INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

T.413

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/94)

TERMINALS FOR TELEMATIC SERVICES

INFORMATION TECHNOLOGY – OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT: ABSTRACT INTERFACE FOR THE MANIPULATION OF ODA DOCUMENTS

ITU-T Recommendation T.413

(Previously “CCITT Recommendation”)

Foreword

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the Members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology, which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation T.413 was approved on 11th of November 1994. The identical text is also published as ISO/IEC International Standard 8613-3.

NOTE

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

CONTENTS

LIST OF TABLES

Page

Summary

This Recommendation | International Standard describes an abstract interface that supports manipulation of ODA documents and the functions that are applicable to document fragments in order to facilitate the handling of ODA documents and to support applications

Foreword

This ITU-T Recommendation | International Standard was prepared as a joint publication by ITU-T Study Group 8 and ISO/IEC Joint Technical Committee 1.

At present, the ITU-T Recommendations in the T.410-Series | International Standard ISO/IEC 8613 consists of:

- Introduction and general principles;
- Document structures;
- Abstract interface for the manipulation of ODA documents;
- Document profile;
- Open document interchange format;
- Character content architectures;
- Raster graphics content architectures;
- Geometric graphics content architectures;
- Audio content architectures;
- Formal specification of the Open Document Architecture (FODA)

  (The formal specification is applicable to ISO/IEC 8613 only.)
- Tabular structures and tabular layout;
- Identification of document fragments;
- Temporal relationships and non-linear structures.

Further Recommendations | International Standards may be added to this series of ITU-T Recommendations | International Standards.

Development of this series of ITU-T Recommendations | International Standards was originally in parallel with ECMA-101 standard: *Open Document Architecture*.

This ITU-T Recommendation | International Standard contains two annexes:

- Annex A (integral) specifies the alignment with DTAM abstract service;
- Annex B (non-integral) specifies the alignment with DOAM set of operations.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT: ABSTRACT INTERFACE FOR THE MANIPULATION OF ODA DOCUMENTS

1 Scope

The purpose of ITU-T Rec. T.410-Series | ISO/IEC 8613 is to facilitate the interchange and manipulation of documents.

In the context of these Recommendations | International Standards, documents are considered to be items such as memoranda, letters, invoices, forms and reports, which may include pictures and tabular material. The content elements used within the documents may include graphic characters, raster graphics elements and geometric graphics element, all potentially within one document.

NOTE – These Recommendations | International Standards are designed to allow for extensions, including spreadsheets and additional types of content such as video.

In addition to the content types defined in these Recommendations | International Standards, ODA also provides for arbitrary content types to be included in documents.

These Recommendations | International Standards apply to the interchange of documents by means of data communications or the exchange of storage media.

These Recommendations | International Standards provide for the interchange of documents for either or both of the following purposes:

- to allow presentation as intended by the originator;
- to allow processing, such as editing and reformatting.

The composition of a document in interchange can take several forms:

- formatted form, allowing presentation of the document;
- processable form, allowing processing of the document;
- formatted processable form, allowing both presentation and processing of the document.

These Recommendations | International Standards also provide for the interchange of ODA information structures used for the processing of interchanged documents.

This Recommendation | International Standard:

- describes an Abstract Interface that supports manipulation of ODA documents;
- defines the operations that are applicable to document fragments in order to facilitate the handling of ODA documents, and to support applications.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T/CCITT Recommendations.

2.1 Identical Recommendations | International Standards

- 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.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*.
- ITU-T Recommendation T.416 (1993) | ISO/IEC 8613-6:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Character content architectures*.
- ITU-T Recommendation T.417 (1993) | ISO/IEC 8613-7:1994, *Information technology – Open Document Architecture (ODA) and interchange format: 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 architectures*.
- ITU-T Recommendation T.419[1)] | ISO/IEC 8613-9: …[1)], *Information technology – Open Document Architecture (ODA) and interchange format: Audio content architectures*.
- ITU-T Recommendation T.421 (1994) | ISO/IEC 8613-11:1995, *Information technology – Open Document Architecture (ODA) and interchange format: Tabular structures and tabular layout.*
- ITU-T Recommendation T.422[1)] | ISO/IEC 8613-12: …[1)], *Information technology – Open Document Architecture (ODA) and interchange format: Identification of document fragments*.
- ITU-T Recommendation T.424[1)] | ISO/IEC 8613-14: …[1)], Information technology – Open Document Architecture (ODA) and interchange format: Temporal relationships and non-linear structures.

2.2 Paired Recommendations | International Standards equivalent in technical content

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

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.
- CCITT X.400-Series Recommendations (1988), *Message Handling System*.

  ISO/IEC 10021:1990, *Information technology – Text communication – Message Oriented Text Interchange Systems (MOTIS)*.

2.3 Additional references

- CCITT Recommendation T.431 (1991), *Document Transfer and Manipulation (DTAM) – Services and protocols - Introduction and general principles*.
- CCITT Recommendation T.432 (1991), *Document Transfer and Manipulation (DTAM) – Services and protocols - Service definition*.
- CCITT Recommendation T.433 (1991), *Document Transfer and Manipulation (DTAM) – Services and protocols - Protocol specification*.
- ITU-T Recommendation T.435 (1995), *Document Transfer and Manipulation (DTAM) – Services and protocols – Abstract service definition and procedures for confirmed document manipulation*.
- ITU-T Recommendation T.436 (1995), *Document Transfer and Manipulation (DTAM) – Services and protocols – Protocol specifications for confirmed document manipulation*.
- ISO/IEC 8613-10:1991, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 10: Formal specifications.*
- 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*.

