19.3 Errors

The reasons for giving an error in the abstract service are:

- DM-ServiceError: The group-end operation cannot be performed due to local reasons.

## 9.4 Definitions of DTAM-DM abstract errors

The abstract errors that can be reported in response to the invocation of abstract operations at the DTAM-DM port are defined and described in this subclause.

If any error occurs during the course of an abstract operation invocation, the DTAM-DM server assures that nothing has been changed by the abstract operation.

The list of errors specified in this subclause are the following:

- DM-AccessError;
- DM-AttributeError;
- DM-UpdateError;
- DM-DocumentError;
- DM-FragmentError;
- DM-ListError;
- DM-ReservationError.
- DM-ServiceError

Unidentified errors are reported by the relevant parameter of the DM-Service-Error.

### 9.4.1 DM-access-error

A DM-AccessError reports a problem encountered when attempting to access a document or document fragment specified in the argument of the abstract operation.

```
DM-AccessError ::= ABSTRACT-ERROR
                  PARAMETER CHOICE {
                            docAccessError       [0] DocAccessError,
                            fragAccessError      [1] FragAccessError }
```

```
DocAccessError ::= SEQUENCE {
                  documentId                     [0] DocumentId,
                  dmDocAccProblem                [1] DmDocAccProblem }
```

```
DmDocAccProblem ::= ENUMERATED {
                  document-already-open          (1),
                  document-not-opened            (2),
                  improper-access-rights         (3) }
```

a) **document-already-open**: The document to be opened has already been opened by the same user;

b) **document-not-opened**: The document of the document fragment to be modified, closed or discarded has not been opened yet;

c) **improper-access-rights**: An attempt to access the document has been made by a DTAM-DM client with insufficient access rights to this entry. If a DTAM-DM client has no read access right to the document a DM-DocumentError will be reported in order to hide the existence of that document.

```
FragAccessError ::= SEQUENCE {
                  entry                     [0] ManipulationObject,
                  dmFragAccProblem          [1] DmFragAccProblem }

DmFragAccProblem ::= ENUMERATED {
                  reserved-by-a-user        (1),
                  improper-access-rights    (2) }
```

d) **reserved-by-a-user**: The document fragment to be modified is at present reserved by another user;

e) **improper-access-rights**: An attempt to access the document fragment has been made by a DTAM-DM client with insufficient access rights to this entry. If a DTAM-DM client has no read access right to the document fragment, a DM-DocumentError will be reported in order to hide the existence of that document.

### 9.4.2 DM-attribute-error

A DM-AttributeError reports a problem encountered while attempting to modify attributes of document fragments.

```
DM-AttributeError ::= ABSTRACT-ERROR
                  PARAMETER SEQUENCE {
                       entry          [0] ManipulationObject,
                       problems       [1] SEQUENCE OF SEQUENCE {
                                      problem     [0] DmAttributeProblem,
                                      type        [1] TypeOfAttributeValue OPTIONAL } }

DmAttributeProblem ::= ENUMERATED {
                  invalid-syntax            (1),
                  constraint-violation      (2),
                  illegal-violation         (3) }

TypeOfAttributeValue ::= CHOICE {
                  odaAttributeValue         [0] AttributeValuesType,
                  anyAttributeValue         [1] ANY }
```

The problems parameter specifies one or several attribute problems encountered. Each problem (identified below) is optional accompanied by the value of the attribute.

a) **invalid-syntax**: A value, specified in the parameter ModificationsType in the argument of the abstract operation, does not conform to the abstract syntax of the type of the attribute;

b) **constraint-violation**: An attribute value supplied in the argument of the abstract operation does not conform to constraints imposed by the standard or document application profile (e.g. the value exceeds the maximum size allowed);

c) **illegal-violation**: An attempt was made to modify an attribute which may not be modified by an abstract operation. For ODA documents, the attributes “identifier,” “subordinates” and “content portions” of a constituent shall not be modified (see 7.3.3/T.413).

### 9.4.3 DM-update-error

A DM-UpdateError reports a problem encountered while attempting to modify a document fragment. Deletion, creation, moving, copying and replacing document fragments are also considered here as modifications.

```
DM-UpdateError ::= ABSTRACT-ERROR
                  PARAMETER SEQUENCE {
                  entry                     [0] ManipulationObject,
                  problem                   [1] DmUpdateProblem }

DmUpdateProblem ::= ENUMERATED { illegal-modification        (1) }
```

A DmUpdateProblem shall be one of the following:

a) **illegal-modification**: An attempt has been made to modify a document fragment which would result in an inconsistent state of the document (e.g. to create a logical object in the layout structure of an ODA document).

### 9.4.4 DM-document-error

A DM-DocumentError reports a problem related to the name of a document specified in the argument of an abstract operation. The document which caused a problem is reported as it was specified, accompanied by an indication of the problem encountered. If a DTAM-DM client has no read access to the document, a DM-DocumentError will be reported in order to hide the existence of that document.

```
DM-DocumentError ::= ABSTRACT-ERROR
            PARAMETER SEQUENCE {
            entry                    [0] DocumentId,
            problem                  [1] DmDocumentProblem }

DmDocumentProblem ::= ENUMERATED {
            invalid-upi                         (1),
            invalid-path-name                   (2),
            ambiguous-path-name                 (3),
            invalid-document-reference          (4),
            invalid-document-name               (5),
            invalid-non-permanent-id            (6) }
```

a) i**nvalid-upi**: The UPI provided in the abstract operation’s argument does not refer to any document in the document store;

b) **invalid-path-name**: The DfrPathName (either absolute or relative) provided in the abstract operations’s argument does not correspond to any existing document in the document store;

c) **ambiguous-path-name**: The DfrEntryName provided in the abstract operation’s argument is ambiguous, that is, corresponds to more than one document;

d) **invalid-document-reference**: The Document-Reference provided in the abstract operation’s argument does not correspond to any exisiting document in the document store;

e) **invalid-document-name**: The DocumentName does not refer to any document in the document store;

f) **invalid-non-permant-id**: The non-permanent-identifier provided in the abstract operation’s argument does not refer to any document in the document store.

### 9.4.5 DM-fragment-error

The DM-FragmentError reports a problem related to the name of the document fragment or the value of the location expression specified in the argument of an abstract operation. The ObjectValue which caused a problem is reported as it was specified, accompanied by an indication of the problem encountered.

```
DM-FragmentError ::= ABSTRACT-ERROR
           PARAMETER SEQUENCE {
           entry                    [0] ObjectValue,
           problem                  [1] DmFragmentProblem }

DmFragmentProblem ::= ENUMERATED {
           invalid-location-expression          (1),
           invalid-expression                   (2) }
```

a) **invalid-location-expression**: The Location-expression provided in the abstract operation’s argument does not refer to a document fragment;

b) **invalid-expression**: The otherExpression or anyexpression provided in the abstract operation’s argument does not refer to a document fragment.

