

AS/NZS 4167.1:1994
ISO/IEC 10166-1:1991

Australian/New Zealand Standard

Information technology—Text and office systems—Document filing and retrieval

Part 1: Abstract service definition and procedures

[ISO/IEC title: Information technology—Text and office systems—Document filing and retrieval (DFR)
Part 1: Abstract service definition and procedures]

STANDARDS AUSTRALIA ✓

STANDARDS NEW ZEALAND
AVA Documents
Document Sharing Hub

AS/NZS 4167.1:1994

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT/1, Information Systems—Interconnection. It was approved on behalf of the Council of Standards Australia on 15 October 1993 and on behalf of the Council of Standards New Zealand on 22 April 1993. It was published on 21 March 1994.

The following interests are represented on Committee IT/1:

Australian Association of Permanent Building Societies
 Australian Bankers Association
 Australian Bureau of Statistics
 Australian Chamber of Commerce and Industry
 Australian Committee of Directors and Principals
 Australian Computer Society
 Australian Computer Users Association
 Australian Information Industry Association
 Australian Telecommunications Users Group
 Australian Vice Chancellors Committee
 Department of Defence, Australia
 Department of Industry, Technology and Commerce, Australia
 Information Exchange Steering Committee, Australia
 Institute of Information and Communication Technologies, CSIRO, Australia
 Life Insurance Federation of Australia
 OPTUS Communications, Australia
 Telecom Australia

Review of Standards. To keep abreast of progress in industry, Joint Australian/New Zealand Standards are subject to periodic review and are kept up to date by the issue of amendments or new editions as necessary. It is important therefore that Standards users ensure that they are in possession of the latest edition, and any amendments thereto.

Full details of all Joint Standards and related publications will be found in the Standards Australia and Standards New Zealand Catalogue of Publications; this information is supplemented each month by the magazines 'The Australian Standard' and 'Standards New Zealand', which subscribing members receive, and which give details of new publications, new editions and amendments, and of withdrawn Standards.

Suggestions for improvements to Joint Standards, addressed to the head office of either Standards Australia or Standards New Zealand, are welcomed. Notification of any inaccuracy or ambiguity found in a Joint Australian/New Zealand Standard should be made without delay in order that the matter may be investigated and appropriate action taken.

AS/NZS 4167.1:1994

Australian/New Zealand Standard

Information technology—Text and office systems—Document filing and retrieval

Part 1: Abstract service definition and procedures

First published as Joint Standard AS/NZS 4167.1:1994.

PUBLISHED JOINTLY BY:

STANDARDS AUSTRALIA
1 The Crescent,
Homebush NSW 2140 Australia

STANDARDS NEW ZEALAND
6th Floor, Wellington Trade Centre,
181-187 Victoria Street,
Wellington 1 New Zealand

PREFACE

This Standard was prepared by the Joint Australia/New Zealand Standards Committee IT/1, on Information Systems—Interconnection. It is identical with, and has been reproduced from, ISO/IEC 10166-1:1991, *Information technology—Text and office systems—Document Filing and Retrieval (DFR)*, Part 1: *Abstract service definition and procedures*.

This Joint Standard is one of a series of Open Systems Interconnection (OSI) Standards which are currently under development. Since OSI Standards are developmental, there may be some minor difficulties encountered in their implementation. For this reason, Standards Australia will be providing, through the OSI Help Desk, a service to coordinate and disseminate information concerning difficulties which are identified in using this Standard.

Under arrangements made between Standards Australia/Standards New Zealand and the international Standards bodies, ISO and IEC, as well as certain other Standards organizations, users of this Joint Standard are advised of the following:

- (a) In Australia, copyright is vested in Standards Australia. In New Zealand, copyright is vested in Standards New Zealand.
- (b) The number of this Standard is not reproduced on each page; its identity is shown only on the cover and title pages.

For the purpose of this Joint Standard, the ISO/IEC text should be modified as follows:

- (i) *Terminology* The words 'Australian Standard', 'New Zealand Standard' or 'Joint Australian/New Zealand Standard' should replace the words 'International Standard' wherever they appear.
- (ii) *References* The references to International Standards should be replaced by references to the following Australian/New Zealand Standards:

<i>Reference to International Standard or other Publication</i>		<i>Australian/New Zealand Standard</i>	
ISO		AS	
7498	Information processing systems— Open Systems Interconnection—Basic Reference Model	2777	Information processing systems— Open Systems Interconnection—Basic reference model
8613	Information processing—Text and office systems—Office Document Architecture (ODA) and interchange format	3951	Information processing—Text and office systems—Office Document Architecture (ODA) and interchange format
8613-1	Part 1: Introduction and general principles	3951.1	Part 1: Introduction and general principles
8614-4	Part 4: Document profile	3951.4	Part 4: Document profile
8614-5	Part 5: Office Document Interchange Format (ODIF)	3951.5	Part 5: Office Document Interchange Format (ODIF)
8649	Information processing systems— Open Systems Interconnection— Service definition for the Association Control Service Element	3683	Information processing systems— Open Systems Interconnection— Service definition for the Association Control Service Element
8650	Information processing systems— Open Systems Interconnection— Protocol specification for the Association Control Service Element	3684	Information processing systems— Open Systems Interconnection— Protocol specification for the Association Control Service Element

iii

ISO/IEC		AS	
7498	Information processing systems—Open Systems Interconnection—Basic Reference Model	2777	Information processing systems—Open Systems Interconnection—Basic reference model
7498-2	Part 2: Security Architecture	2777.2	Part 2: Security architecture
8824	Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)	3625	Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)
8825	Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)	3626	Information technology—Open Systems Interconnection—Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)
9066	Information processing systems—Text communication—Reliable Transfer	4016	Information processing systems—Text communication—Reliable transfer
9066-1	Part 1: Model and service definition	4016.1	Part 1: Model and service definition
9066-2	Part 2: Protocol specification	4016.2	Part 2: Protocol specification
9072	Information processing systems—Text communication—Remote Operations	3893	Information processing systems—Text communication—Remote Operations
9072-1	Part 1: Model, notation and service definition	3893.1	Part 1: Model, notation and service definition
9072-2	Part 2: Protocol specification	3893.2	Part 3: Protocol specification
9594	Information technology—Open Systems Interconnection—The Directory	4019	Information technology—Open Systems Interconnection—The Directory
9594-1	Part 1: Overview of concepts, models and services	4019.1	Part 1: Overview of concepts, models and services
9594-2	Part 2: Models	4019.2	Part 2: Models
9594-3	Part 3: Abstract service definition	4019.3	Part 3: Abstract service definition
10021	Information technology—Text communication—Message Oriented Text Interchange Systems (MOTIS)	4033	Information technology—Text communication—Message-oriented text interchange systems
10021-3	Part 3: Abstract service definition conventions	4033.3	Part 3: Abstract service definition conventions
		AS/NZS	
10031	Information technology—Text and office systems—Distributed-office-applications model	4109	Information technology—Text and office systems—Distributed-office-applications model
10031-1	Part 1: General model	4109.1	Part 1: General model
10031-2	Part 2: Distinguished-object-reference and associated procedures	4109.2	Part 2: Distinguished-object-reference and associated procedures
10166	Information technology—Text and office systems—Document Filing and Retrieval (DFR)	4167	Information technology—Text and office systems—Document filing and retrieval
10166-2	Part 2: Protocol specification	4167.2	Part 2: Protocol specification

CONTENTS

	<i>Page</i>
Section 1 : General	
1 Scope	1
2 Normative references	2
3 Definitions	4
4 Abbreviations	7
5 Conventions	8
 Section 2 : DFR Abstract Service Definition	
6 DFR Abstract Model	10
7 Abstract-bind and Abstract-unbind Parameters	25
8 Abstract-operations	30
 Section 3 : DFR Attributes	
9 Attribute Definitions	74
 Section 4 : DFR Realization	
10 Supply of the DFR Abstract Service	95
11 Port Realization	98

Annexes	<i>Page</i>
A Overview of Attributes mapping: ODA Document Profile to DFR	99
B Formal Assignment of Object Identifiers	100
C Formal Definition of the DFR Abstract-service	103
D Formal Definition of DFR-Basic-Attribute-Set	119
E Formal Definition of DFR-Extension-Attribute-Set	124
F Introduction for Attributes and Filter	128

© Copyright – STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

Users of Standards are reminded that copyright subsists in all Standards Australia and Standards New Zealand publications and software. Except where the Copyright Act allows and except where provided for below no publications or software produced by Standards Australia or Standards New Zealand may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from Standards Australia or Standards New Zealand. Permission may be conditional on an appropriate royalty payment. Australian requests for permission and information on commercial software royalties should be directed to the head office of Standards Australia. New Zealand requests should be directed to Standards New Zealand.

Up to 10 percent of the technical content pages of a Standard may be copied for use exclusively in-house by purchasers of the Standard without payment of a royalty or advice to Standards Australia or Standards New Zealand.

Inclusion of copyright material in computer software programs is also permitted without royalty payment provided such programs are used exclusively in-house by the creators of the programs.

Care should be taken to ensure that material used is from the current edition of the Standard and that it is updated whenever the Standard is amended or revised. The number and date of the Standard should therefore be clearly identified.

The use of material in print form or in computer software programs to be used commercially, with or without payment, or in commercial contracts is subject to the payment of a royalty. This policy may be varied by Standards Australia or Standards New Zealand at any time.

Information technology—Text and office systems—Document filing and retrieval

Part 1: Abstract service definition and procedures

Section 1: General

1 Scope

This part of ISO/IEC 10166 specifies the Document Filing and Retrieval Abstract Service that enables a user to communicate with a remote Document Filing and Retrieval server (DFR-Server) in order to access a remote document store.

This part of ISO/IEC 10166

- specifies a client-server type model in accordance with the Distributed-Office-Application Model [ISO/IEC 10031-1];
- specifies functions and services provided by Document Filing and Retrieval servers;
- specifies a specific *Document Filing and Retrieval model* for managing documents and groups of documents;
- specifies the Document Filing and Retrieval Abstract Service using the principles established by the Abstract Service Definition Conventions (ISO/IEC 10021-3);
- specifies the usage of other services.