---

1) Presently at the stage of draft.

3 Definitions

For the purposes of this Recommendation | International Standard, the definitions given in ITU-T Rec. T.411 | ISO/IEC 8613-1 and ITU-T Rec. T.422 | ISO/IEC 8613-12 apply.

The following additional definitions are used within this Recommendation | International Standard:

3.1 basic operation: An operation that could not consist of a sequence of other operations without a clear loss of efficiency.

3.2 compound operation: An operation that could be substituted by a sequence of other operations.

3.3 document identifier: An identification of an ODA document that may be permanent (the "document reference" document profile attribute) or not (an integer value).

3.4 query: A logical expression comprising logical operators that connect values of attributes.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the abbreviations given in ITU-T Rec. T.411 | ISO/IEC 8613-1 apply.

The following additional abbreviations are used within this Recommendation | International Standard:

AGCS	Audio Graphics Conferencing Service
AVIS	Audio Visual Interactive Service
CDH	Cooperative Document Handling
DAP	Document Application Profile
DFR	Document Filing and Retrieval
DOAM	Distributed Office Applications Model
DTAM	Document Transfer And Manipulation
DTAM-DM	Document Transfer And Manipulation - Document Manipulation
ROSE	Remote Operations Service Element

5 Conventions

For the purposes of this Recommendation | International Standard, the conventions given in ITU-T Rec. T.411 | ISO/IEC 8613-1 and ITU-T Rec. T.412 | ISO/IEC 8613-2 apply.

6 Purpose

6.1 Concept of the Abstract Interface

Documents may be edited, formatted, imaged, manipulated, and stored locally or in an open environment. The Abstract Interface applies to those aspects of manipulation that may be defined in an open environment. Nevertheless, it may be up to a specific implementation to provide interfaces in a compatible way for the open and the local environments.

Although the Abstract Interface defines functions to handle ODA documents, it does not preclude any definition of any specific application, but the applications themselves are responsible for the proper use of the operations provided by the Abstract Interface.

The objectives of the Abstract Interface are summarized in the following list of functions:

- Ensure consistency of manipulated ODA documents.
- Facilitate distributed processing of ODA documents. The Abstract Interface hides distribution details. Furthermore, local integration of products from different vendors can be achieved by using the Abstract Interface.
- Enable the defined operations to be mapped onto existing services, such as Document Transfer And Manipulation (DTAM).

- Facilitate the integration of a variety of applications that manipulate ODA documents. The applications can be designed and interfaced at a higher level of abstraction.
- Present the user a simple conceptual model of a complex ODA document.

The operations that are defined in the Abstract Interface are specified on a general level to support document manipulation in the context of various types of applications, such as:

- Reading and manipulation of fragments of a document.
- Simple conferencing, where two or more users possess identical copies of a document and are exchanging updates on document fragments in terms of operations.
- Audio Graphics Conferencing Service (AGCS).
- Remote editing, where one or more users have a complete or partial view of a document. The document is read and updated by exchanging operations.
- Cooperative Document Handling (CDH), where several partners handle ODA documents in a distributed environment.
- Videotex Interworking, for compatibility with ITU-T applications.
- AudioVisual Interactive Service (AVIS).
- ODA document processing applications development, including word processors and desktop publishing systems.
- Asynchronous distributed document handling.

6.2 Relationship to functional profiles

For document manipulation, specific functional profiles or amendments to existing ones may be defined to specify:

- which operations may be used;
- which operations may be applied to which type of constituents;
- which sets of constituents shall be carried together by one operation;
- which rules have to be followed to guarantee consistency of the document;
- how the grouping of operations shall be used;
- how the reserving of constituents of a document shall be realized.

6.3 Identification

In the Abstract Interface for the manipulation of ODA documents, the mechanism for identifying document fragments, to be used as arguments to the manipulation operations, is based on the use of location expressions, according to ITU-T Rec. T.422 | ISO/IEC 8613-12.

6.4 Consistent handling of ODA documents

It is a basic requirement of the manipulation of ODA documents that consistency of the documents being manipulated is guaranteed. After successful manipulation and closing of a document, it shall be a conformant ODA document and it shall conform to the Document Application Profile (DAP) specified in the document profile, if any.

The Abstract Interface defines some general rules to describe the effect that the different operations have on the documents. The implementations of the Abstract Interface shall guarantee the consistency of the documents. To help in this task, BeginGroup and EndGroup operations are provided, that could be used to indicate that the manipulated document has to be updated in order to achieve the required consistency.

6.5 Relationship between operations

The List operation applies to the document store, and it is used to select one or more documents for further manipulation.

The Open operation identifies a document which is to be handled. Several documents may be opened at the same time (e.g. for copying a document fragment from one document to another). There is a Close operation to complement the Open operation.

Before issuing any read only, altering, or reservation operation (see clause 7), the Open operation shall be issued. The Open operation also allows to create a completely new document.