### 9.4.6 DM-list-error

The DM-ListError reports a problem related to the query of the documents available in the document store. The type of query which caused a problem is reported as it was specified.

```
DM-ListError ::= ABSTRACT-ERROR
                PARAMETER ListProblem

ListProblem ::= CHOICE {
                all                [0] BOOLEAN,
                prefix             [1] OCTET STRING,
                suffix             [2] OCTET STRING,
                typeOfQuery        [3] QueryType,
                anyCriteria        [4] ANY }
```

### 9.4.7 DM-reservation-error

The DM-ReservationError reports a problem occurring when an attempt has been made to reserve or unreserve a document fragment.

```
DM-ReserveError ::= ABSTRACT-ERROR
                PARAMETER SEQUENCE {
                    entry          [0] ManipulationObject,
                    problem        [1] DmReservationProblem }

DmReservationProblem ::= ENUMERATED {
                already-reserved   (1),
                not-yet-reserved   (2) }
```

The DmReservationProblem shall be one of the following:

a) **already-reserved**: An attempt has been made to reserve or unreserve a document fragment which has already been reserved by another user;

b) **not-yet-reserved**: An attempt has been made to unreserve a document fragment which is not reserved at the moment.

### 9.4.8 DM-service-error

The DM-ServiceError reports a problem related to the provision of the service, which is not due to an incorrect abstract operation request or the requestor’s access rights.

```
DM-ServiceError ::= ABSTRACT-ERROR
               PARAMETER DmServiceProblem

DmServiceProblem ::= ENUMERATED {
               server-busy                 (1),
               server-unavailable          (2),
               resource-limit-exceeded     (3),
               operation-too-complex       (4),
               unclassified-server-error   (5) }
```

The DmServiceProblem shall be one of the following:

a) **server-busy**: The DTAM-DM server is presently too busy to perform the requested abstract operation, but may be able to do so after a short while;

b) **server-unavailable**: The DTAM-DM server is currently unavailable. The proposed operation is to unbind DTAM-DM server and DTAM-DM client;

c) **resource-limit-exceeded**: The resources of the DTAM-DM server can be exceeded for example when a very large document fragment is to be created or copied in the document store;

d) **operation-too-complex**: The requested abstract operation is too complex syntactically or semantically (e.g. the location-expression to determine a document fragment cannot be dissolved);

e) **unclassified-server-error**: An error which cannot be categorized in any other way. The reason for this error is implementation specific.

### 9.4.9 Error precedence

Should several error conditions occur simultaneously for the same DTAM-DM abstract operation, only one of them is reported to the requestor. The precedence of these error conditions is as follows beginning with:

- DM-ServiceError;
- DM-DocumentError;
- DM-FragmentError;
- DM-AccessError;
- DM-ReservationError;
- DM-AttributeError;
- DM-UpdateError;
- DM-ListError.

## 9.5 Definition of DTAM-TK abstract services

This Recommendation defines the following DTAM-TK abstract services:

- TK-TOKEN-GIVE;
- TK-TOKEN-PLEASE.

These DTAM-TK abstract services are defined in the following subclauses.

### 9.5.1 TK-TOKEN-GIVE abstract service

The token-give abstract service surrenders the application token from a DTAM-TK client/server, who temporarily acts as a client, to a DTAM-TK client/server, who temporarily acts as a server. This abstract operation changes the roles of the DTAM-TK client/server objects in the communication process (from client to server and vice versa).

```
TK-TOKEN-GIVE ::= ABSTRACT-OPERATION
        ARGUMENT            TK-TOKEN-GIVE-Arguments
        RESULT              TK-TOKEN-GIVE-Result
        ERRORS
        {
                TK-ServiceError
        }

TK-TOKEN-GIVE-Arguments ::= NULL

TK-TOKEN-GIVE-Result ::= NULL
```

#### 9.5.1.1 Arguments

None.

#### 9.5.1.2 Result

If the request succeeds, the TK-TOKEN-GIVE-Result will be returned. There are no parameters.

#### 9.5.1.3 Errors

The reasons for giving an error in the abstract service are:

- TK-ServiceError: The token-give operation cannot be performed due to local reasons.

### 9.5.2 TK-TOKEN-PLEASE abstract service

The token-please abstract service is used by the DTAM-TK client/server, who temporarily acts as a server, to request the application token from a DTAM-TK client/server, who temporarily acts as a client. This request changes the role of the application entities in the communication process.

The DTAM-TK client/server responses with the TK-TOKEN-GIVE operation if he accepts to exchange the application token. He does not respond if he refuses. The DTAM-TK client/server may respond immediately after reception of the TK-TOKEN-PLEASE abstract operation or may invoke one or more other DTAM-DM operations before he decides to surrender the token.

```
TK-TOKEN-PLEASE ::= ABSTRACT-OPERATION
        ARGUMENT            TK-TOKEN-PLEASE-Arguments
        RESULT              TK-TOKEN-PLEASE-Result
        ERRORS
        {
                TK-ServiceError
        }

TK-TOKEN-PLEASE-Arguments ::= NULL

TK-TOKEN-PLEASE-Result ::= NULL
```

### 9.5.2.1 Arguments

None.

### 9.5.2.2 Result

If the request succeeds, the TK-TOKEN-PLEASE-Result will be returned. There are no parameters.

### 9.5.2.3 Errors

The reasons for giving an error in the abstract service are:

- TK-ServiceError: The token-please operation cannot be performed due to local reasons.

## 9.6 Definitions of DTAM-TK abstract errors

The DTAM-TK abstract errors that can be reported in response to the invocation of abstract operations at the DTAM-TK port are defined and described in this subclause.

### 9.6.1 TK-service-error

The TK-ServiceError reports a problem related to the provision of the service, which is not due to an incorrect abstract operation request or the requestor’s access rights.

```
TK-ServiceError ::= ABSTRACT-ERROR
            PARAMETER TKServiceProblem

TKServiceProblem ::= ENUMERATED {
            server-busy                  (1),
            server-unavailable           (2),
            unclassified-server-error    (3) }
```

The TKServiceProblem shall be one of the following:

a) **server-busy**: The DTAM-TK server is presently too busy to perform the requested abstract operation, but may be able after some shorter time period;

b) **server-unavailable**: The DTAM-TK server is currently unavailable;

c) **unclassified-server-error**: An error which cannot be categorized in any other way. The reason for this error is implementation specific.