ISO/IEC 10166 serves the following important fields of application:

- supports large capacity document storage for use by multiple users in a distributed system;
- supports ordered filing and multi-key retrieval of documents;
- supports structured organization of groups of documents;
- supports storage of an open-ended number of different document types;
- supports referencing documents and groups ;
- supports filing and referencing of documents outside of the document storage (for example, non-electronic hard copy documents);
- supports the association of attributes to documents, groups, references and search result lists independent of the content;
- supports storage, retrieval and deletion of documents of the document store whatever their content;
- supports searching for, ordering, retrieval, and deletion of single documents or groups of documents using document attributes;
- supports management of different versions of a document, including such concepts as "previous version", "next version" and "last version";
- supports protection against unauthorized storage and retrieval of documents;
- supports the control of concurrent access to DFR objects.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10166. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10166 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498:1984,	<i>Information processing systems - Open Systems Interconnection - Basic Reference Model.</i>
ISO/IEC 7498-2:1989,	<i>Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.</i>
ISO 8613-1:1989,	<i>Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 1: Introduction and general principles.</i>
ISO 8613-4:1989,	<i>Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 4: Document profile.</i>
ISO 8613-5:1989,	<i>Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 5: Office Document Interchange Format (ODIF).</i>
ISO 8649:1988,	<i>Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element.</i>
ISO 8650:1988,	<i>Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element.</i>
ISO/IEC 8824:1990,	<i>Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).</i>
ISO/IEC 8825:1990,	<i>Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).</i>
ISO/IEC 9066-1:1989,	<i>Information processing systems - Text communication - Reliable Transfer - Part 1: Model and service definition.</i>
ISO/IEC 9066-2:1989,	<i>Information processing systems - Text communication - Reliable Transfer - Part 2: Protocol specification.</i>
ISO/IEC 9072-1:1989,	<i>Information processing systems - Text communication - Remote Operations - Part 1: Model, notation and service definition.</i>
ISO/IEC 9072-2:1989,	<i>Information processing systems - Text communication - Remote Operations - Part 2: Protocol specification.</i>
ISO/IEC 9594-1:1990,	<i>Information technology - Open Systems Interconnection - The Directory - Part 1: Overview of concepts, models and services.</i>

- ISO/IEC 9594-2:1990, *Information technology - Open Systems Interconnection - The Directory - Part 2: Models.*
- ISO/IEC 9594-3:1990, *Information technology - Open Systems Interconnection - The Directory - Part 3: Abstract 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 10031-2:1991, *Information technology - Text and office systems - Distributed-office-applications model - Part 2: Distinguished-object-reference and associated procedures.*
- ISO/IEC 10166-2:1991, *Information technology - Text and office systems - Document Filing and Retrieval (DFR) - Part 2: Protocol specification.*

3 Definitions

3.1 General Terminology

3.1.1 The following terms are used with the meanings defined in ISO 7498:

Application Layer
application-entity
Presentation Layer
protocol
service definition

3.1.2 The following terms are used with the meanings defined in ISO/IEC 7498-2:

access control
authentication
authorization
credentials
security policy

3.1.3 The following terms are used with the meanings defined in ISO 8824:

macro

3.1.4 The following terms are used with the meanings defined in ISO 8649:

application context
Association Control Service Element

3.1.5 The following terms are used with the meanings defined in ISO/IEC 9066-1:

Reliable Transfer Service Element

3.1.6 The following terms are used with the meanings defined in ISO/IEC 9072-1:

Remote Operations: bind-operation, unbind-operation, operation
Remote Operation Service Element

3.1.7 The following terms are used with the meanings defined in ISO/IEC 10031:

accessee
accessor
consume-operation
distributed-office-application
data-object-value
distinguished-object-reference
document
produce-operation
privilege-attributes
referenced-object-access
ROA-operation
ROA-protocol

security-attributes
 security-object
 security-subject
 user

3.2 Specific Terminology

For the purpose of ISO/IEC 10166 the following definitions apply:

- 3.2.1 ancestor:** The *parent* of a *DFR-object* and, recursively, any ancestor of the former, including the *DFR-root-group*.
- 3.2.2 attribute-type:** That component of an attribute which indicates the type of information given by the *attribute-value*.
- 3.2.3 attribute-value:** A particular instance of that class of information indicated by an *attribute-type*.
- 3.2.4 attribute-value-assertion:** A proposition, which may be true, false, or undefined, concerning the values of *DFR-attributes* in a *DFR-entry*.
- 3.2.5 conceptual-document:** A set of *DFR-documents*, considered to be "different *versions* of the same document".
- 3.2.6 Control-Attribute-Package:** A collection of attributes used to control access to a *DFR-object*.
- 3.2.7 descendant:** For a given *DFR-group*, any of the *DFR-group-members*, and recursively, any descendant thereof.
- 3.2.8 DFR-attribute:** A data item that identifies a *DFR-object*, describes its *DFR-content*, helps control access to it, or in some other way is associated with the *DFR-object*.
- 3.2.9 DFR-basic-attribute-set:** The set of *DFR-attributes*, that will mandatorily be supported by every *DFR-server*.
- 3.2.10 DFR-content:** The prime information content of a *DFR-object*. The nature of the *DFR-content* depends on the *DFR-object-class* of the *DFR-object*.
- 3.2.11 DFR-document:** A structured amount of information that can be filed, retrieved, and interchanged consisting of a *DFR-document-content* and associated *DFR-attributes*.
- 3.2.12 DFR-document-content:** A body of information actually contained within the document, e.g an office document, and not interpreted by DFR.
- 3.2.13 DFR-document-store:** A named collection of *DFR-objects* which is logically arranged in a hierarchical structure.
- 3.2.14 DFR-entry:** A *DFR-object* together with additional *DFR-attributes* describing its hierarchical place in the *DFR-document-store*.
- 3.2.15 DFR-extension-attribute-set:** The set of *DFR-Attributes* (beyond the *DFR-basic-attribute-set*) which are optionally supported by some *DFR-server*.

- 3.2.16 DFR-group:** A collection of *DFR-objects* in a *DFR-document-store* which are called *DFR-group-members* of the *DFR-group*. A *DFR-group* consists of *DFR-attributes* which are associated with the *DFR-group* as a whole and a *DFR-group-content*.
- 3.2.17 DFR-group-content:** A sequence of UPIs identifying all *DFR-group-members* of the *DFR-group*.
- 3.2.18 DFR-group-member:** A *DFR-object* which is identified in the *DFR-content* of its parent *DFR-group*.
- 3.2.19 DFR-membership-criteria:** A *DFR-attribute* of a *DFR-group* establishing constraints on *DFR-group* membership based on attribute values.
- 3.2.20 DFR-object:** One of a set of information entities managed by a *DFR-server*. *DFR-objects* defined are *DFR-documents*, *DFR-groups*, *DFR-references* and *DFR-search-result-lists*.
- 3.2.21 DFR-object-class:** A *DFR-attribute* indicating the class of a *DFR-object* (*DFR-document*, *DFR-group*, *DFR-reference* or *DFR-search-result-list*).
- 3.2.22 DFR-object-tree:** The *DFR-object-tree* of a *DFR-group* is the tree formed by this *DFR-group* and all its *descendants*.
- 3.2.23 DFR-pathname:** A *DFR-attribute* used to help identify a *DFR-object* in a *DFR-document-store*. The *DFR-pathname* is formed by a sequence of values of the *DFR-title* attribute of all *ancestors* of the *DFR-object* to be identified with the *DFR-title* of the *DFR-object* itself being the last in the sequence.
- 3.2.24 DFR-proper-group:** Any *DFR-group* other than the *DFR-root-group*.
- 3.2.25 DFR-reference:** A *DFR-object* which acts as a link to another *DFR-object*, which is called the *referent* of the *DFR-reference*.
- 3.2.26 DFR-reference-content:** The information stored in a *DFR-reference* for the purpose of identifying the *referent*.
- 3.2.27 DFR-root-group:** The distinguished *DFR-group* within a *DFR-document-store* having no *Ancestor* and whose *DFR-object-tree* encompasses all *DFR-objects* in the *DFR-document-store*.
- 3.2.28 DFR-search-criteria:** A *filter*.
- 3.2.29 DFR-search-result-list:** A *DFR-object* which has information about a set of *DFR-objects* satisfying specified search criteria.
- 3.2.30 DFR-search-result-list-content:** Information about the result of a *DFR Search* abstract operation.
- 3.2.31 DFR-server:** That part of the *DFR* application which supplies Document filing and retrieval services.
- 3.2.32 DFR-Unique-Permanent-Identifier:** A *DFR-attribute* assigned to every *DFR-object* by the *DFR-server* to identify unambiguously a *DFR-object* within the *DFR-document-store*.

- 3.2.33 DFR-user:** The consumer of services supplied by a *DFR-server*. At any time it is acting for a security subject and takes on the privileges of that security subject.
- 3.2.34 filter:** A construct specifying assertions about the presence or value of *DFR-attributes*, it is the same as in Directory (ISO/IEC 9594).
- 3.2.35 member:** see *DFR-group-member*.
- 3.2.36 owner:** A security subject, with owner access right to a specific *DFR-object*.
- 3.2.37 parent:** Each *DFR-object*, except the *DFR-root-group*, is a *DFR-group-member* of a *DFR-group*, which is termed its *parent*.
- 3.2.38 Privilege-Attribute-Certificate:** A certified set of access privileges that can be presented by a *DFR-user* to establish access rights.
- 3.2.39 referent:** That *DFR-Object* to which a *DFR-Reference* refers.
- 3.2.40 version:** *DFR-document* specified by the user as a derivation of one or more other *DFR-documents* by means of specific *DFR-attributes*.

4 Abbreviations

AE	application-entity
ASN.1	Abstract Syntax Notation One
CAP	Control-Attribute-Package
DFR	Document filing and retrieval
DOAM	Distributed-office-application model
DOR	Distinguished-object-reference
DS	DFR-Document-Store
PAC	Privilege-Attribute-Certificate
QoS	Quality of Service
ROA	Referenced-object-access
ROSE	Remote Operations Service Element
UPI	DFR-Unique-Permanent-Identifier

5 Conventions

This part of ISO/IEC 10166 uses the description conventions listed in the following clauses.