A Reserve operation (and a complementary Unreserve operation) is provided to handle the possible concurrent manipulation of a document fragment. If a document fragment is manipulated without a previous use of the Reserve operation, the result may be unpredictable if other users have access to that document fragment.

Read only and altering operations may be issued without any particular order restriction, apart from those given in this subclause.

The use of all the operations is explained in the corresponding subclauses of clause 7.

6.6 Grouping of operations

A mechanism for grouping a sequence of operations is provided by the Abstract Interface.

The application shall be responsible for the correct use of this feature and shall define the meaning of the grouping of operations.

The main objective of this mechanism is to group operations with some meaning for the application that is using the Abstract Interface, in order to:

- avoid communications overhead;
- ease the handling of consistency, including the provision of recovery procedures applied to a group of operations.

This mechanism is provided by BeginGroup and EndGroup operations (see 7.5.1 and 7.5.2), that allow grouping to be performed during operations invocation.

6.7 Relationship to communication aspects

The Abstract Interface is independent of the communication mechanism. However, it is aligned with the communication support for access and manipulation provided by DTAM, as explained in Annex A.

In the context of interactive handling, three aspects are distinguished:

- the access to an ODA document;
- the inner manipulation of an accessed ODA document;
- the communication environment supporting access and manipulation.

In the case that the Abstract Interface is used by communication applications, specific communication application profiles, or functional profiles, may be specified to:

- transfer manipulations of ODA documents in an appropriate way;
- negotiate the application environment;
- negotiate the capabilities of the user system.

NOTE – Rules for specifying communication application profiles are defined, for example, in CCITT Recommendation T.431.

7 Operations on ODA documents

This clause specifies the operations that constitute the Abstract Interface to support the manipulation of documents.

The operations are specified as abstract operations in terms of:

- argument(s);
- result(s);
- error(s).

Operations are applied to the constituents of the document architecture.

Five kinds of operations are distinguished by the Abstract Interface:

- document level operations;
- read only operations;
- altering operations;
- reservation operations;
- ODA-independent operations.

Furthermore, operations are classified as:

- *basic operations;*
- *compound operations.*

Compound operations can be performed as a sequence of basic operations.

The operations defined by the Abstract Interface are aligned with those defined in ISO/IEC 10031-1 Distributed Office Applications Model (DOAM), as explained in Annex B.

Some errors are defined that apply to several operations. These are:

- *Error in the constituents* – The constituent or constituents of the argument are not valid.
- *Invalid location expression or document* – The document or the location expression used as arguments are invalid.
- *Location expression does not match* – The location expression of the argument does not match.
- *Document fragments reserved (source region reserved, target region reserved)* – The location expression of the argument contains reserved constituents. For some operations which work with two documents, this error is referred to as "source region reserved" (for the origin document) and "target region reserved" (for the destination document).
- *Invalid document identifier* – The document identifier of the argument is not valid.
- *Document does not exist* – The document that is intended to be opened or closed does not exist.
- *Improper access rights* – The requestor of the operation does not have the necessary access rights to perform the operation.
- *Document is not open* – A manipulation operation is requested onto a document that has not been previously open using the Open operation.
- *Unspecified error* – An error has occurred that is not specified.

7.1 Document level operations

7.1.1 List

The List abstract operation obtains a list of ODA documents, matching particular criteria, in a document store. This operation is independent of the structure of the store. If the store has a standard structure, like, for example, that specified in ISO/IEC 10166 Document Filing and Retrieval (DFR), selection of documents may be achieved using DFR operations.

The selection criteria are expressed in a *query* that consists of a logical expression comprising 'not', 'and' and 'or' logical operators that connect values of document profile management attributes. In this way, for example, documents with specific values in one or more attributes can be selected.

If the "query" argument is not provided, then all documents from the store are listed.

(Identifiers of) complete documents are selected with the List operation. For selecting (identifiers of) document constituents the Search operation (see 7.2.2) shall be used.

Possible errors specific for this operation are:

- *Invalid query* – The query of the argument is not valid.
- *Query does not match* – The query of the argument does not match with any document.

This is a basic operation.

Argument: query. *(optional)*

Result: sequence of document identifiers.

Errors: invalid query;
query does not match;
improper access rights;
unspecified error.

7.1.2 Open

The Open abstract operation selects a particular document in order to make further manipulations on fragments of it. The document identifier used as the argument may be obtained from a previous use of the List operation.

Several documents may be maintained open at the same time. This shall be achieved by issuing several Open operations before closing all open documents.

An Open operation is always necessary before inner manipulation of a document.

A complete document is selected with the Open operation. The Reserve operation (see 7.4.1) is used to select and reserve a fragment of a document already open.

The "document identifier" argument shall be a permanent identifier (the "document reference" attribute of the document profile). The "mode" optional argument may have three values: 'read', 'modify' and 'create'. If a document is open with "mode" equal to 'read', it will only be possible to perform read operations on it. In case it is open with "mode" equal to 'modify', it will be possible to apply all manipulation operations. Finally, with the 'create' value, it means that the document is to be created. In this latter case, the permanent identifier is the identification to be given to the document.

The "document identifier" result is a non-permanent (temporary) identifier of the document (as an integer value) that is provided when opening a document.