# Annex A

## Formal assignment of object identifiers

(This annex forms an integral part of this Recommendation)

All object identifiers which are used in this Recommendation are formally assigned in this annex using the ASN.1 notation:

```
DTAM-MAN-ObjectIdentifiers { ccitt recommendation t 435 arch(3) modules(0) object-identifiers(0) }
DEFINITIONS ::=
BEGIN
--  Prologue
--  Exports everything
--  Imports nothing

ID ::= OBJECT IDENTIFIER
        id-dtam-dm          ID ::= { ccitt recommendation t 435 dm(1) }     --  Document Manipulation
        id-dtam-tk          ID ::= { ccitt recommendation t 435 tk(2) }     --  Token Handling
        id-arch             ID ::= { ccitt recommendation t 435 arch(3)}    --  Overall Architecture

--  Categories

        id-mod              ID ::= { id-arch 0 }            --  modules

--  Modules

           id-object-identifiers ID ::= { id-mod 0 }

END --  of DTAM-MAN-ObjectIdentifiers


DTAM-DM-ObjectIdentifiers { ccitt recommendation t 435 dm(1) modules(0) object-identifiers(0) }
DEFINITIONS ::=
BEGIN

--  EXPORTS everything

IMPORTS
        ID, id-dtam-dm

        FROM DTAM-MAN-ObjectIdentifiers { ccitt recommendation t 435 arch(3) modules(0) object-identifiers(0) };

-- Categories

        id-mod                      ID ::= { id-dtam-dm 0 }  --  modules
        id-ot                       ID ::= { id-dtam-dm 1 }  --  objects
        id-pt                       ID ::= { id-dtam-dm 2 }  --  ports

--  Modules

        id-mod-object-identifiers   ID ::= { id-mod 0 }
        id-mod-abstract-services    ID ::= { id-mod 1 }

--  Objects

        id-ot-dtam-dm-client        ID ::= { id-ot 0 }
        id-ot-dtam-dm-server        ID ::= { id-ot 1 }
        id-ot-dtam-dm-dfr-client    ID ::= { id-ot 2 }
        id-ot-dtam-dm-dfr-server    ID ::= { id-ot 3 }

--  Ports

        id-pt-dtam-dm-port          ID ::= { id-pt 0 }
        id-pt-dtam-dm-sym-port      ID ::= { id-pt 1 }      --  only for DTAM-DM/TK environment

END --  of DTAM-DM-ObjectIdentifiers
```

```
DTAM-TK-ObjectIdentifiers { ccitt recommendation t 435 tk(2) modules(0) object-identifiers(0) }
DEFINITIONS ::=
BEGIN
--  EXPORTS everything

IMPORTS
          ID, id-dtam-tk

          FROM DTAM-MAN-ObjectIdentifiers { ccitt recommendation t 435 arch(3) modules(0) object-identifiers(0) };

--  Categories

          id-mod                          ID ::= { id-dtam-tk 0 }     --  modules
          id-ot                           ID ::= { id-dtam-tk 1 }     --  objects
          id-pt                           ID ::= { id-dtam-tk 2 }     --  ports

--  Modules

          id-mod-object-identifiers       ID ::= { id-mod 0 }
          id-mod-abstract-services        ID ::= { id-mod 1 }

--  Objects

          id-ot-dtam-dm-tk-client-server  ID ::= { id-ot 0 }

--  Ports

          id-pt-dtam-tk-port              ID ::= { id-pt 0 }

END --  of DTAM-TK-ObjectIdentifiers
```

# Annex B

## Formal definition of DTAM-DM and DTAM-TK abstract services

(This annex forms an integral part of this Recommendation)