5.1 Conventions for Abstract-services

This part of ISO/IEC 10166 uses the following ASN.1-based descriptive conventions for the indicated purposes:

- a) ASN.1 itself, to specify the abstract-syntax of information-objects and their components, common data-types, and state-variables.
- b) The ASN.1 OBJECT and PORT macros and associated abstract-service definition conventions of ISO/IEC 10021-3, to specify the DFR port.
- c) The ASN.1 ABSTRACT-BIND, ABSTRACT-UNBIND, ABSTRACT-OPERATION, and ABSTRACT-ERROR macros of ISO/IEC 10021-3, to specify the DFR abstract-service.
- d) The ASN.1 ATTRIBUTE MACRO and ATTRIBUTE SYNTAX MACRO from ISO/IEC 9594-2, to specify attributes and attribute syntaxes.

NOTE - ASN.1 specifications in this International Standard make full use of ISO 8824:1990 features, especially such syntactical constructs as "WITH COMPONENTS" (subtyping of sequences, sets and choices). All specifications 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.

The DFR as a ROSE-based International Standard does not exploit the Presentation Layer facilities for coping with difference between the local encoding and local syntaxes of each open system (see the use of EXTERNAL in 6.3.2.1).

5.2 Conventions for Text in General

For the terms used in this part of ISO/IEC 10166 the following rules apply:

- a) Single terms beginning with a capital letter and compound terms (chained by hyphens and each word also beginning with a capital letter), are defined terms. For the definitions refer to clause 3 (if it is an attribute see also clause 9); exceptions are titles of other International Standards which also begin with capital letters, see clause 2.
- b) Single terms and compound terms (written together without hyphens) which are rendered in **bold** are either ASN.1-specified data-type names or their component identifiers. For the definitions refer to the annexes or to the corresponding sections in the main text.

The following characters are used in this part of ISO/IEC 10166 to indicate whether a parameter, an attribute or other items described are mandatory, optional or conditional. That is:

- M (Mandatory) stands for the condition that an item shall be present in any case (shall be supported by DFR);

- **O** (Optional) stands for the condition that an item shall be present at the discretion of a DFR entity;
- **C** (Conditional) stands for the condition that an item shall be present under some circumstances defined in this part of ISO/IEC 10166.

Section 2 - DFR Abstract Service Definition

6 DFR Abstract Model

This clause provides an abstract functional model of Document Filing and Retrieval. For an introduction and description of the abstract-service concept and its definition conventions, see ISO/IEC 10021-3.

The Document Filing and Retrieval environment comprises two atomic objects, the Document Filing and Retrieval-Server (DFR-Server) and the Document Filing and Retrieval-User (DFR-User). A DFR-Server is modelled as an atomic object, which acts as a provider of services to the DFR-User. The DFR-Server is described using an abstract model in order to define the service provided by the DFR-Server - the Document Filing and Retrieval abstract service. Figure 1 shows the DFR model.

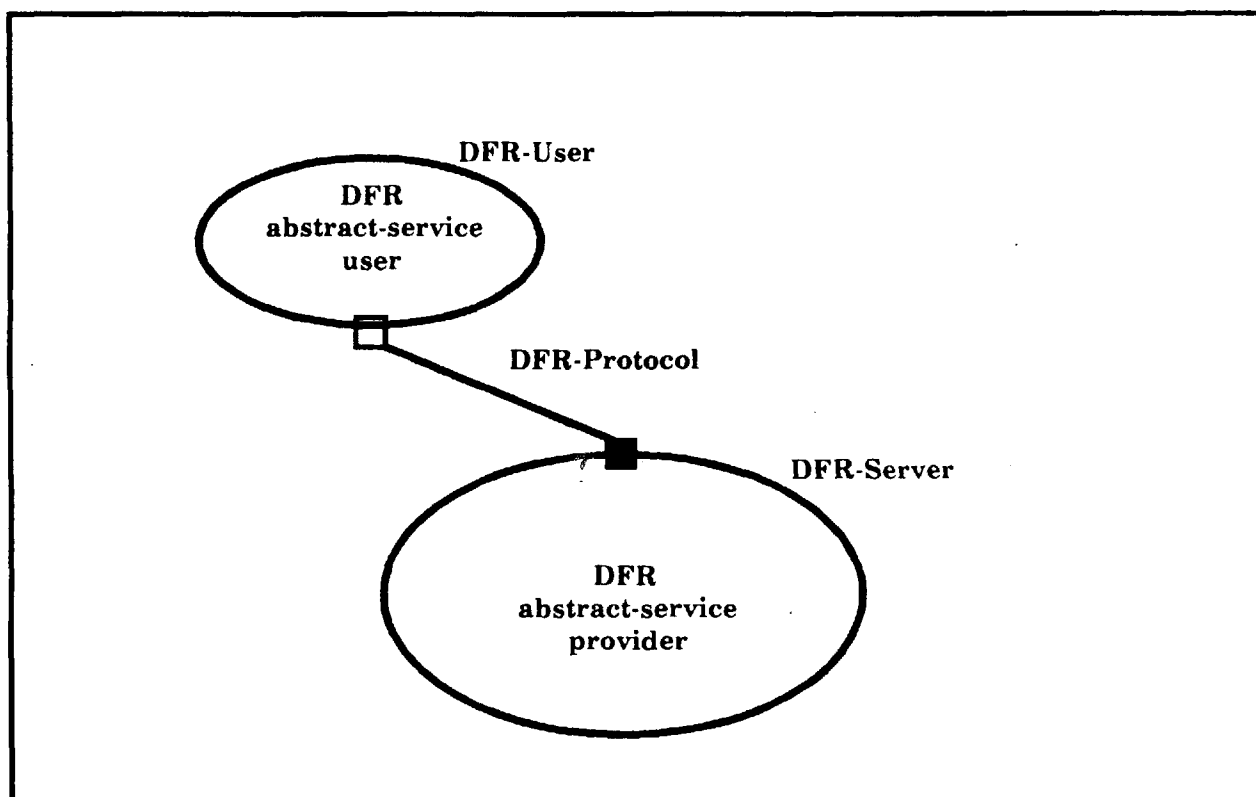


Figure 1 - Document Filing and Retrieval Abstract-service

6.1 Objects in DFR Environment

The DFR-Server is modelled as an atomic object. It supplies the DFR port abstract-services to the DFR-User.

The formal definition of the DFR-Server object is as follows:

```

dfr-server    OBJECT
              PORTS { dfr-port [S]}
              ::= id-dfr-server

```

The DFR-User is modelled as a separate object. The DFR-User consumes the DFR port abstract-services supplied by the DFR-Server.

```

dfr-user      OBJECT
              PORTS { dfr-port [C] }
              ::= id-dfr-user

```

The DFR port is the concern of this part of ISO/IEC 10166 for operations on DFR-Objects.

6.2 DFR Port

A DFR-User is joined to, and interacts with, a DFR-Server by means of the DFR port. The collection of capabilities provided by this port forms the DFR-Server abstract-service. These capabilities include obtaining information on, fetching, and deleting documents residing in the DFR-Server.

By means of the bind-operation, the DFR-Server authenticates a user by checking the user's credentials or certified identity and access privileges (PAC) before providing it with any of the filing and retrieval capabilities.

This asymmetric abstract port is defined for any DFR-User (in the consumer role) and for any DFR-Server (in the supplier role). Such a pair of abstract ports makes it possible for any DFR-User to communicate with a DFR-Server in order to use ordinary DFR-operations. All these operations are also asymmetrical. Each of them is invoked by the consumer (the DFR-User) and performed by the supplier (the DFR-Server).

The DFR port is defined as follows:

```

Dfr PORT
  CONSUMER INVOKES {
    Create,
    Delete,
    Copy,
    Move,
    Read,
    Modify,
    List,
    Search,
    Reserve,
    Abandon    }
  SUPPLIER INVOKES { }
  ::= id-pt-dfr

```

6.3 Information Model

A DFR-Server offers its users operations on DFR-Objects in its DFR-Document-Store. The object classes of DFR-Objects are DFR-Documents, DFR-Groups, DFR-References, and DFR-Search-Result-Lists. A DFR-Group is either a DFR-Root-Group or a DFR-Proper-Group. This gives the following specification for all DFR-Object classes.

```
DfrObjectClass ::= ENUMERATED {
    dfr-document      (0),
    dfr-root-group    (1),
    dfr-proper-group  (2),
    dfr-reference      (3),
    dfr-search-result-list (4) }
```

A DFR-Document-Store is accessed through a DFR-Server. A DFR-Document-Store is assigned to one DFR-Server only. A DFR-Server manages one DFR-Document-Store. A DFR-Document-Store is a named collection of DFR-Objects which is logically arranged in a hierarchical structure. An example of how different DFR-Objects in a DFR-Document-Store are related to each other is illustrated in Figure 2 according to the definitions below. Group membership is depicted in the figure by solid lines. A DFR-