A possible error specific for this operation is:

– *Document already open* – This error is used when there is an attempt to open an already open document.

This is a basic operation.

Arguments:	document identifier; mode.	*(permanent identifier)* *(optional)*
Result:	document identifier.	*(non-permanent identifier)*
Errors:	invalid document identifier; document does not exist; improper access rights; document already open; unspecified error.	

7.1.3 Close

The Close abstract operation is used to finish the manipulation process of a document. The document identifier used as the argument has to correspond to an already open document. No argument is needed if only one document is open.

When the application wants to close a document, the document could be inconsistent if altering operations (see 7.3) have been applied to the document. In this case, and to avoid closing a non-conformant document, the document is implicitly updated.

Furthermore, when closing a document, all reserved document fragments are implicitly unreserved (see 7.4.1 and 7.4.2).

The "document identifier" result is the permanent identifier of the document (the "document reference" attribute of the document profile), that is provided when closing the document.

Possible errors specific for this operation are:

– *Document impossible to update* – There is a problem in the document to be closed that makes the document impossible to update.

– *Groups not yet ended* – Some BeginGroup operations still need EndGroup operations to be issued.

This is a basic operation.

Argument:	document identifier.	*(optional)*
Result:	document identifier.	*(permanent identifier)*
Errors:	invalid document identifier; improper access rights; document is not open; document impossible to update; groups not yet ended; unspecified error.	

7.2 Read only operations

7.2.1 Get

The Get operation is used to request one or more constituents of the ODA document. In the case of more than one constituent, the result can be one or more subtrees.

The Get operation returns the relevant constituents, identified by the "location expression" argument, with all their attributes. The "no-defaults" argument determines whether or not defaulted values are returned. If this argument is not provided, defaulted attributes are returned.

The "document identifier" argument is not needed if only one document is open.

The location expression used as argument may be a constituent identifier obtained from a previous use of the Search operation (see 7.2.2).

This is a basic operation.

Arguments: document identifier; *(optional)*
location expression;
no-defaults. *(optional)*

Result: one or more constituent descriptions.

Errors: invalid location expression or document;
location expression does not match;
document fragments reserved;
improper access rights;
document is not open;
unspecified error.

7.2.2 Search

The Search abstract operation looks for the information specified in the location expression of the argument, that may be:

- structure information;
- content portions information (the way this information is searched for depends on the content architecture in use);
- structure and content information.

The result of the Search operation is the sequence of identifiers matching the search criteria specified in the "location expression" argument.

The number of occurrences of the target information that are returned may be limited with the "maximum number of occurrences" argument. If this argument is not provided, then all occurrences are returned. In any case, the number of returned occurrences is given in the result.

The "document identifier" argument is not needed if only one document is open.

The Search and Get operations may each return more than one constituent. The difference is that the Search operation returns only constituent identifiers, while the Get operation returns complete constituents.

NOTE 1 – A normal use of the Search operation is before a Get operation to select the constituent or constituents that are required.

The difference between the Search and List operations is that the former applies to one document and returns constituent identifiers, while the latter applies to a set of documents and returns document identifiers.

This is a basic operation.

Arguments: document identifier; *(optional)*
location expression;
maximum number of occurrences. *(optional)*

Results: identification of all the objects where the target information was found, up to the specified maximum; the number of occurrences of the specified information.

Errors: invalid location expression or document;
location expression does not match;
document fragments reserved;

improper access rights;
document is not open;
unspecified error.

NOTE 2 – Following is an example of the use of the Search operation.

A location expression (see ITU-T Rec. T.422 | ISO/IEC 8613-12) that could be used as an argument to a Search operation to get the identifiers of all footnotes in a document conforming to ISO/IEC ISP 11181-1 would be:

```
OBJECT-WITH          (attribute-name =          "application-comments",
                     value-specification =      "$Footnote",
                     object-locator =           "3" -- document logical root--,
                     ( , end-counter =          0),
                     defaulting =               true)
```

where "$Footnote" stands for the value that ISO/IEC ISP 11181-1 mandates for the attribute "application-comments" of those constituents representing footnotes.

The result would be a list of identifiers, looking like (see ASN.1 specification in clause 8):

```
SearchResult =
    constituentList =       -- In the case of the Search operation, "ConstituentType" includes identifiers only.
        ConstituentType =       "3 2 0 1"
        ConstituentType =       "3 4 5 1"
    numberOfOccurrences =       2
```

7.3 Altering operations

7.3.1 Create

The Create abstract operation effects the creation and insertion of a new constituent or several new constituents (in the case of objects in the form of a subtree) into the document.

It is necessary to specify the constituents to be created and (except for root objects, styles and classes) the place where they are to be included (a location expression) and also the relative position of the new constituent(s) with respect to the constituent obtained from the location expression.

The place to include the created constituents shall be indicated by a constituent location, i.e. a location expression that (as specified in ITU-T Rec. T.422 | ISO/IEC 8613-12) selects a simple constituent.

Two options are provided for the relative position, that include all possible cases: *before* the identified constituent, or *after last child* of the identified constituent.

The “document identifier” argument is not needed if only one document is open.