All abstract services which are used in this Recommendation are formally defined in this Annex B using the ASN.1 notation:

```
DTAM-DM-AbstractServices { ccitt recommendation t 435 dm(1) modules(0) abstract-services(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--  EXPORTS everything

IMPORTS

          ABSTRACT-BIND, ABSTRACT-UNBIND, ABSTRACT-ERROR, ABSTRACT-OPERATION, OBJECT, PORT

              FROM AbstractServiceNotation
              { joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) }
              --  see Recommendation X.407

id-ot-dtam-dm-client, id-ot-dtam-dm-server,
id-ot-dtam-dm-dfr-client, id-ot-dtam-dm-dfr-server,
id-pt-dtam-dm-port

              FROM DTAM-DM-ObjectIdentifiers
              { ccitt recommendation t 435 dm(1) modules(0) object-identifiers(0) }

Non-Basic-Doc-Characteristics, Non-Basic-Struc-Characteristics

              FROM Document-Profile-Descriptor { 2 8 1 5 6 }
```

```
ConstituentType, AttributeValuesType, QueryType, ConstituentIdentifier

            FROM Abstract-Operations { 2 8 1 3 0 }
            --  see Recommendation T.413

Location-expression, Basic-location-expression, Constituent-locator

            FROM Location-Model { 2 8 1 12 0 }
            --  see ITU-T Recommendation T.422

DistinguishedName

            FROM InformationFramework
            { joint-iso-ccitt ds(5) modules(1) informationFramework(1) }

Document-Reference

            FROM Document-Profile-Descriptor { 2 8 1 5 6 }
            --  see Recommendation T.415

DfrEntryName

            FROM DFRAbstractService { iso standard 10166 part-1(1) modules(0) abstract-service(1) };
            --  see ISO/IEC 10166-1

--  Objects

            dtam-dm-client    OBJECT
                    PORTS { dtam-dm-port [ C ] }
                    ::= id-ot-dtam-dm-client

            dtam-dm-server   OBJECT
                    PORTS { dtam-dm-port [ S ] }
                    ::= id-ot-dtam-dm-server

            dtam-dm-dfr-client    OBJECT
                    PORTS { dtam-dm-port [ C ], dfr-port [ C ] }
                    ::= id-ot-dtam-dm-dfr-client

            dtam-dm-dfr-server    OBJECT
                    PORTS { dtam-dm-port [ S ], dfr-port [ S ] }
                    ::= id-ot-dtam-dm-dfr-server

--  Ports

            dtam-dm-port     PORT
                  CONSUMER INVOKES
                  {
                  DM-DOCUMENT-OPEN,
                  DM-DOCUMENT-SAVE,
                  DM-DOCUMENT-DISCARD,
                  DM-DOCUMENT-CLOSE,
                  DM-DOCUMENT-LIST,
                  DM-GET,
                  DM-SEARCH,
                  DM-CREATE,
                  DM-DELETE,
                  DM-MODIFY,
                  DM-COPY,
                  DM-MOVE,
                  DM-REPLACE,
                  DM-RESERVE,
                  DM-UNRESERVE,
                  DM-POINT,
                  DM-MACRO-CALL,
                  DM-GROUP-BEGIN,
                  DM-GROUP-END
                  }
                    SUPPLIER INVOKES
                  {
                  }
                  ::= id-pt-dtam-dm-port
```

```
--  only in the DTAM-DM/DTAM-TK environment

dtam-dm-sym-port     PORT
          ABSTRACT OPERATIONS {
                  DM-DOCUMENT-OPEN,
                  DM-DOCUMENT-SAVE,
                  DM-DOCUMENT-DISCARD,
                  DM-DOCUMENT-CLOSE,
                  DM-DOCUMENT-LIST,
                  DM-GET,
                  DM-SEARCH,
                  DM-CREATE,
                  DM-DELETE,
                  DM-MODIFY,
                  DM-COPY,
                  DM-MOVE,
                  DM-REPLACE,
                  DM-RESERVE,
                  DM-UNRESERVE,
                  DM-POINT,
                  DM-MACRO-CALL,
                  DM-GROUP-BEGIN,
                  DM-GROUP-END }
                  ::= id-pt-dtam-dm-sym-port

-- Data types

DocumentId ::= CHOICE {
dfrName                                       [0] DfrEntryName,
documentReference                             [1] Document-Reference,
documentName                                  [2] OCTET STRING,
nonPermanentIdentifier                        [3] NonPermanentIdentifier
}

ManipulationObject ::= SEQUENCE {
          documentIdentifier                  [0] DocumentId OPTIONAL,
          objectValue                         [1] ObjectValue OPTIONAL
}

ObjectValue ::= CHOICE
{
          odaExpression                       [0] OdaExpression,
          otherExpression                     [1] OCTET STRING,
          anyExpression                       [2] ANY
}

OdaExpression ::= CHOICE
{
          locationExpression                  [0] LocationExpression,
          basicLocationexpression             [1] BasicLocationExpression,
          constituentLocator                  [2] ConstituentLocator,
          ConstituentIdentifier               [3] ConstituentIdentifier
}

ObjectPosition ::= ENUMERATED {
          before                              (0),
          afterLastChild                      (1)
}

ObjectContent ::= CHOICE {
          odaContent                          [0] ConstituentType,
          octetAligned                        [1] OCTET STRING,
          anyContent                          [2] ANY
}

AssignedId ::= CHOICE {
          odaIdentifier                       [0] ConstituentIdentifier,
          nonOdaIdentifier                    [1] INTEGER,
          any                                 [2] ANY
}
```

```
NonPermanentIdentifier ::= CHOICE
{
                                        [0] INTEGER,
                                        [1] OCTET STRING
}

--  Abstract bind

DTAM-DMBind ::= ABSTRACT-BIND
TO {dtam-dm-port[S], dfr-port[S], dtam-dm-sym-port, dtam-tk-port} --  depending on the environment
BIND
      ARGUMENT      DtamDmBindArgument
      RESULT        DtamDmBindResult
      BIND-ERROR    DtamDmBindError

DtamDmBindArgument ::= [0] SEQUENCE
{
      dtamProtocolVersion              [0] BIT STRING {
                                       version-1 (0)} DEFAULT {version-1},
      dtamManipulationCapabilities     [1] ManipulationCapabilities,
      odaApplicationCapabilities       [2] SEQUENCE OF OdaApplicationCapabilities OPTIONAL,
      dfrCapablities                   [3] DfrCapabilities OPTIONAL,
      bilateralInformation             [4] SEQUENCE OF BilateralInformation OPTIONAL,
      applicationRequirements          [5] ApplicationRequirements OPTIONAL
}

OdaApplicationCapabilities ::= SEQUENCE
{
      odaDocumentApplicationProfile            [0] OBJECT IDENTIFIER,
      nonBasicOdaDocumentCharacteristics       [1] Non-Basic-Doc-Characteristics OPTIONAL,
      nonBasicOdaStructuralCharacteristics     [2] Non-Basic-Struc-Characteristics OPTIONAL
}

ManipulationCapabilities ::= CHOICE
{
      profileSelection                 [0] OBJECT IDENTIFIER,
      manipulationLevelSelection       [1] ENUMERATED {
                                       basicReadOnlyLevel         (0),
                                       basicManipulationLevel     (1),
                                       extendedLevel              (2) },
      bilateralAgreement               [2] IMPLICIT OCTET STRING }

DfrCapabilities ::= SEQUENCE {
      dfrProtocolVersion               [0] DfrProtocolVersion DEFAULT {version-1},
      dfrProfile                       [1] OBJECT IDENTIFIER OPTIONAL }

DfrProtocolVersion ::= BIT STRING
{
               version-1 (1),
               version-2 (2)
}

BilateralInformation ::= SEQUENCE
{
      serverName                       [0] ServerName,
      information                      [1] EXTERNAL
}

ServerName ::= DistinguishedName

ApplicationRequirement ::= CHOICE
{
         [0] OCTET STRING,
         [1] OBJECT IDENTIFIER
}
```

```
DtamDmBindResult ::= [1] SEQUENCE
{
        serverSelectedDtamProtocolVersion     [0] BIT STRING
                                              { version-1 (0) } DEFAULT {version-1},
        odaApplicationCapabilities            [1] SEQUENCE OF OdaApplicationCapabilities,