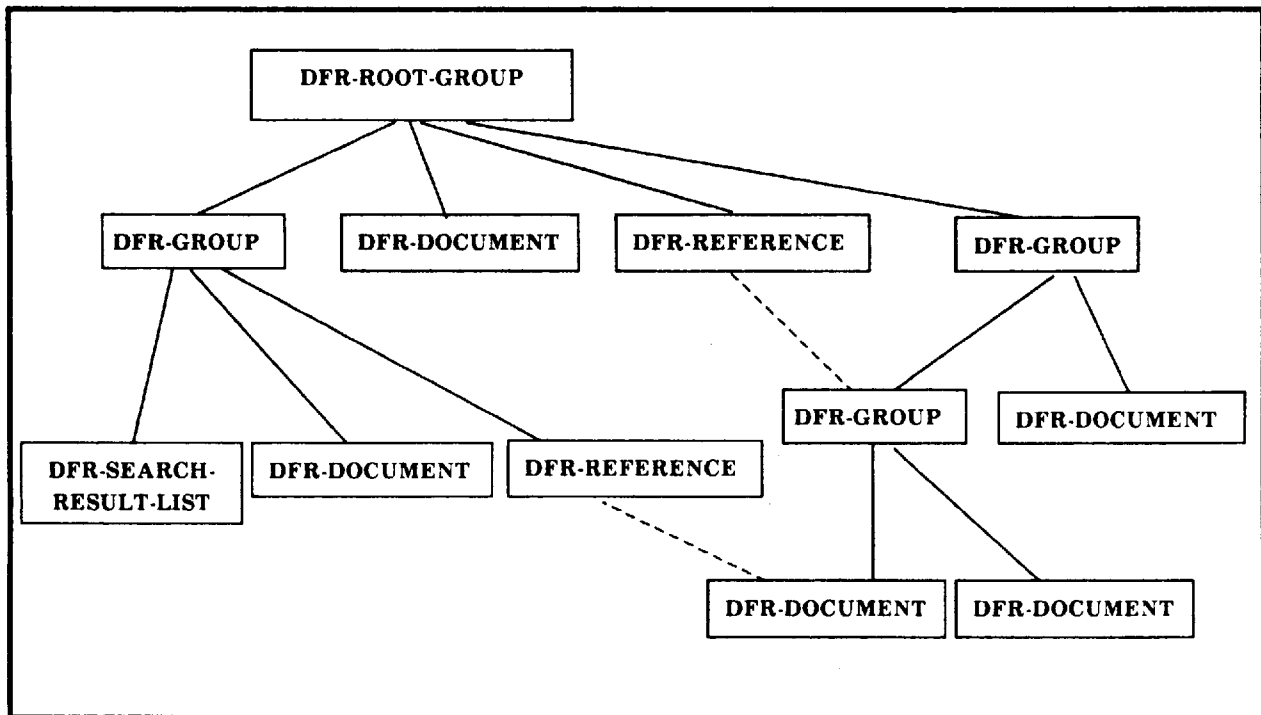


Figure 2 - Example DFR Document Store structure

Object is a Member of one and only one DFR-Group (parent group), an exception is the DFR-Root-Group that has no parent group. A DFR-Object can participate indirectly in more than one DFR-Group by means of DFR-References. A DFR-Reference refers to one and only one DFR-Object. A reference to a DFR-Reference is not permitted. A DFR-Object can be referenced from more than one DFR-Reference. A DFR-Group can be viewed as the root of a DFR-Object-Tree consisting of all descendants of that DFR-Group. A DFR-Search-Result-List contains information about a set of DFR-Objects satisfying some selection criteria.

A DFR-Object consists of DFR-Attributes and DFR-Content. A DFR-Object is introduced into a DS by creating a DFR-Entry for it. A DFR-Entry is formed by a DFR-Object together with additional DFR-Attributes describing its hierarchical place in the DFR-Document-Store. The immediately containing DFR-Group is the Parent of the DFR-Object, and the DFR-Object is a Member of that DFR-Group.

```
DfrEntry ::= SEQUENCE {
    attributes    [0] DfrEntryAttributes,
    content       [1] DfrObjectContent }
```

The attributes of a DFR-Entry are those of the corresponding DFR-Object plus two supplementary attributes, DFR-Parent-Identification and DFR-Pathname.

The attributes forming the **DfrEntryAttributes** are defined either in this part of ISO/IEC 10166, or may be defined externally. This is modelled by attribute sets, a basic attribute set and extension attribute sets. Two attribute sets are defined in this International Standard, the mandatory DFR-Basic-Attribute-Set and one optional DFR-Extension-Attribute-Set derived from the ODA-Document-Profile. Other optional extension attribute sets, for example one extension attribute set which includes security-attributes, may be defined elsewhere. The DFR-Basic-Attribute-Set is contained in the abstract syntax of the DFR access protocol. Each extension attribute set requires at least one supplementary abstract syntax. The abstract syntax of the DFR-Extension-Attribute-Set is defined in Annex E.

NOTE 1 Negotiation of extension attribute sets is performed by means of negotiation of the corresponding abstract syntax during the bind-operation.

```
DfrEntryAttributes ::= SET OF Attribute
```

The **DfrObjectContent** is the actual information stored with the DFR-Object. The nature of this information depends on the **DfrObjectClass**. The DFR-Content of a DFR-Group is a sequence of UPIs of all its Members. The DFR-Content of a DFR-Document is a body of information, e.g. an office document. The DFR-Content of a DFR-Reference is a pointer to some other DFR-Object (DFR-Group or DFR-Document or DFR-Search-Result-List), called the Referent.

```
DfrObjectContent ::= CHOICE {
    document-content      [0] DfrDocumentContent,
    root-group-content    [1] DfrGroupContent,
    proper-group-content  [2] DfrGroupContent,
    reference-content     [3] DfrReferenceContent,
    search-result-list-content [4] DfrSearchResultListContent }
```

Each DFR-Object has a unique identification within the DFR-Document-Store. This identification is given by a DFR-Unique-Permanent-Identifier (UPI). A UPI is assigned to each DFR-Object by the DFR-Server. Once assigned by a DFR-Server, the value of each UPI will not be changed within the life of a DFR-Object. In addition, this UPI value will differ from that of all DFR-Objects which once existed in the same DFR-Server (including all existing DFR-Objects and all deleted DFR-Objects).

```
DfrUniquePermanentIdentifier ::= OCTET STRING
```

NOTE 2 UPI is also modelled as a DFR-Attribute, see 9.2.1.

6.3.1 DFR-Document-Store

A DFR-Document-Store (DS) is a collection of DFR-Objects logically arranged in a hierarchical structure.

All DFR-Objects in a DFR-Document-Store are accessible via the DFR-Server.

Any DFR-Object within a DS can be accessed via its UPI or via the DFR-Pathname. The UPI is assigned to a DFR-Object by the DFR-Server when the object is stored in a DFR-Document-Store in order to identify unambiguously the DFR-Object. The DFR-Pathname is formed by a sequence of values of the DFR-Title attribute from all Ancestors of the DFR-Object to be identified with the DFR-Title of this DFR-Object being the last in this sequence. Since the DFR-Title attribute is only unique if such a restriction is defined for the DFR-Document-Store, access via the DFR-Pathname may fail.

6.3.2 DFR-Documents

A DFR-Document is a structured amount of information that can be interchanged, stored, and retrieved. A DFR-Document consists of a DFR-Document-Content along with DFR-Attributes which are associated with the DFR-Content. A DFR-Document is contained in one DFR-Document-Store. Maintenance of consistency between DFR-Documents and their copies in different DFR-Document-Stores is outside the scope of this part of ISO/IEC 10166, and left to the responsibility of the DFR-User.

DFR-Attributes are additional data items used to identify and to describe properties of a DFR-Document. The DFR-Document-Content is the actual information stored in the DFR-Document, and is not interpreted by the DFR-Server.

6.3.2.1 DFR-Document-Content

The DFR-Document-Content is a body of information that has been provided to a DFR-Server for the purpose of storage.

```
DfrDocumentContent :: = EXTERNAL (WITH COMPONENTS { ... , direct-reference PRESENT,
                                                    indirect-reference ABSENT,
                                                    encoding (WITH COMPONENTS { ... , arbitrary ABSENT } ) } )
```

The direct-reference component is an OBJECT IDENTIFIER whose value is the same as the value of the DFR-Document-Type attribute of the DFR-Document (see 9.2.3).

Upon request a DFR-Server transfers the content of a DFR-Document to the DFR-User. A DFR-Server never interprets the content of the documents it stores.

NOTE - The co-operation of DFR with a document content access protocol may be the subject of future standardization.

6.3.2.2 DFR-Document Attributes

DFR-Document attributes are data items that identify a DFR-Document, describe its content, help control access to it, or are in some other way associated with the document.

DFR-Attributes serve to qualify or enhance a DFR-Document's content, or to define additional characteristics of the document or its intended use. The values of certain DFR-Attributes have a

particular meaning to the DFR-Server and elicit defined behavior while others are DFR-User defined and controlled.

6.3.3 DFR-References

A DFR-Reference allows a DFR-Object to participate in more than one DFR-Group without requiring distinct copies of that DFR-Object to be created. A DFR-Reference consists of a DFR-Content containing a "pointer" to the referenced DFR-Object (the Referent) and of DFR-Attributes (see 6.3.3.2).

The Referent can be a DFR-Document, a DFR-Group or a DFR-Search-Result-List; it shall not be another DFR-Reference. The DFR-Object-Class of the Referent is noted in the DFR-Reference-Content.

A DFR-Reference refers either to a DFR-Object within the same DFR-Document-Store or to a DFR-Object in a different DFR-Document-Store.

A DFR-User with read access permission (see 6.3.8.3) to the DFR-Reference can also access the Referent by using the DOR stored in the DFR-Reference-Content. In the case of a local access to the Referent (i.e. DFR-Reference and Referent are in the same DS) only those DFR-Users who are noted in the DFR-Access-List attribute of the Referent can access the Referent. In the case of a remote access to the Referent (i.e. DFR-Reference and Referent are in different DS) the access permission can only be verified by the optional token in the DOR (see ISO/IEC 10031-1).

If the DFR-Server detects at any time that the Referent has been deleted, it does not delete the DFR-Reference, but sets to true the DFR-Referent-Deleted attribute of the DFR-Reference.

The content of a DFR-Reference is structured like a Distinguished Object Reference (see ISO/IEC 10031-2) to simplify its use in an ROA-operation (see ISO/IEC 10031-1). The Distinguished Object Reference (DOR) is used by DFR in two different ways:

- a) Use of a DOR as the content of a DFR-Reference: A DOR to an entire DFR-Object can be produced at the DFR-User's request by the DFR-Server. This DOR can then be stored in the content of a DFR-Reference at the DFR-User's request. This DFR-Reference can later be used in an ROA-operation. The DFR-User can also later read the content of this DFR-Reference and transfer it to another application for subsequent access to the referenced DFR-Object by that application.

The DOR can be stored in a DFR-Reference of the same DFR-Document-Store that contains also the referenced DFR-Object or in another DFR-Document-Store. In the first case, the content of the DFR-Reference is a DOR whose **ae-identifier** and application sub-components point to the DFR-Server which manages both Referent and reference.

- b) Use of a DOR for immediate transfer: A DOR to an entire DFR-Object, to the attributes of a DFR-Object, or to the content of a DFR-Object can be produced at the DFR-User's request by the DFR-Server of the Referent, to be used for immediate transfer of the referenced data-object-value to some other DFR-Server or to another application. If some DFR data-object-value is to be transferred by an ROA-operation to another application, it is necessary that the latter be able to interpret this DFR data-object-value (for example the print application might only accept a DFR-Document-Content).