If a constituent or a subtree is successfully created (the document is still complete, and, in the case of objects, the generic structure remains consistent), a value is automatically assigned to its “identifier” attribute (or to the “identifier” attribute of all the constituents of the subtree). The “identifier” attribute of the constituent (or of the root of the subtree) is returned in the “constituent identifier” result. When objects (other than the root) and content portions are created, the “subordinates” or “content portions” attribute of the superior of the constituent (or of the root of the subtree) is updated.

This is a basic operation.

Arguments:	one or more constituent descriptions;	
	document identifier:	*(optional)*
	constituent location;	*(only for objects, except roots)*
	position (before, after last child).	*(optional)*

Result: constituent identifier.

Errors: error in the constituents;
invalid location expression or document;
location expression does not match;
document fragments reserved;
improper access rights;
document is not open;
unspecified error.

7.3.2 Delete

The Delete abstract operation causes the deletion of one or more constituents of the ODA document, identified by the “location expression” argument, and implies the deletion of all constituents that are subordinate to the identified constituent(s). In the case of objects (other than the root) and content portions, the “subordinates” or “content portions” attribute of the superior is updated.

The “document identifier” argument is not needed if only one document is open.

The location expression used as argument may be a constituent identifier obtained from a previous use of the Search operation.

This is a basic operation.

Arguments: document identifier; *(optional)*
location expression.

Result: success or failure.

Errors: invalid location expression or document;
location expression does not match;
document fragments reserved;
improper access rights;
document is not open;
unspecified error.

7.3.3 Modify

The Modify abstract operation assigns new values to attributes of the identified (by the "location expression" argument), already existing, constituents.

Attributes that are not mentioned as a parameter of the Modify operation retain their previous values, unless the optional argument "deleting" is set to 'true'; in this case, optional attributes not included in the argument "attribute values" are deleted. This is the way of deleting attributes with the Modify operation.

Optional attributes, that have not been present in the identified constituent, may be added to the relevant constituent by use of the Modify operation. This is the way of adding new attributes with the Modify operation.

The "document identifier" argument is not needed if only one document is open.

Identification attributes, if present, shall not be changed by the Modify abstract operation. The constituent "identifier" attributes and the "subordinates", "object type" and "content portions" attributes shall not be modified.

The location expression used as argument may be a constituent identifier obtained from a previous use of the Search operation.

A possible error specific for this operation is:

– *Error in the attributes* – The constituent or constituents, giving the attribute values, of the argument are wrong.

This is a basic operation.

Arguments: document identifier; *(optional)*
location expression;
attribute values;
deleting. *(optional)*

Result: success or failure.

Errors: error in the attributes;
invalid location expression or document;
location expression does not match;
document fragments reserved;
improper access rights;
document is not open;
unspecified error.

7.3.4 Replace

The Replace abstract operation effects the deletion of the identified constituent, including all subordinate constituents, and the replacement by one new constituent, which may have subordinate constituents, specified in the argument of the operation.

The “document identifier” argument is not needed if only one document is open.

Replace is a compound operation that is provided by the Abstract Interface for efficiency and compatibility reasons. The Replace abstract operation is equivalent to the application of the Delete and Create operations.

Arguments:	one or more constituent descriptions;	
	document identifier;	*(optional)*
	constituent location.	(*only for objects, except roots)*

Result: success or failure.

Errors: error in the constituents;
invalid location expression or document;
location expression does not match;
document fragments reserved;
improper access rights;
document is not open;
unspecified error.

7.3.5 Copy

The Copy abstract operation duplicates one constituent or several constituents (in the case of objects in the form of a subtree) of an ODA document in another location in the same document or in a different document.

It is necessary to specify the constituents to be copied and (except for root objects, styles and classes) the place where they are to be copied (a target location expression) and also the relative position of the new constituent(s) with respect to the constituent obtained from the target location expression.

For the “position” argument, the same options as for the Create operation (see 7.3.1) are provided.

The place to include the copied constituents shall be indicated by a constituent location, as for the Create operation (see 7.3.1).

The document identifier arguments are not needed if only one document is open.

If a constituent or a subtree is successfully copied (the document is still complete, and, in the case of objects, the generic structure remains consistent), a value is automatically assigned to the “identifier” attribute of the target constituent (or of all the constituents of the target subtree). The “identifier” attribute of the target constituent (or of the root of the target subtree) is returned in the “constituent identifier” result. When objects (other than the root) and content portions are copied, the “subordinates” or “content portions” attribute of the superior of the target constituent (or of the root of the target subtree) is updated.

Copy is provided as a basic operation for efficiency reasons.

Arguments:	source document identifier;	*(optional)*
	source basic location expression;	
	target document identifier;	*(optional)*
	target constituent location;	*(only for objects, except roots)*
	position (before, after last child).	*(optional)*

Result: constituent identifier.

Errors: invalid source location expression or document;
invalid target location expression or document;
source location expression does not match;
target location expression does not match;
source region reserved;
target region reserved;
improper access rights;
source document is not open;
target document is not open;
unspecified error.

7.3.6 Move

The Move abstract operation changes the position of one constituent or several constituents (in the case of objects in the form of a subtree) inside an ODA document, or moves constituents from one document to another.

It is necessary to specify the constituents to be moved and (except for root objects, styles and classes) the place where they are to be moved (a target location expression) and also the relative position of the new constituent(s) with respect to the constituent obtained from the target location expression.