        serverSelectedDfrVersion              [2] BIT STRING
                                              { version-1 (1),
                                                version-2 (2) } DEFAULT {version-1},
        bilateralInformation                  [3] SEQUENCE OF BilateralInformation OPTIONAL }

DtamDmBindError ::= [2] CHOICE
{
        serviceError        [0] DmServiceProblem,
        protocolError       [1] ProtocolProblem,
        profileEror         [2] ProfileProblem,
        applicationError    [3] ApplicationProblem
}

ProtocolProblem ::= SEQUENCE
{
        protocol-version-not-supported      [0] SEQUENCE OF ENUMERATED {
                                            dtam-version-not-supported  (0),
                                            dfr-version-not-supported   (1) },
        supportedDtamProtocolVersions       [1] BIT STRING {
                                            version-1 (0) } OPTIONAL,
        supportedDfrProtocolVersions        [2] BIT STRING {
                                            version-1 (0),
                                            version-2 (1) } OPTIONAL
}

ProfileProblem ::= SEQUENCE
{
        profile-not-supported               [0] SEQUENCE OF ENUMERATED {
                                            dtam-manipulation-level-not-supported (0),
                                            dfr-profile-not-supported (1) },
        supportedDtamManipulationLevels     [1] SEQUENCE OF ManipulationCapabilities OPTIONAL,
        supportedDfrProfiles                [2] SEQUENCE OF OBJECT IDENTIFIER OPTIONAL
}

ApplicationProblem ::= SEQUENCE OF ApplicationRequirements OPTIONAL

DTAM-DMUnBind ::= ABSTRACT-UNBIND
FROM {dtam-dm-port[S], dfr-port[S], dtam-dm-sym-port, dtam-tk-port}
        --  depending on the environment

DTAM-DMAbortInfo ::= [3] SEQUENCE
{
        abortSource             [0] DtamDmAbortSource,
        abortReason             [1] DtamDmAbortReason OPTIONAL,
        userInformation         [2] EXTERNAL OPTIONAL }

DtamDmAbortSource ::= ENUMERATED
{
        dtamDmServiceUser       (0),
        dtamDmServiceProvider   (1)
}

DtamDmAbortReason ::= ENUMERATED
{
        local-system-problem    (0),
        temporary-problem       (1),
        protocol-error          (2)
}
```

```
--  Abstract operations

DM-DOCUMENT-OPEN ::= ABSTRACT-OPERATION
        ARGUMENT                        DM-DOCUMENT-OPEN-Arguments
        RESULT                          DM-DOCUMENT-OPEN-Result
        ERRORS {
                DM-DocumentError,
                DM-AccessError,
                DM-ServiceError
        }

DM-DOCUMENT-OPEN-Arguments ::= SEQUENCE
{
        documentId          [0] DocumentId
        mode                [1] ModeType OPTIONAL
}

ModeType ::= ENUMERATED
{
        read                (0),
        modify              (1),
        create              (2)
}

DM-DOCUMENT-OPEN-Result ::= CHOICE {
        noValue                     [0] NULL,
        nonPermanentIdentifier      [1] NonPermanentIdentifier
}

DM-DOCUMENT-SAVE ::= ABSTRACT-OPERATION
        ARGUMENT                    DM-DOCUMENT-SAVE-Arguments
        RESULT                      DM-DOCUMENT-SAVE-Result
        ERRORS {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-DOCUMENT-SAVE-Arguments ::= SEQUENCE
{
        documentId                  [0] DocumentId
}

DM-DOCUMENT-SAVE-Result ::= NULL

DM-DOCUMENT-DISCARD ::= ABSTRACT-OPERATION
        ARGUMENT                    DM-DOCUMENT-DISCARD-Arguments
        RESULT                      DM-DOCUMENT-DISCARD-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-DOCUMENT-DISCARD-Arguments ::= SEQUENCE
{
        documentId              [0] DocumentId
}

DM-DOCUMENT-DISCARD-Result ::= NULL

DM-DOCUMENT-CLOSE ::= ABSTRACT-OPERATION
        ARGUMENT                    DM-DOCUMENT-CLOSE-Arguments
        RESULT                      DM-DOCUMENT-CLOSE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-ServiceError
        }
```

```
DM-DOCUMENT-CLOSE-Arguments ::= SEQUENCE
{
        documentId      [0] DocumentId
}

DM-DOCUMENT-CLOSE-Result ::= CHOICE
{
        noValue         NULL,
        permanentId     [0] DocumentId
}

DM-DOCUMENT-LIST ::= ABSTRACT-OPERATION
        ARGUMENT                DM-DOCUMENT-LIST-Arguments
        RESULT                  DM-DOCUMENT-LIST-Result
        ERRORS
        {
                DM-ListError,
                DM-ServiceError
        }

DM-DOCUMENT-LIST-Arguments ::= CHOICE
        {
        odaCriteria             [0] QueryType,
        otherCriteria           [1] OtherCriteria,
        anyCriteria             [2] ANY
        }

OtherCriteria ::= CHOICE {
        all                     [0] BOOLEAN,
        prefix                  [1] OCTET STRING,
        suffix                  [2] OCTET STRING }

DM-DOCUMENT-LIST-Result ::= SEQUENCE OF DocumentId

DM-GET ::= ABSTRACT-OPERATION
        ARGUMENT        DM-GET-Arguments
        RESULT          DM-GET-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-GET-Arguments::= SEQUENCE
{
        object      [0] ManipulationObject
}

DM-GET-Result ::= SEQUENCE OF SEQUENCE OF ObjectContent

DM-SEARCH ::= ABSTRACT-OPERATION
        ARGUMENT                DM-SEARCH-Arguments
        RESULT                  DM-SEARCH-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-ServiceError
        }

DM-SEARCH-Arguments ::=SEQUENCE
{
        object                  [0] ManipulationObject,
        limit                   [1] INTEGER OPTIONAL
}
```

```
DM-SEARCH-Result ::=SEQUENCE
{
        objectList              [0] SEQUENCE OF DocFragment-Id,
        number                  [1] INTEGER
}

DocFragment-Id ::= ObjectContent

DM-CREATE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-CREATE-Arguments
        RESULT                  DM-CREATE-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-CREATE-Arguments ::= SEQUENCE
{
        destination             [0] ManipulationObject,
        position                [1] ObjectPosition,
        content                 [2] SEQUENCE OF ObjectContent OPTIONAL
}

DM-CREATE-Result ::= CHOICE
{
        noValue                 [0] NULL,
        assignedID              [1] SEQUENCE OF AssignedId
}

DM-DELETE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-DELETE-Arguments
        RESULT                  DM-DELETE-Result
        ERRORS
        {
                DM-AccessError,
                DM-FragmentError,
                DM-DocumentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-DELETE-Arguments ::= SEQUENCE
{
        object                  [0] ManipulationObject
}

DM-DELETE-Result ::= NULL

DM-MODIFY ::= ABSTRACT-OPERATION
        ARGUMENT                DM-MODIFY-Arguments
        RESULT                  DM-MODIFY-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
           }
```

```
DM-MODIFY-Arguments ::= SEQUENCE
{
          object                        [0] ManipulationObject,
          modifications                 [1] ModificationsType
}

DM-MODIFY-Result ::= NULL

ModificationsType ::= CHOICE
{
          odaModifications              [0] OdaModifications,
          otherModifications            [1] OCTET STRING,
          anyModifications              [2] ANY
}

OdaModifications ::= SEQUENCE
{
          attributeValue                [0] AttributeValuesType,
          deleting                      [1] BOOLEAN DEFAULT FALSE
}

DM-COPY ::= ABSTRACT-OPERATION
          ARGUMENT                      DM-COPY-Arguments
          RESULT                        DM-COPY-Result
          ERRORS
          {
                    DM-AccessError,
                    DM-DocumentError,
                    DM-FragmentError,
                    DM-AttributeError,
                    DM-UpdateError,
                    DM-ServiceError
              }

DM-COPY-Arguments ::= SEQUENCE
{
          source                        [0] ManipulationObject OPTIONAL,
          destination                   [1] ManipulationObject OPTIONAL,