6.3.3.1 DFR-Reference Content

The DOR stored in the content of a DFR-Reference contains among others a pointer to the Referent (local reference) and an identifier (data-object-type) of the Referent (to indicate whether the Referent is a DFR-Document, a DFR-Group or a DFR-Search-Result-List).

The ASN.1 specification of a DFR-Reference-Content is as follows:

```

DfrReferenceContent ::= DOR
    (WITH COMPONENTS (
        ae-identifier,
        local-reference,
        data-object-type (DfrObjectClassID (
            id-dfr-document |
            id-dfr-root-group |
            id-dfr-proper-group |
            id-dfr-search-result-list)),
        quality-of-service,
        token ABSENT    ))

DfrObjectClassID ::= OBJECT IDENTIFIER (
    id-dfr-document |
    id-dfr-root-group |
    id-dfr-proper-group |
    id-dfr-search-result-list )

```

The semantics of the different components of the DFR-Reference-Content (the DOR) are as defined in ISO/IEC 10031-2.

6.3.3.2 DFR-Reference Attributes

DFR-Reference attributes are data items associated with a DFR-Reference.

The DFR-Reference attributes are explicitly associated with the DFR-Reference; they can differ from attributes of the Referent. The semantics of some DFR-Reference attributes however, will normally, at the users discretion, be related to the Referent; in this way they provide additional information about the latter. Some other DFR-Reference attributes are solely related to the DFR-Reference itself.

Certain attributes have a particular meaning to a DFR-Server and elicit defined behaviour while others are user defined and controlled.

6.3.3.3 Support of the ROA-model by DFR

This clause describes the support of the referenced-object-access functional model (ROA-model) by DFR. The ROA-model is described in ISO/IEC 10031-1. DFR uses the DOR as defined in ISO/IEC 10031-2.

6.3.3.3.1 Role of DFR in the ROA-model

A DFR-Server shall perform both the accessee role and the accessor role in the ROA-model:

- 1) The DFR-Server as an accessee performs the following role as defined in the ROA-model.
 - a) The DFR-Server accepts produce-operations.
 - b) The DFR-Server accepts certain ROA-operations.
- 2) The DFR-Server as an accessor performs the following role as defined in the ROA-model.
 - a) The DFR-server accepts consume-operations.
 - b) The DFR-Server can issue certain ROA-operations.

6.3.3.3.2 Produce-operations in DFR

A DFR abstract operation is termed a produce-operation as defined in ISO/IEC 10031-1 if a DFR-User requests the transfer of a reference to the data-object-value in that operation, that is the DFR-Server returns a DOR instead of the data-object-value. A DFR-User can request a DOR in the result of the following DFR abstract operations:

- **Create**
- **Copy**
- **Move**
- **Read**
- **Modify**

At the time of a produce-operation, DFR-Users can request the required Quality-of-Service (QoS) by the Requested-QoS-level parameter. The DFR-Server determines its available QoS and sets the appropriate values in the resulting DOR.

NOTE - It is outside the scope of this International Standard to define whether the capability to change a QoS associated with a DOR is supported or not.

6.3.3.3.3 Consume-operations in DFR

A DFR abstract operation is termed a consume-operation as defined in ISO/IEC 10031-1 if a DFR-User either provides a DOR instead of a data-object-value to the DFR-Server or provides an identifier (DfrEntryName) for a DOR already stored (in a DFR-Reference) in the DFR-Server.

A DOR can be included in the following DFR abstract operations:

- **Create**

- **Copy**
- **Move**
- **Read**
- **Modify**

After receiving the consume-operation, the DFR-Server performs the accessor role of an ROA-operation by issuing an appropriate ROA-protocol abstract operation acceptable to the accessee. How the DFR-Server knows the appropriate ROA-protocol and ROA-operation to use when accessing the accessee is outside the scope of this part of ISO/IEC 10166. After receiving the result of the ROA-operation, the DFR-Server performs the requested original consume-operation as if the data-object-value was included in the original abstract operation.

6.3.4 DFR-Groups

A DFR-Group is a collection of DFR-Objects in a DFR-Document-Store which are called DFR-Group-Members of the DFR-Group. A DFR-Group consists of DFR-Attributes which are associated with the DFR-Group as a whole and a DFR-Group-Content which is a sequence of UPIs of all Members of the DFR-Group.

A DFR-Group can be either a DFR-Root-Group or a DFR-Propor-Group. A DFR-Propor-Group itself is always a Member of some other DFR-Group (Parent). A DFR-Root-Group is not a Member of any other DFR-Group; each DFR-Document-Store contains only one DFR-Root-Group. Any DFR-Object of a given DS can be reached from the DFR-Root-Group of that DS. The creation (deletion) of the DFR-Root-Group can not be performed by the user of the Document Filing and Retrieval port.

NOTE - The creation and deletion of the DFR-Root-Group including the first registration of Owner(s) and other security subjects in the DFR-Access-List (see 6.3.8) is performed by some administration authority. This administration authority is not defined in this part of ISO/IEC but left to local implementation.

6.3.4.1 DFR-Group-Content

The DFR-Group-Content is a sequence of UPIs of all Members of the DFR-Group.

Members of a DFR-Group can be DFR-Documents, DFR-Groups, DFR-References and DFR-Search-Result-Lists.

The ordering of the DFR-Group-Members is as defined in the DFR-Ordering attribute (see 9.2.9) of the DFR-Group.

DfrGroupContent :: = SEQUENCE OF DfrUniquePermanentIdentifier

A DFR-Group can be assigned a DFR-Membership-Criteria attribute. The membership criteria govern membership in the group; new Members are required to satisfy the membership criteria before being added. The attributes of the new Member are verified against the membership criteria and the addition is denied if the verification fails. A group having no specified membership criteria permits any new direct Member to be added.

When a request to change the membership criteria of a DFR-Group will cause any conflict with the membership for one or more current Members of the group, such a request shall be rejected.

The membership criteria for a group apply only to Members. That means, if a DFR-Reference is a Member of a given DFR-Group, only the attributes of this reference are the criteria to be verified for

the membership in this DFR-Group; it is not relevant for the reference's membership in the DFR-Group if the attributes of the corresponding Referent are changed and thus no longer satisfy the membership criteria.

The Members of a DFR-Group can be enumerated by a **List** abstract operation and their attributes examined. When a group is enumerated, the attributes returned for a reference Member are never those of the Referent.

The DFR-Object-Tree of a DFR-Group is formed by the DFR-Group itself and all its Descendants. These Descendants are either Members of this DFR-Group or Members of other Descendants of this DFR-Group, recursively. Some DFR abstract operations apply to the entire DFR-Object-Tree of a DFR-Group rather than to its DFR-Group-Members only.

6.3.4.2 DFR-Group Attributes

DFR-Group attributes are data items that identify a group, describe its content, help control access to it, or are in some other way associated with the group.

DFR-Group attributes serve to qualify or enhance a group's content, or to define additional characteristics of the group or its intended use. Certain DFR-Attributes have a particular meaning to a DFR-Server and elicit defined behavior while others are user defined and controlled.

6.3.5 DFR-Search-Result-List

A DFR-Search-Result-List is a DFR-Object intended exclusively for holding results of a **Search** abstract operation.

A DFR-Search-Result-List can only be created with empty content, which is filled in as result of a **Search** abstract operation. The user can read or list a DFR-Search-Result-List, as well as modify its attributes, but the content can only be modified by another **Search** abstract operation. A **Search** abstract operation can also be done without specifying new search criteria (thus updating the content of an existing DFR-Search-Result-List).

6.3.5.1 DFR-Search-Result-List Content

The DFR-Content of a DFR-Search-Result-List holds a sequence of the DFR-Attributes UPI and DFR-Object-Class from all DFR-Objects satisfying the search expression specified in the Filter of a **Search** abstract operation. It also contains additional information regarding the domain and criteria of the search as well as the time and conditions of the most recent **Search** abstract operation execution.

```
DfrSearchResultListContent ::= CHOICE { empty      NULL,
                                     produced  SEQUENCE {
start-date-and-time  [0] GeneralizedTime,
end-date-and-time    [1] GeneralizedTime,
object-list          [2] DfrEntryList,
ordering             [3] OrderingRule OPTIONAL,
search-domain        [4] SearchDomain,
search-criteria      [5] SearchCriteria } }
```

A **DfrSearchResultListContent** can only be "produced" by a **Search** abstract operation; at the DFR-Search-Result-List creation time it is empty. There is no possibility to change the DFR-Content of a DFR-Search-Result-List except by a re-execution of the **Search** abstract operation in such a way as either to replace or to append to its existing content. **DfrSearchResultListContent** will always be

placed in an existing DFR-Search-Result-List (produced by a previous **Create** abstract operation) which is either empty, i.e. never used, or not empty, i.e. used in a previous **Search** abstract operation. In the latter case the search criteria and search domain stored in the **DfrSearchResultListContent** can either be reused (e.g. for search continuation) or overwritten by the search criteria and search domain used in the **Search** abstract operation.

The ASN.1 types of individual components of **DfrSearchResultListContent** are formally defined in 8.1.6.

6.3.5.2 DFR-Search-Result-List Attributes

DFR-Search-Result-List attributes are data items that identify a DFR-Search-Result-List, describe its content, help control access to it, or are in some other way associated with the DFR-Search-Result-List.

DFR-Search-Result-List attributes serve to qualify or enhance a DFR-Search-Result-List-Content, or to define additional characteristics of the DFR-Search-Result-List or its intended use. Certain DFR-Attributes have a particular meaning to a DFR-Server and elicit defined behavior while others are user defined and controlled.

6.3.6 DFR Version management

The DFR Version management is achieved by a combination of specific DFR-Attributes, managed by the DFR-Server in a predefined way, in order to make it possible for the user to have a very flexible user-defined Version structure, and to provide the user with DFR-Server assistance while navigating through this structure or attempting to modify it. Briefly, the DFR Version management may be qualified as user-defined and server assisted.