For the "position" argument, the same options as for the Create operation (see 7.3.1) are provided.

The place to include the moved constituents shall be indicated by a constituent location, as for the Create operation (see 7.3.1).

The document identifier arguments are not needed if only one document is open.

If a constituent or a subtree is successfully moved to a new place in the document (the document is still complete, and, in the case of objects, the generic structure remains consistent), a value is automatically assigned to its "identifier" attribute (or to the "identifier" attribute of all the constituents of the subtree). The "identifier" attribute of the constituent (or of the root of the subtree) is returned in the "constituent identifier" result. When objects (other than the root) and content portions are moved, the "subordinates" or "content portions" attribute of the new superior of the constituent (or of the root of the subtree) is updated.

When the moved constituent(s) are objects (other than the root) and content portions, the "subordinates" or "content portions" attribute of the superior of the constituent (or of the root of the subtree), before moving, is updated.

This is a compound operation that is provided by the Abstract Interface for efficiency and compatibility reasons. This operation is, in general, equivalent to a pair of Copy and Delete operations.

Arguments:

source document identifier;	*(optional)*
source basic location expression;	
target document identifier;	*(optional)*
target constituent location;	*(only for objects, except roots)*
position (before, after last child).	*(optional)*

Result: constituent identifier.

Errors: invalid source location expression or document;
invalid target location expression or document;
source location expression does not match;
target location expression does not match;
source region reserved;
target region reserved;
improper access rights;
source document is not open;
target document is not open;
unspecified error.

7.4 Reservation operations

7.4.1 Reserve

This operation, together with the Unreserve operation (see 7.4.2), is provided for multi-use of a document.

The Reserve abstract operation causes the reservation, if possible, of one or more constituents (including all subordinate constituents), identified by the "location expression" argument, of the ODA document for further manipulation by the requesting user.

With the Reserve operation, it is possible to select fragments of a document previously open with the Open operation.

The “document identifier” argument is not needed if only one document is open.

A “reservation identifier” result (as an integer value) is returned by this operation, that temporarily identifies the reserved location expression.

Reserving of constituents of a document that have impact on, or are referenced by, other constituents may require a strategy to reserve also implicitly or explicitly the constituents that may be influenced by the subsequent manipulation.

This is a basic operation.

Arguments:	document identifier;	*(optional)*
	location expression.	
Result:	reservation identifier.	
Errors:	invalid location expression or document;	
	location expression does not match;	
	document fragments reserved;	
	improper access rights;	
	document is not open;	
	unspecified error.	

7.4.2 Unreserve

This operation, together with the Reserve operation (see 7.4.1), is provided for multi-use of a document.

The Unreserve abstract operation causes the release of one or more constituents (including all subordinates), identified by the “location expression” argument, previously reserved by the Reserve operation.

The “document identifier” argument is not needed if only one document is open.

A “reservation identifier” argument may be used instead of a location expression.

The Unreserve operation, like the Reserve operation, applies to document fragments, not to complete documents.

The strategy followed by applications on reserving document fragments has to be coherent with the way of using the Unreserve operation.

A possible error specific for this operation is:

- *Constituents are not reserved* – This error is returned when trying to unreserve constituents not previously reserved.

This is a basic operation.

Arguments:	document identifier;	*(optional)*
	location expression;	*(optional)*
	reservation identifier.	*(optional)*
Result:	success or failure.	
Errors:	invalid location expression or document;	
	location expression does not match;	
	improper access rights;	
	document is not open;	
	constituents are not reserved;	
	unspecified error.	

7.5 ODA-independent operations

Two operations that are ODA-independent are provided. Both are related to grouping.

BeginGroup and EndGroup operations (see 7.5.1 and 7.5.2) mark the beginning and end of a group of operations. They may be used for the following purposes:

- informing a communication protocol that the selected group of operations (those between BeginGroup and EndGroup) could be communicated in one single interchange;
- informing a server (document store) that the selected group of operations (those between BeginGroup and EndGroup) have all to be processed at the same time.

Nesting is allowed for mixing both kinds of groups of operations.

7.5.1 BeginGroup

The BeginGroup operation marks the beginning of a group of operations that have a special meaning for the application.

This operation is always used in combination with the EndGroup operation (see 7.5.2).

Groups of operations may be nested if allowed by the application. An invocation identifier is used to distinguish between different nested instances of the same group of operations, identified by the Operation group identifier.

A possible error specific for this operation is:

– *Invocation identifier duplicated* – An existing invocation identifier is being used.

This is a basic operation.

Arguments: operation group identifier;
invocation identifier.

Result: success or failure.

Errors: invocation identifier duplicated;
unspecified error.

7.5.2 EndGroup

The EndGroup operation marks the end of a previously initiated group of operations.

This operation is always used in combination with the BeginGroup operation (see 7.5.1).

A possible error specific for this operation is:

– *No BeginGroup* – A previous BeginGroup operation has not been issued.

This is a basic operation.

Arguments: operation group identifier;
invocation identifier.

Result: success or failure.

Errors: no BeginGroup;
unspecified error.