          position                      [2] ObjectPosition OPTIONAL
}

DM-COPY-Result ::= SEQUENCE
{
          newConstituent                [0] ManipulationObject
}

DM-MOVE ::= ABSTRACT-OPERATION
          ARGUMENT                      DM-MOVE-Arguments
          RESULT                        DM-MOVE-Result
          ERRORS {
                    DM-AccessError,
                    DM-DocumentError,
                    DM-FragmentError,
                    DM-AttributeError,
                    DM-UpdateError,
                    DM-ServiceError
          }

DM-MOVE-Arguments ::= SEQUENCE
{
          source                        [0] ManipulationObject,
          destination                   [1] ManipulationObject,
          position                      [2] ObjectPosition
}
```

```
DM-MOVE-Result ::= SEQUENCE
{
        newConstituent          [0] ManipulationObject
}

DM-REPLACE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-REPLACE-Arguments
        RESULT                  DM-REPLACE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-AttributeError,
                DM-UpdateError,
                DM-ServiceError
        }

DM-REPLACE-Arguments ::= SEQUENCE
{
        object                  [0] ManipulationObject,
        content                 [1] ObjectContent
}

DM-REPLACE-Result ::= NULL

DM-RESERVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-RESERVE-Arguments
        RESULT                  DM-RESERVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ReservationError,
                DM-ServiceError
        }

DM-RESERVE-Arguments ::= SEQUENCE
{
        object                  [0] ManipulationObject
}

DM-RESERVE-Result ::= NULL

DM-UNRESERVE ::= ABSTRACT-OPERATION
        ARGUMENT                DM-UNRESERVE-Arguments
        RESULT                  DM-UNRESERVE-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ReservationError,
                DM-ServiceError
        }

DM-UNRESERVE-Arguments ::= SEQUENCE
{
        object                  [0] ManipulationObject
}
```

```
DM-UNRESERVE-Result ::= NULL

DM-POINT ::= ABSTRACT-OPERATION
        ARGUMENT                DM-POINT-Arguments
        RESULT                  DM-POINT-Result
        ERRORS
        {
                DM-AccessError,
                DM-DocumentError,
                DM-FragmentError,
                DM-ServiceError
        }

DM-POINT-Arguments ::= SEQUENCE
{
        object          [0] ManipulationObject
}


DM-POINT-Result ::= NULL

--  Abstract errors

DM-AccessError ::= ABSTRACT-ERROR
PARAMETER CHOICE {
        docAccessError                  [0] DocAccessError,
        fragAccessError                 [1] FragAccessError }

DocAccessError ::= SEQUENCE {
        documentId                      [0] DocumentId,
        dmDocAccProblem                 [1] DmDocAccProblem }

DmDocAccProblem ::= ENUMERATED {
        document-already-open           (1),
        document-not-opened             (2),
        improper-access-rights          (3) }

FragAccessError ::= SEQUENCE {
        entry                           [0] ManipulationObject,
        dmFragAccProblem                [1] DmFragAccProblem }

DmFragAccProblem ::= ENUMERATED {
        reserved-by-a-user              (1),
        improper-access-rights          (2) }

DM-AttributeError ::= ABSTRACT-ERROR
PARAMETER SEQUENCE {
        entry      [0] ManipulationObject,
        problems   [1] SEQUENCE OF SEQUENCE {
        problem    [0] DmAttributeProblem,
             type           [1] TypeOfAttributeValue OPTIONAL } }

DmAttributeProblem ::= ENUMERATED {
                        invalid-syntax          (1),
                        constraint-violation    (2),
                        illegal-violation       (3) }

TypeOfAttributeValue ::= CHOICE {
        odaAttributeValue               [0] AttributeValuesType,
        anyAttributeValue               [1] ANY }

DM-UpdateError ::= ABSTRACT-ERROR
PARAMETER SEQUENCE {
        entry                   [0] ManipulationObject,
        problem                 [1] DmUpdateProblem }

DmUpdateProblem ::= ENUMERATD { illegal-modification        (1) }
```

```
DM-DocumentError ::= ABSTRACT-ERROR
PARAMETER SEQUENCE {
          entry                     [0] DocumentId,
          problem                   [1] DmDocumentProblem }

DmDocumentProblem ::= ENUMERATED {
                         invalid-upi                     (1),
                         invalid-path-name               (2),
                         ambiguous-path-name             (3),
                         invalid-document-reference      (4),
                         invalid-document-name           (5),
                         invalid-non-permanent-id        (6) }

DM-FragmentError ::= ABSTRACT-ERROR
PARAMETER SEQUENCE {
          entry                     [0] ObjectValue,
          problem                   [1] DmFragmentProblem }

DmFragmentProblem ::= ENUMERATED {
                         invalid-location-expression     (1),
                         invalid-expression              (2) }

DM-ListError ::= ABSTRACT-ERROR
PARAMETER ListProblem

ListProblem ::= CHOICE {
          all                       [0] BOOLEAN,
          prefix                    [1] OCTET STRING,
          suffix                    [2] OCTET STRING,
          typeOfQuery               [3] QueryType,
          anyCriteria               [4] ANY }

DM-ReserveError ::= ABSTRACT-ERROR
PARAMETER SEQUENCE {
          entry                     [0] ManipulationObject,
          problem                   [1] DmReservationProblem }

DmReservationProblem ::= ENUMERATED {
                             already-reserved      (1),
                             not-yet-reserved      (2) }

DM-ServiceError ::= ABSTRACT-ERROR
PARAMETER DmServiceProblem

DmServiceProblem ::= ENUMERATED
{
                      server-busy                (1),
                      server-unavailable         (2),
                      resource-limit-exceeded    (3),
                      operation-too-complex      (4),
                      unclassified-server-error  (5)
}

END -- of DTAM-DM-AbstractServices

DTAM-TK-AbstractServices { ccitt recommendation t 435 tk(2) modules(0) abstract-services(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything

IMPORTS

          ABSTRACT-ERROR, ABSTRACT-OPERATION, OBJECT, PORT

              FROM AbstractServiceNotation
              { joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) }
```

```
id-pt-dtam-dm-sym-port

        FROM DTAM-DM-AbstractServices
        { ccitt recommendation t 435 dm(1) modules(0) abstract-services(1) }

id-pt-dtam-tk-port,
id-ot-dtam-dm-tk-client-server

        FROM DTAM-TK-ObjectIdentifiers
        { ccitt recommendation t 435 tk(2) modules(0) object-identifiers(0) };


--  Objects

        dtam-dm-tk-client-server        OBJECT
                                        PORTS { dtam-dm-sym-port, dtam-tk-port  }
                                        ::= id-ot-dtam-dm-tk-client-server


--  Ports

dtam-tk-port        PORT
ABSTRACT OPERATIONS {
      TK-TOKEN-GIVE,
      TK-TOKEN-PLEASE }
::= id-pt-dtam-tk-port


-- Abstract operations

TK-TOKEN-GIVE ::= ABSTRACT-OPERATION
      ARGUMENT            TK-TOKEN-GIVE-Arguments
      RESULT              TK-TOKEN-GIVE-Result
      ERRORS
      {
            TK-ServiceError
      }

TK-TOKEN-GIVE-Arguments ::= NULL

TK-TOKEN-GIVE-Result ::= NULL

TK-TOKEN-PLEASE ::= ABSTRACT-OPERATION
      ARGUMENT              TK-TOKEN-PLEASE-Arguments
      RESULT                TK-TOKEN-PLEASE-Result
      ERRORS
      {
            TK-ServiceError
      }

TK-TOKEN-PLEASE-Arguments ::= NULL

TK-TOKEN-PLEASE-Result ::= NULL


--  Abstract errors

TK-ServiceError ::= ABSTRACT-ERROR
          PARAMETER TKServiceProblem

TKServiceProblem ::= ENUMERATED {
                server-busy                 (1),
                server-unavailable          (2),
                unclassified-server-error   (3) }

END --  of DTAM-TK-AbstractServices
```