A DFR-Document can have several versions, simultaneously present in a DFR-Document-Store. In fact, each "version of a document" is itself an individual DFR-Document (an individual entry in the DFR-Document-Store). The set of all DFR-Documents considered to be "different versions of the same document", defines a Conceptual-Document. Different DFR-Documents which are versions of the same Conceptual-Document are related by means of two DFR-Server maintained attributes, DFR-Next-Versions and DFR-Previous-Versions. The set of all DFR-Documents which are versions of one Conceptual-Document is structured as a directed graph with regard to these two DFR-Attributes. The relationships among different versions of a Conceptual-Document are independent of the group structure of the DFR-Document-Store.

All versions of the same Conceptual-Document are distinguished from all other DFR-Documents by means of a special DFR-Server managed attribute DFR-Version-Root. The value of this attribute is the UPI of the (historically) first version of the Conceptual-Document; this value remains valid and unchanged even if this first version is later deleted (because it is no more considered, in this context, as the UPI of the first version, but is instead identifying the overall Conceptual-Document). Versions of the same document can also have other attribute values in common, but this is not mandatory and is not verified by the DFR-Server.

In contrast, different versions of the same (conceptual) document are distinguished, apart from their UPIs, by another special attribute, called Version-Name. This is a DFR-User-specified attribute, intended for the user's perception; the DFR-Server can only optionally reinforce the uniqueness of its values among all versions of the same document.

When a DFR-User creates a DFR-Reference to a multi-version document, this DFR-Reference always points to a particular version (that is, to one specific DFR-Document in the DFR-Document-Store).

In the simplest case, all versions of the same document are linearly ordered: each version except the first has exactly one previous version, and each version except the last has exactly one next version. In this case the latest version of a Conceptual-Document can be reached by navigating through it using the DFR-Next-Versions attribute.

In a more general case, a given version of a Conceptual-Document can have several next versions (tree model). In addition, a version can be declared following more than one previous versions (directed graph model). Loops are not possible even in this case, as versions are created sequentially, and each version can be linked only to existing ones (a DFR-Previous-Versions attribute shall not be modified for any Document which has a next version).

It is always the DFR-User's action to define some existing or newly created DFR-Document as a new version of some other DFR-Document or DFR-Documents; the latter are explicitly specified by the user as previous version(s) for the newly defined one. The DFR-User shall have at least read access right to the DFR-Document(s) specified as previous version(s). Links from the new version to all its predecessors (i.e. their UPIs) are the values of the attribute DFR-Previous-Versions.

At the same time, links in the opposite direction are created: each time a new version is created as a successor of an existing one, the DFR-Next-Versions attribute of the latter contains the link to the successor as a supplementary value.

When "discarding" an existing version, all links to it from its successors are redirected to point to each of its predecessors, and all links to it from its predecessors are redirected to point to each of its successors. An existing version can be "discarded" either by deletion of the whole DFR-Document or by deletion of the attribute DFR-Previous-Versions of the DFR-Document in which case the DFR-Server also deletes the attributes DFR-Next-Versions and DFR-Version-Root.

Version management applies only to DFR-Documents; versions of DFR-Groups, or of DFR-References, or of DFR-Search-Result-Lists, are not defined.

6.3.7 Attributes and Filter

DFR uses **Attribute**, **ATTRIBUTE MACRO**, **ATTRIBUTE-SYNTAX MACRO**, **Filter** and **FilterItem** as defined in ISO/IEC 9594.

Each DFR-Object consists of two parts, a set of attributes, and a "content" entity. DFR-Attributes are managed independently of the content. Attributes can be read and changed. If an attribute is changed the content of that object is not changed. Attributes characterize an object, that is each attribute provides a piece of information about, or derived from, the object to which it corresponds. Attributes affect storage and retrieval of an object and control access to it.

For the convenience of the reader a shortened description of Attributes and Filter is in annex F. The complete definition is in ISO/IEC 9594.

For an overview of the DFR-Attributes see clause 9.

NOTE - Some selected attributes are intended for ordering the objects by the value of these attributes. In the case of textual attributes ordering presupposes the existence of some collating sequences. The definition of collating sequences is outside the scope of this part of ISO/IEC 10166.

6.3.8 Security in DFR

Two specific security mechanisms are addressed here: authentication and access authorization.

6.3.8.1 Authentication

At bind time the security subject represented by the DFR-User is authenticated either by using password(s) or in the case of a user who has been previously authenticated elsewhere, by checking a certified identity. The authentication mechanism verifies the credentials of the DFR-User requesting access to the DFR-Document-Store. This does not, however, qualify the DFR-User to access all DFR-Objects stored in the DFR-Document-Store. Certified identities can also be presented subsequent to the bind-operation, as arguments to individual abstract operations. This permits the bind to be shared between multiple security subjects, the DFR-User representing a different security subject for each of the abstract operations concerned. The DFR-User is not allowed multiple binds, at the same time, to the same DFR-Server.

6.3.8.2 Access authorization

Access authorization is controlled by security attributes collectively known as a Control-Attribute-Package (CAP).

The CAP is a collection of security related attributes, assigned to DFR-Objects. In order to provide the design freedom necessary for DFR-Servers to operate under a wide variety of security policies, the DFR implementer should be free to choose control attribute types appropriate to the security regimes to which the implementer is targeted. In the longer term standards will be defined for access authorization rules and for control attribute types, from which implementers will make their choice. Particular examples of such types might be the DFR object's security classification, integrity level or other access group category.

In the short term these standards are not available, and this part of ISO/IEC 10166 defines one specific control attribute, the DFR-Access-List.

If the accessing DFR-User has presented a Privilege-Attribute-Certificate (PAC) containing privilege attributes for a DFR-Object, to determine what access the user has, the privilege attributes will be used where appropriate in conjunction with the attributes in the CAP. Privilege attributes are optionally presented either at abstract bind time via a PAC type credentials argument (see 7.1.1) or explicitly at the time of the abstract operation request in the privileges argument (see 8.1.3.5), or both. The DFR-User can also specify a proxy PAC in a specific abstract operation request. This proxy PAC is used by the DFR-Server to access another server on behalf of the requesting DFR-User (see 8.1.3.5).

Each specific DFR realisation shall use either the DFR-Access-List attribute, or other CAP attributes, according to some defined matching algorithms.

6.3.8.3 DFR-Security Policy

All DFR-Objects in a DFR-Document-Store have one or more Owners. An Owner is a security subject with specific privileges with regard to the owned DFR-Object.

Each Owner of a DFR-Object is noted in the DFR-Access-List attribute of the CAP of that DFR-Object. An Owner of a DFR-Object can add further Owners or delete existing ones. Attempts to delete the last Owner result in an error.

The DFR-Access-List attribute defines the list of security subjects allowed to access a specific DFR-Object, together with their access rights. A security subject registered in the DFR-Access-List (of the CAP) of a DFR-Object is allowed, according to the registered access rights, to access that DFR-Object. A DFR-Object will not be made visible to the DFR-User unless at least read access was granted by the Owner of this DFR-Object. For example, if a DFR-Object satisfies a filter condition specified in a **Search** abstract operation, it will not be made visible to a DFR-User that does not have read access permission to this DFR-Object. The DFR-Access-List is an attribute of the DFR-Object. The contents of the DFR-Access-List attribute can be modified only by an Owner. A DFR-User having only read access right to a DFR-Object can only read its own access rights to that DFR-Object. A DFR-Access-List consists of one or more elements. Each element consists of two parts:

```
DfrAccessListElement :: = SEQUENCE {
    access-id      AccessId,
    access-rights  AccessRights }
```

```
AccessId :: = DistinguishedName
```

NOTE - In the case of a DFR-User having only read access to the DFR-Object, the DFR-Access-List returned differs from the DFR-Access-List assigned to the DFR-Object, because it contains only the users own **AccessId** and **AccessRights**.

```
AccessRights :: = ENUMERATED {
    read                (0),
    extended-read       (1),
    read-modify         (2),
    read-modify-delete  (3),
    owner               (4) }
```

Read access to a DFR-Object allows a DFR-User to access that DFR-Object, using **Read**, **Copy**, **Search**, and **List** abstract operations. These abstract operations are permitted to both the content and attributes of the DFR-Object, except those parts of the DFR-Access-List attribute not related to this DFR-User and the attributes DFR-Created-By, DFR-Attributes-Modified-By, DFR-Content-Modified-By, DFR-Reserved-By. Attempts to read the DFR-Access-List attribute returns only the initiating DFR-User's access rights. Even if a DFR-Object satisfies the search criteria specified in the **Search** abstract operation, that DFR-Object is not listed in the resulting DFR-Search-Result-List unless the initiating DFR-User has read access permission to the DFR-Object. The DFR-Object will be enumerated in the result of a **List** abstract operation only if the initiating security subject is specified in its DFR-Access-List attribute.

Extended-read access to a DFR-Object includes all privileges of read access and also permits the user to read the entire DFR-Access-List attribute of that DFR-Object and all other attributes.

Read-modify access to a DFR-Object allows a DFR-User to apply an uncommitted reserve or unreserve to that DFR-Object and modify its content and/or attributes with the exception of the attribute DFR-Access-List. Modification of the content of a DFR-Group means new Members can be inserted in the group; operations on existing Members in the DFR-Group depend on the rights noted in their DFR-Access-List attributes. Read-modify access includes extended-read access.

Read-modify-delete access to a DFR-Object allows a DFR-User to delete or move that DFR-Object. In the case of a DFR-Group a DFR-User with the read-modify-delete access right to this DFR-Group can move the whole DFR-Object-Tree of this DFR-Group regardless of the access rights in the DFR-Access-Lists of the Descendants, but the DFR-User can delete the whole DFR-Object-Tree of this DFR-Group or any Descendant of the DFR-Group only if this DFR-User is noted with read-modify-delete access permission in the DFR-Access-List of all Descendants if the whole DFR-Object-Tree of this DFR-Group

shall be deleted or in the DFR-Access-List of a Descendant if this particular Descendant shall be deleted. Read-modify-delete access includes read-modify access.

Owner access to a DFR-Object allows a DFR-User to modify the DFR-Access-List attribute of that DFR-Object and a committed reservation can be applied on that DFR-Object. Owner access includes read-modify-delete access. A DFR-User creating a DFR-Object (by **Create** or **Copy** abstract operations) is automatically included in the DFR-Access-List attribute of the new DFR-Object as the Owner.