8 ASN.1 specification of the Abstract Operations

In this clause, the arguments and results of the abstract operations defined in clause 7 are specified using ASN.1. The use of ASN.1 does not imply any encoding of the operations; such encoding is outside the scope of this Specification, and may be specified in other Recommendations | International Standards.

NOTE – Rules for specifying how to encode operations are defined, for example, in ITU-T Recommendation T.435 DTAM Document Manipulation.

```
Abstract-Operations { 2 8 1 3 0 }

DEFINITIONSIMPLICIT TAGS ::= BEGIN

EXPORTS     EVERYTHING;

IMPORTS     Location-expression, Basic-location-expression, Constituent-locator
                FROM Location-Model { 2 8 1 12 0 }
                    -- see ITU-T Rec. T.422 | ISO/IEC 8613-12

            Document-Reference, Document-Profile-Descriptor
                FROM Document-Profile-Descriptor { 2 8 1 5 6 }
                    -- see ITU-T Rec. T.415 | ISO/IEC 8613-5

            Object-or-Class-Identifier, Content-Portion-Identifier, Style-Identifier
                FROM Identifiers-and-Expressions { 2 8 1 5 7 }
                    -- see ITU-T Rec. T.415 | ISO/IEC 8613-5

            Interchange-Data-Element
                FROM Interchange-Data-Elements { 2 8 1 5 5 };
                    -- see ITU-T Rec. T.415 | ISO/IEC 8613-5

-- COMMON DATA TYPES

LocationInDocumentType ::= SEQUENCE {
        document        [0]     Document-Id OPTIONAL,
        location        [1]     Location-expression }

BasicLocationInDocumentType ::= SEQUENCE {
        document        [0]     Document-Id OPTIONAL,
        basic-location  [1]     Basic-location-expression }

ConstituentLocationInDocumentType ::= SEQUENCE {
        document                [0]     Document-Id OPTIONAL,
        constituent-location    [1]     Constituent-locator OPTIONAL }  -- It is optional for styles and
                                                                        -- root

Document-Id ::= CHOICE {
        permanent       [0]     Document-Reference,
        non-permanent   [1]     INTEGER }

ConstituentType ::= ManipulationDataElement

ManipulationDataElement ::= Interchange-Data-Element

SuccessType ::= BOOLEAN

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

ConstituentIdentifier ::= CHOICE {
        object-or-class         [0]     Object-or-Class-Identifier,
        content-portion         [1]     Content-Portion-Identifier,
        style                   [2]     Style-Identifier }

-- ABSTRACT OPERATIONS ARGUMENTS AND RESULTS

-- DOCUMENT LEVEL OPERATIONS

-- List

ListArgument ::= QueryType

QueryType ::= CHOICE {
        basic[0]    Document-Profile-Descriptor,    -- Some values of document profile attributes
        not         [1]     QueryType,
        and         [2]     SET OF QueryType,
        or          [3]     SET OF QueryType }

ListResult ::= SEQUENCE OF Document-Id              -- Permanent identifiers

-- Open

OpenArgument ::= SEQUENCE {
```

```
        identifier [0] Document-Id,        -- Permanent identifier
        mode       [1] ModeType OPTIONAL }

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

OpenResult ::= Document-Id          -- Non-permanent identifier

-- Close

CloseArgument ::= Document-Id

CloseResult ::= Document-Id         -- Permanent identifier

-- READ ONLY OPERATIONS

-- Get

GetArgument ::= SEQUENCE {
        location-in-document [0] LocationInDocumentType,
        no-defaults          [1] BOOLEAN DEFAULT FALSE }

GetResult ::= SEQUENCE OF SEQUENCE OF ConstituentType

-- Search

SearchArgument ::= SEQUENCE {
        location-in-document [0] LocationInDocumentType,
        maxOccurrences       [1] INTEGER OPTIONAL }

SearchResult ::= SEQUENCE {
        constituentList    [0] SEQUENCE OF ConstituentIdentifier,
        numberOfOccurrences [1] INTEGER }

-- ALTERING OPERATIONS

-- Create

CreateArgument ::= SEQUENCE {
        constituent-or-subtree [0] SEQUENCE OF ConstituentType,   -- If the elements of this
                                                                  -- sequence are objects, they
                                                                  -- form a subtree
        location-in-document   [1] ConstituentLocationInDocumentType OPTIONAL,
        position               [2] PositionType OPTIONAL }

CreateResult ::= ConstituentIdentifier

-- Delete

DeleteArgument ::= LocationInDocumentType

DeleteResult ::= SuccessType

-- Modify

ModifyArgument ::= SEQUENCE {
        location-in-document [0] LocationInDocumentType,
        attributeValue       [1] AttributeValuesType,
        deleting             [2] BOOLEAN DEFAULT FALSE }

AttributeValuesType ::= Interchange-Data-Element

ModifyResult ::= SuccessType

-- Replace
```

```
ReplaceArgument ::= SEQUENCE {
        constituent-or-subtree      [0]     SEQUENCE OF ConstituentType,        -- If the elements of this
                                                                                -- sequence are objects, they
                                                                                -- form a subtree
        location-in-document        [1]     ConstituentLocationInDocumentType }

ReplaceResult ::= SuccessType

-- Copy

CopyArgument ::= SEQUENCE {
        source      [0]   BasicLocationInDocumentType,
        target      [1]   ConstituentLocationInDocumentType,
        position    [2]   PositionType OPTIONAL }

CopyResult ::= ConstituentIdentifier
-- Move

MoveArgument ::= SEQUENCE {
        source      [0]   BasicLocationInDocumentType,
        target      [1]   ConstituentLocationInDocumentType,
        position    [2]   PositionType OPTIONAL }

MoveResult ::= ConstituentIdentifier
-- OTHER OPERATIONS
-- Reserve

ReserveArgument ::= LocationInDocumentType

ReserveResult ::= Reservation-Id

Reservation-Id ::= INTEGER
-- Unreserve

UnreserveArgument ::= CHOICE {
        location-in-document        [0]     LocationInDocumentType,
        reservation                 [1]     Reservation-Id }

UnreserveResult ::= SuccessType
-- BeginGroup

BeginGroupArgument ::= SEQUENCE {
        group-identifier   [0]      INTEGER,
        invocation-identifier       [1]     INTEGER }

BeginGroupResult ::= SuccessType
-- EndGroup

EndGroupArgument ::= SEQUENCE {
        group-identifier   [0]      INTEGER,
        invocation-identifier       [1]     INTEGER }

EndGroupResult ::= SuccessType

END
```