# Appendix I

## Basic procedures for the use of DTAM-DM abstract services

(This appendix does not form an integral part of this Recommendation)

The following basic procedures are considered for the use of the DTAM-DM abstract services:

### I.1 DTAM association establishment

The DTAM-DM client creates a DTAM association to the DTAM-DM server using the ACSE A-ASSOCIATE service to establish a communication link. During this establishment, the DTAM server exchanges its application capabilities including the manipulation capabilities. These capabilities indicated from the server are interpreted as the server's receiving capabilities, and are not recognized as capabilitiy negotation.

### I.2 Document open

Before manipulating document fragments of a document, the DTAM-DM client must open the document using the document open abstract service. The DTAM-DM client can open more than one document within a DTAM association.

### I.3 Document manipulations

The DTAM-DM client can manipulate document fragments of documents which have been opened before using convenient DM-* abstract services. Furthermore, the DTAM-DM client can manipulate among more than one document, e.g. copying a part of one document into another document.

### I.4 Document save

After a sequence of document fragment manipulations, the DTAM-DM client can save this document(s) to register the result of this sequence of manipulations.

### I.5 Document discard

After a sequence of document fragment manipulations, the DTAM-DM client can undo this sequence of manipulations using the document discard service. All manipulations are discarded until the previous document save operation.

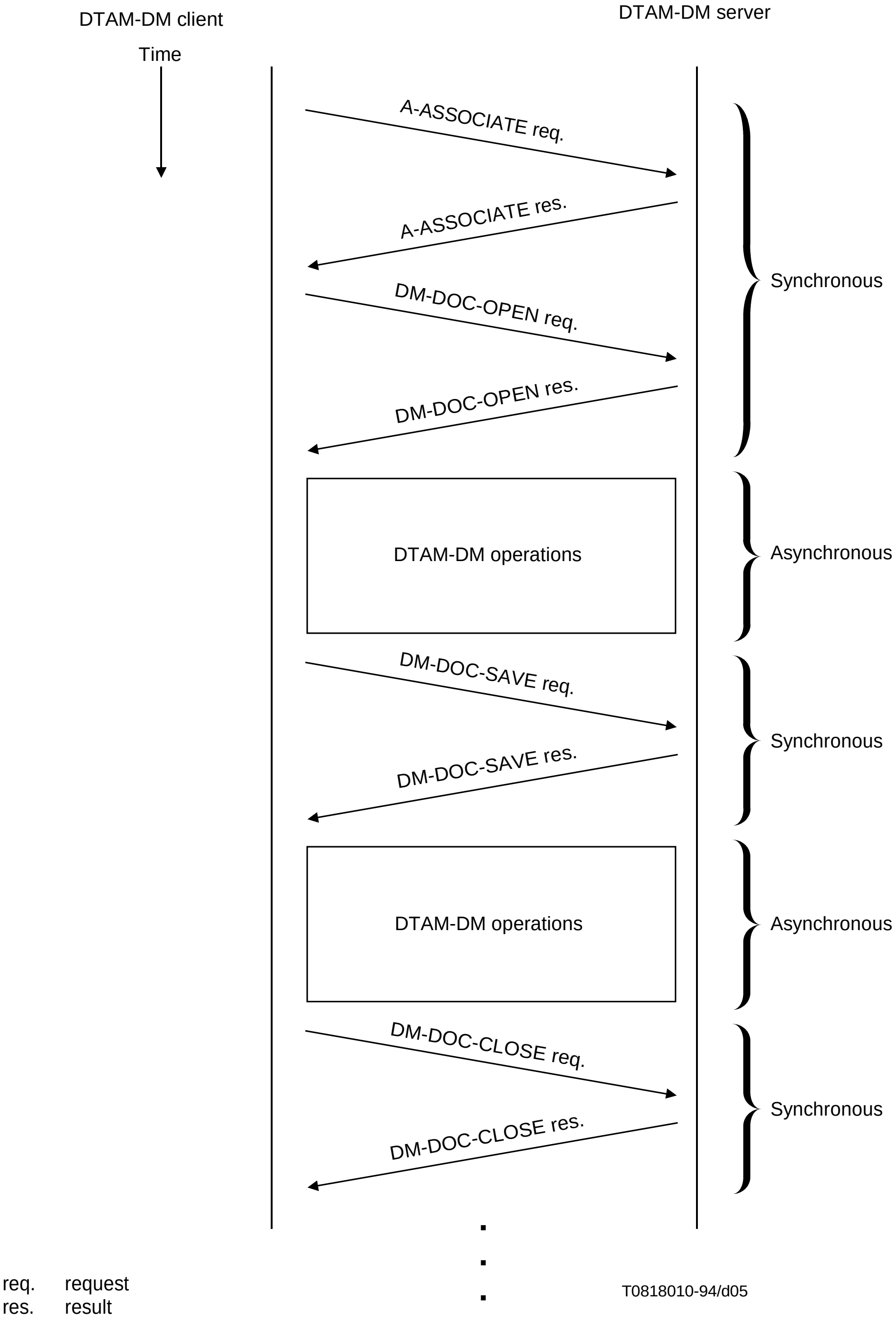
### I.6 Document close

After the manipulation of the document fragments, the DTAM-DM client closes the document.

### I.7 DTAM association termination

After all the activities for document manipulations, the DTAM-DM client terminates the association using the ACSE A-RELEASE service to close the communication link between the DTAM users.

Figure I.1 illustrates an example for the use of DTAM-DM abstract services.



NOTE – The only operations that are asynchronously invoked and results return asynchronously as well, are the operations contained in the DTAM-DM confirmed document manipulation functional unit.

FIGURE I.1/T.435

**Example for the use of DTAM-DM abstract services**

# Appendix II

## Basic procedures for the combined use of DTAM-DM and DFR abstract services

(This appendix does not form an integral part of this Recommendation)

The configuration for combined use of DTAM-DM and DFR constitutes a client-server system. The DTAM-DM/DFR client makes use of both DTAM-DM and DFR abstract services while the DTAM-DM/DFR server provides both DTAM-DM and DFR abstract services.