If a DFR-User does not have at least read access permission to a DFR-Group, this user cannot access its Descendants via this DFR-Group. Direct access to a Descendant object via the object's UPI can still be granted depending upon the DFR-Access-List attribute in place on the DFR-Object. Operations involving access to more than one DFR-Object require the appropriate access right to each of these objects.

NOTE - The constraints concerning access of a Referent by using a DOR are noted in 6.3.3.

7 Abstract-bind and Abstract-unbind Parameters

7.1 Abstract-bind Parameters

This clause defines and describes the bind-operation parameters for the DFR port.

```
DfrBind ::= ABSTRACT-BIND
  TO {dfr-port[S]}
  BIND
    ARGUMENT      DfrBindArgument
    RESULT        DfrBindResult
    BIND-ERROR    DfrBindError
```

7.1.1 Bind-argument Parameters

The **DfrBindArgument** parameter identifies or authenticates the DFR-User (see also 6.3.8). It also accepts a set of restrictions for entries to be returned as result of a DFR abstract operation.

The definition is:

```
DfrBindArgument ::= SEQUENCE {
  initiator-name      [0] DistinguishedName,
  credentials         [1] Credentials,
  retrieve-restrictions [2] Restrictions OPTIONAL, -- default is none--
  dfr-configuration-request [3] BOOLEAN DEFAULT FALSE,
  bind-security       [4] BindSecurity OPTIONAL,
  priority            [5] Priority DEFAULT medium,
  dor-for-produce-operations [6] BOOLEAN DEFAULT TRUE,
  dor-for-consume-operations [7] BOOLEAN DEFAULT TRUE }
```

a) **initiator-name** (M): This argument contains the distinguished name of the initiator of the association and is supplied by the DFR-User.

b) **credentials** (M): Credentials can be exchanged between the DFR-User and the DFR-Server. Whether the DFR-User is representing a specific end user or not is of no concern to the DFR-Server. The DFR-Server's view of the accessing subject's identity and access privileges is obtained from the credentials passed. Credentials serve to identify a user, authenticate a user, or certify the identity of a user previously externally authenticated. In the latter case, access control privileges associated with the user can also be passed. The full syntax and semantics of credentials when used for authentication purposes is outside the scope of this part of ISO/IEC 10166 (these matters are common to all bind-operations). Use of certified security attributes is described in the outline below. Semantic details are not defined however since this is dependent on the actual security policy formulated and implemented by the organization operating the DFR application.

The authentication of the user may have taken place external to the DFR-Server; the resulting access control attributes will then be passed in the bind-operation to allow the DFR-Server to make subsequent access control decisions. This is the function of the PAC construct. Either the DFR-User or the DFR-Server can abort the bind-operation if the authentication parameters do not justify successful completion of the bind-operation.

The credentials passed in a bind-operation can be modified by the **Privileges** parameter of a specific DFR abstract operation (see 8.1.3.5).

```
Credentials :: = CHOICE {
    simple [0] Creds,      -- used for initial authentication --
    certified [1] PrivilegeAttributeCertificate } -- used when initial authentication has --
                                                    -- already taken place external to --
                                                    -- the DFR-Server --
```

A **Creds** contains a password associated with the DFR-User.

```
Creds :: = OCTET STRING
```

A **PrivilegeAttributeCertificate** contains attributes associated with the DFR-User such as the user's name, job title or security clearance. These can be used in making access control decisions (see 6.3.8).

```
PrivilegeAttributeCertificate :: = EXTERNAL
```

NOTE - The detailed syntax of the **PrivilegeAttributeCertificate** is under study elsewhere. The access rights needed to make a bind may under some security policies be different from those needed to be established over the bind; the possibility of using two PACs, one to establish the rights to make the bind, the other to apply to access to DFR-Objects in the context of the bind is also under study elsewhere.

c) **retrieve-restrictions** (O): This contains the restrictions on objects to be returned as a result of a DFR abstract operation. The restrictions remain until an unbind-operation is issued.

In the absence of this argument, the default is that no restrictions exist.

This argument consists of the following components:

```
Restrictions :: = SET {
    allowed-document-types [0] SET OF OBJECT IDENTIFIER OPTIONAL, --default is no restriction--
    maximum-length [1] INTEGER OPTIONAL } --default is no restriction--
```

1) **allowed-document-types** (C): The document types that the DFR-User is prepared to accept as result of a retrieve abstract operation. Any document with a document type other than the ones specified will not be returned, but will result in an error, unless the abstract operation has explicitly overridden the restriction.

In the absence of this component, the default is that no retrieve restrictions on document-types exist.

2) **maximum-length** (C): The maximum-length that the DFR-User is prepared to provide as the argument or to accept as the result of an abstract operation. Any result prepared by the DFR-Server exceeding the maximum length specified will not be returned, but will result in an error.

In the absence of this component, the default is that no restrictions on result-length exist.

d) **dfr-configuration-request** (C): The dfr-configuration-request is specified to obtain information relating to which optional extension attribute sets and constraints the DFR-Server supports.

In the absence of this component, the default is FALSE which indicates that no such request is being made.

e) **bind-security** (O): This specifies OSI security services required in the bind-operation, for example peer entity authentication of the software entities involved, or confidentiality, or integrity protection.

BindSecurity :: = EXTERNAL

f) **priority** (C): Priority requested for this DFR-User during the association. If there is no other priority specified in the abstract operations (see 8.1.3.4), this priority shall be applied. By default medium priority is applied.

g) **dor-for-produce-operations** (O): This parameter, if left TRUE, indicates that the DFR-Server shall, if requested, provide a DOR in the result of an abstract operation. Whether the DFR-Server commits to this request is reported in the DfrBindResult parameters.

h) **dor-for-consume-operations** (O): This parameter, if left TRUE, indicates that the DFR-Server shall, if requested, accept a DOR in the arguments of an abstract operation. Whether the DFR-Server commits to this request is reported in the DfrBindResult parameters.

7.1.2 Bind-result Parameters

The **DfrBindResult** parameter returns any authentication-attributes needed.

The definition is:

```
DfrBindResult :: = SET {
  authentication-attributes      [0] SET OF AuthenticationAttribute,
  constraints-supported          [1] SET OF ConstraintsType OPTIONAL,
  dfr-document-types-supported  [2] SET OF TypeAndAttribute OPTIONAL,
  function-set-supported        [3] FunctionSetType OPTIONAL,
  maximum-length-supported      [4] INTEGER OPTIONAL,
  dor-for-produce-operations    [5] BOOLEAN DEFAULT TRUE,
  dor-for-consume-operations    [6] BOOLEAN DEFAULT TRUE,
  rOA-protocols-accessee       [7] ROAProtocols OPTIONAL,
  rOA-protocols-accessor       [8] ROAProtocols OPTIONAL }
```

a) **authentication-attributes** (M): Any information returned as confirmation of an authentication check. This information is unconstrained by this part of ISO/IEC 10166.

AuthenticationAttribute :: = EXTERNAL

b) **constraints-supported** (C): Specifies the set of all constraints defined for a DFR-Document-Store. Only present if a dfr-configuration-request is made.

```
ConstraintsType :: = SEQUENCE {
  name-constraint [0] NameConstraint,
```

version-constraint [1] VersionConstraint }

NameConstraint ::= ENUMERATED {

no-name-constraint (0),
local-unambiguity (1),
global-unambiguity (2) }

VersionConstraint ::= ENUMERATED {

no-version-constraint (0),
version-unambiguity (1) }

For a DFR-Document-Store the following constraints can be specified:

- 1) **Name-constraint** specifies the uniqueness of the DFR-Title :
 - i) No constraints for DFR-Titles (no uniqueness of DFR-Titles enforced);
 - ii) Global unambiguity of DFR-Titles (each DFR-Entry has a DFR-Title unique within the DS);
 - iii) Local unambiguity of DFR-Titles (each DFR-Entry has a DFR-Title unique within its parent group).
- 2) **Version-constraint** specifies the uniqueness of the Version-Name attribute:
 - i) No constraints for Version-Names (no uniqueness of Version-Names enforced);
 - ii) Version unambiguity (each Version of a Conceptual-Document is uniquely identified by its Version-Name in the scope of its Conceptual-Document).

c) **dfr-document-types-supported** (C): This is a list of document types supported by this DFR-Server, with a list of extension attributes supported for each document type. Only present if a dfr-configuration-request was made.

TypeAndAttribute ::= SEQUENCE {

document-types [0] OBJECT IDENTIFIER,
attributes [1] SET OF AttributeType }

d) **function-set-supported** (C): Specifies which function-set of the three defined sets (Flat Store Set, Pre-defined Store Structure Set, Full Set - see 8.4) is supported. Only present if a dfr-configuration-request is made.

FunctionSetType ::= ENUMERATED {

flat-store (1),
pre-defined-store (2),
full-set (3) }

e) **maximum-length-supported** (C): Specifies the maximum-length the server supports. This value shall be less than or equal to the maximum length specified by the DFR-User in the retrieve-restrictions of the bind-operation. This parameter is only present if a maximum length was specified in the bind arguments.

f) **dor-for-produce-operations** (O): This parameter, if left TRUE, indicates that the DFR-Server can provide a DOR in the result of an abstract operation.

g) **dor-for-consume-operations** (O): This parameter, if left TRUE, indicates that the DFR-Server accepts a DOR in the arguments of an abstract operation.

h) **rOA-protocols-accessee** (O): The DFR optionally returns information about which ROA-protocol application context(s) it supports, as accessee, for the transfer of objects corresponding to DORs.

ROAProtocols ::= SEQUENCE OF OBJECT IDENTIFIER

i) **rOA-protocols-accessor** (O): The DFR optionally returns information about which ROA-protocol application context(s) it supports, as accessor, for the transfer of objects corresponding to DORs.

7.1.3 Bind-error Parameters

The DFR-port can report either of two errors, **SecurityProblem** or **ServiceProblem**. **SecurityProblem** indicates that the credentials parameter was not adequate to gain access to the DFR-Server. **ServiceProblem** reports that the DFR-Server can not establish the association due to some operational exception. The same errors can occur in DFR abstract operations, see 8.3.