Annex A

Alignment with DTAM document manipulation abstract service

(This annex forms an integral part of this Recommendation | International Standard)

A.1 Introduction

The ODA Abstract Interface is intended (but not restricted) to be used in conjunction with other Recommendations | International Standards that facilitate the open interchange of the manipulations defined by this Specification.

Even if, in some applications, operations may be transferred asynchronously as messages, or may be combined in a file, e.g. for a bulk-update of a document, the most typical way to use the Abstract Interface is the direct mapping to the DTAM Document Manipulation (DTAM-DM) abstract service, defined in ITU-T Recommendation T.435.

A.2 Description of the mapping to DTAM-DM abstract service

The operations defined in this Specification are mapped to operations defined in ITU-T Recommendation T.435.

This Specification defines the abstract operations that can be applied to a document. DTAM-DM provides the service to transfer these operations and defines the encoding of operations for the open interchange. Abstract Interface operations, arguments, results and errors are then mapped to DTAM-DM operations, arguments, results and errors. Furthermore, DTAM-DM specifies their encoding and how the Remote Operations Service Element (ROSE) shall be used.

Each operation contained in the Abstract Interface has one corresponding operation in the DTAM-DM abstract service. Both operations have identical names, except that DTAM-DM operations are combined with the prefix DM- (e.g. the Create abstract operation of the Abstract Interface is mapped to the DM-CREATE of the DTAM-DM abstract service). There are few exceptions to this rule:

- there are no equivalent operations to the DTAM-DM abstract operations Save, Discard, Point and Macro-Call;
- the Open, Close, List, Save and Discard operations have the DM-DOCUMENT- prefix because they apply to complete documents;
- in ITU-T Recommendation T.435 the BeginGroup and EndGroup operations (called DM-GROUP-BEGIN and DM-GROUP-END) are left for further study, as the DM-MACRO-CALL abstract service.

DTAM-DM operations may apply to other document formats than ODA (including proprietary formats). For this reason, names of parameters contained in DTAM-DM operations may be more general than the corresponding names in the Abstract Interface.

Table A.1 specifies the mapping of arguments from the Abstract Interface operations to the corresponding arguments of the DTAM-DM abstract service operations.

Table A.1 – Mapping of arguments between Abstract Interface and DTAM-DM operations

Operation	ODA Abstract Interface	DTAM-DM abstract service
LIST	query	criteria
OPEN	document identifier mode	document id mode
CLOSE	document identifier	document id
GET	location in document, no-defaults	manipulation object
SEARCH	location in document, maximum number of occurrences	manipulation object limit
CREATE	constituent or subtree location in document position	content manipulation object position
DELETE	location in document	manipulation object
MODIFY	location in document attribute values, deleting	manipulation object modifications
REPLACE	constituent or subtree location in document	object content manipulation object
COPY	source location in document target location in document position	source manipulation object target manipulation object position
MOVE	source location in document target location in document position	source manipulation object target manipulation object position
RESERVE	location in document	manipulation object
UNRESERVE	location in document, reservation identifier	manipulation object

Annex B

Alignment with DOAM set of operations

(This annex does not form an integral part of this Recommendation | International Standard)

ISO/IEC 10031-1 specifies in subclause 6.6 a standard set of abstract operations for Distributed Office Applications, and gives details of their functionality in Annex K.

The set of operations defined in the Abstract Interface for the manipulation of ODA documents is aligned with the DOAM set, as shown in Table B.1.

Table B.1 – Alignment of operations between Abstract Interface and DOAM

ODA Abstract Interface	DOAM
List	List
Get	Read
Modify	Modify
Copy	Copy
Move	Move
Search	Search
Create	Create
Delete	Delete
Reserve	Reserve
–	Notify
–	Abandon
Open	–
Close	–
Replace	–
BeginGroup	–
EndGroup	–

NOTES

1 Get (ODA Abstract Interface) and Read (DOAM) is the only difference in name.

2 The ODA Abstract Interface Reserve operation is complemented by the Unreserve operation.

Index