The following procedures are applied to the combined use of DTAM-DM and DFR abstract services:

### II.1 DTAM-DM/DFR association establishment

The DTAM-DM/DFR client creates a DTAM-DM/DFR association to the DTAM-DM/DFR server by the ACSE A-ASSOCIATE service. Subsets of DTAM-DM and DFR services may be used as agreed during association establishment. During this establishment, the DTAM-DM/DFR client and DTAM-DM/DFR server also have to agree on their application capabilities including manipulation of document fragments.

### II.2 Document filing and retrieval (use of DFR abstract services)

After DTAM-DM/DFR association establishment the DTAM-DM/DFR client can use any DFR abstract service, for example to browse, retrieve, manage or delete documents in the document store.

The DTAM-DM/DFR client selects a document using a DFR service and obtains a document identification. Before manipulating the document, the DTAM-DM/DFR client should modify the access rights of the entire document for other DTAM-DM/DFR clients, using the DFR reserve service, in order to prevent other DTAM-DM/DFR clients to access the document.

### II.3 Document manipulation (use of DTAM-DM-abstract services)

Before manipulating document fragments, the DTAM-DM/DFR client opens the document. The DTAM-DM/DFR client is able to manipulate fragments of more than one document.

The DTAM-DM/DFR client uses the DTAM-DM abstract services to manipulate the document fragments remotely. For more details about DTAM-DM abstract services, see Appendix I.

During document manipulation, the DTAM-DM/DFR client can use all convenient DFR abstract services.

Upon completion of all document manipulation operations, the DTAM-DM/DFR client closes the document and then unreserves the document using the DFR reserve service in order to allow other DTAM-DM/DFR clients to access the document.

### II.4 DTAM-DM/DFR association termination

After all the activities for document filing, retrieval and manipulation the DTAM-DM/DFR client terminates the DTAM-DM/DFR association using the ACSE A-RELEASE service to close the communication link between the DTAM-DM/DFR client and the DTAM-DM/DFR server.

Figure II.1 illustrates an example for the combined use of DTAM-DM and DFR abstract services.

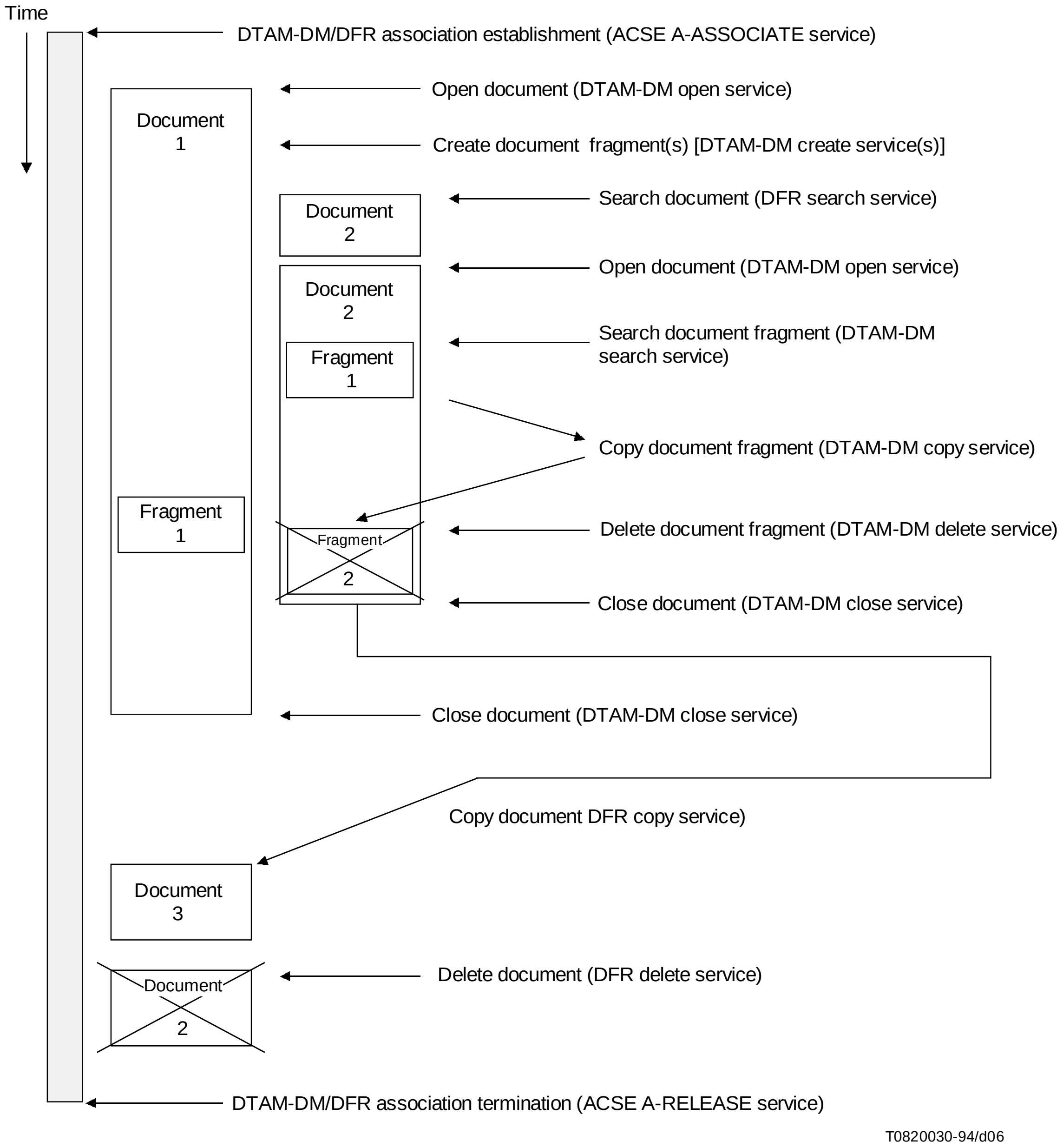

Time
DTAM-DM/DFR association establishment (ACSE A-ASSOCIATE service)
Open document (DTAM-DM open service)
Document 1
Create document fragment(s) [DTAM-DM create service(s)]
Document 2
Search document (DFR search service)
Document 2
Open document (DTAM-DM open service)
Fragment 1
Search document fragment (DTAM-DM search service)
Copy document fragment (DTAM-DM copy service)
Fragment 1
Fragment 2
Delete document fragment (DTAM-DM delete service)
Close document (DTAM-DM close service)
Close document (DTAM-DM close service)
Copy document DFR copy service)
Document 3
Document 2
Delete document (DFR delete service)
DTAM-DM/DFR association termination (ACSE A-RELEASE service)
T0820030-94/d06

FIGURE II.1/T.435

**Example for the combined use of DTAM-DM and DFR abstract services**