DfrBindError ::= CHOICE {
 service-error [0] ServiceProblem,
 security-error [1] SecurityProblem }

7.2 Abstract-unbind Parameters

An unbind-operation closes the DFR port. The issuing of an unbind-operation results in the deletion of any retrieve-restrictions that were specified in the bind-operation argument. There are no arguments or errors associated with the unbind-operation.

DfrUnbind ::= ABSTRACT-UNBIND
 FROM {dfr-ports [S]}

8 Abstract operations

All abstract operations are invoked by the consumers (DFR-Users) and each abstract operation always reports either success or an error.

8.1 Common Data types used in Abstract operations

8.1.1 Data types used for DFR-Object specification

This subclause identifies, and in some cases defines, a number of data types which are subsequently used in the definition of Document Filing and Retrieval abstract operations. The data types concerned are those which are common to more than one abstract operation, or are likely to be so in the future, or which are sufficiently complex or self-contained as to merit being defined separately from the abstract operation which uses them.

Some of these data types are already defined in clause 6 of this part of ISO/IEC 10166. They are used for specification of different DFR-Objects contained in a DFR-Document-Store. These data types are the following:

- DfrObjectClass** (see 6.3),
- DfrEntry** (see 6.3),
- DfrEntryAttributes** (see 6.3),
- DfrObjectContent** (see 6.3),
- DfrUniquePermanentIdentifier** (see 6.3),
- DfrDocumentContent** (see 6.3.2.1),
- DfrReferenceContent** (see 6.3.3.1),
- DfrObjectClassID** (see 6.3.3.1)
- DfrGroupContent** (see 6.3.4.1),
- DfrSearchResultListContent** (see 6.3.5.1)

8.1.2 Imported data types

Some data types used in clause 6 as well as in this clause are actually defined in other International Standards. These imported data types are:

From the OSI Directory (ISO/IEC 9594-2):

- Attribute,**
- AttributeType,**
- AttributeValue,**
- AttributeValueAssertion,**
- DistinguishedName.**

From the OSI Directory (ISO/IEC 9594-3):

- Filter,**
- FilterItem.**

From the DOAM (ISO/IEC 10031-2):

DOR,
Requested-QoS-level

8.1.3 Data types common for most DFR abstract operations

An argument of any DFR abstract operation is a sequence of "parameters" (or "arguments"), some of them being mandatory for the given abstract operation type, while others are optional. Each individual parameter has its own context-specific tag. Parameters eventually used in (almost) any DFR abstract operation have their tags at the end of the [0, 30] integer range, while more specific parameters have their tags at the beginning of this range. The common parameters are specified as follows:

CommonArguments	:: = SEQUENCE {
task-id	[26] TaskId OPTIONAL,
reservation	[27] Reservation OPTIONAL,
error-handling	[28] ErrorHandlingMode DEFAULT all-or-nothing,
priority	[29] Priority DEFAULT medium,
privileges	[30] Privileges OPTIONAL }

8.1.3.1 Task-id

TaskId :: = OCTET STRING

TaskId is an optional identifier for an abstract operation. A DFR-User can include this parameter in an abstract operation for identification purposes, if it is to be abandoned or continued. The **TaskId** can either be specified by the DFR-User or can be that previously provided by the DFR-Server as continuation-parameter in the result of a **List** or **Search** abstract operation if such an abstract operation was stopped due to a limit-encountered situation (see 8.1.6).

A DFR-User shall not re-use the **TaskId** value in another abstract operation until the respective abstract operation is (perhaps after a continuation) completed, or abandoned.

8.1.3.2 Reservation

The semantics of this parameter when it is specified in an abstract operation are the same as defined in the **Reserve** abstract operation (see 8.2.9). Omission of this parameter means no change in the reservation level of the DFR-Object concerned. For each DFR abstract operation, the DFR-Object to which a reservation applies is specified in the description of that abstract operation. The reservation is defined by reservation-duration, reservation-level and reservation-status.

The **reservation-duration** optionally specifies the time for which the reservation shall exist. If no reservation-duration is specified the DFR-Object will remain reserved indefinitely.

The **reservation-level** specifies which level of reservation shall be applied. Level unreserved is the lowest level, level read-only2 is the highest level. If a change to a lower level from an existing higher level in the case of an uncommitted reservation is requested it is always performed as the last step, in the execution of the abstract operation. If a change to a higher level from a lower level or from uncommitted to committed is requested, it is always performed as the first step in the execution of the abstract operation.

The **reservation-status** defines the quality of the reservation. The reservation-status "committed" is only applicable for the Owner of a DFR-Object. Committed means, the reservation will be maintained until the reservation duration expires. After a committed reservation is specified by the Owner it can not be withdrawn, i.e. the reservation level can only be increased and/or the reservation duration can only be set to a later date or be omitted, in which case the reservation will exist indefinitely. It is also not permitted to set the reservation status from committed to uncommitted.

After a DFR-Object is reserved by some user reservation requests of other users will be rejected. The reserved DFR-Object is only re-available for a reservation by another user if it is set to unreserved by the user who has reserved it in the case of an uncommitted reservation or the reservation duration has expired. The DFR-Reserved-By attribute is used to indicate the identity of the DFR-User who reserved the DFR-Object.

```
Reservation ::= SEQUENCE {
    reservation-duration    [0] GeneralizedTime OPTIONAL,
    reservation-level       [1] ReservationLevel,
    reservation-status      [2] ReservationStatus DEFAULT uncommitted }
```

```
ReservationLevel ::= ENUMERATED {
    unreserved      (0),
    exclusive-write (1),
    exclusive-access (2),
    read-only1      (3),
    read-only2      (4) }
```

```
ReservationStatus ::= ENUMERATED {
    uncommitted (0),
    committed   (1) }
```

The reservation of DFR-Objects is handled according to the following rules:

- a) **unreserved**
The DFR-Users have unrestricted access to the DFR-Object by using the abstract operations defined in this part of ISO/IEC 10166.
- b) **exclusive-write**

- 1) If the reserved DFR-Object is a DFR-Document, a DFR-Reference or a DFR-Search-Result-List and the reservation level is exclusive-write:
 - i) other users shall not **Delete** or **Modify** the DFR-Object (neither any attribute nor the content of the object);
 - ii) other users shall not move the DFR-Object if the **Move** abstract operation is applied directly to the reserved DFR-Object; but the DFR-Object can be implicitly moved, if the **Move** abstract operation is applied to an Ancestor of this DFR-Object;
 - iii) other users shall not use a reserved DFR-Search-Result-List in a **Search** abstract operation.
 - 2) If the reserved DFR-Object is a DFR-Group and the reservation level is exclusive-write:
 - i) other users shall not **Delete** or **Modify** the DFR-Group (neither any attribute nor the content of the group);
 - ii) other users shall not move the DFR-Group if the **Move** abstract operation is applied directly to the reserved DFR-Group; but the DFR-Group can be implicitly moved if the **Move** abstract operation is applied to an Ancestor of this DFR-Group;
 - iii) other users shall not
 - **Delete** or **Move** any Member from the DFR-Group,
 - **Copy** or **Move** any DFR-Object to the DFR-Group,
 - insert new Members into the DFR-Group by **Create** abstract operations;
 - iv) if a Member of the DFR-Group is itself a DFR-Group then the Descendants of this member DFR-Group are not reserved.
- c) **exclusive-access**
- 1) If the reserved DFR-Object is a DFR-Document, a DFR-Reference or a DFR-Search-Result-List and the reservation level is exclusive-access:
 - i) other users shall not **Delete** or **Modify** the DFR-Object (neither any attribute nor the content of the object);
 - ii) other users shall not move the DFR-Object if the **Move** abstract operation is applied directly to the reserved DFR-Object; but the DFR-Object can be implicitly moved, if the **Move** abstract operation is applied to an Ancestor of this DFR-Object;
 - iii) other users shall not **Read** or **Copy** the content of the DFR-Object;
 - iv) other users shall not use a reserved DFR-Search-Result-List in a **Search** abstract operation;
 - v) other users shall not **List** the elements of a DFR-Search-Result-List.
 - 2) If the reserved DFR-Object is a DFR-Group and the reservation level is exclusive-access:
 - i) other users shall not **Delete** or **Modify** the DFR-Group (neither any attribute nor the content of the group);
 - ii) other users shall not move the DFR-Group if the **Move** abstract operation is applied directly to the reserved DFR-Group; but the DFR-Group can be implicitly moved, if the **Move** abstract operation is applied to an Ancestor of this DFR-Group;
 - iii) other users shall not
 - **Delete** or **Move** any Member from the DFR-Group,
 - insert new Members into the DFR-Group by **Create** abstract operations,
 - **Modify** a Member of the DFR-Group (neither any attribute nor the content of a Member) if it is a DFR-Document, a DFR-Reference or a DFR-Search-Result-List,
 - **Modify** the attributes of a Member if it is a DFR-Group
 - **List** the Members of the DFR-Group,
 - **Read** or **Copy** the content of a Member if it is a DFR-Document, a DFR-Reference or a DFR-Search-Result-List,
 - **List** the DFR-Entries noted in the elements of a DFR-Search-Result-List which is a Member of the reserved DFR-Group,

- use a DFR-Search-Result-List which is a Member of the reserved DFR-Group in a **Search** abstract operation;
- iv) if a Member of the DFR-Group is itself a DFR-Group then the Descendants of this member DFR-Group are not reserved.

d) **read-only1**

The restrictions on this reservation level are the same as described for the reservation level exclusive-write. But in this case the restrictions are valid for other users and for the user who has reserved the DFR-Object.

e) **read-only2**

The restrictions on this reservation level are for other users the same as described for the reservation level exclusive-access. But the access is also restricted for the user who has reserved the DFR-Object, i.e. the access to the read-only2 reserved DFR-Object is restricted for the user who has reserved it as described for other users for the reservation level exclusive-write.

8.1.3.3 Error handling mode

```
ErrorHandlingMode ::= CHOICE {
    all-or-nothing           [0] NULL,
    until-first-warning      [1] NULL,
    report-all-warnings     [2] NULL,
    report-n-warnings        [3] INTEGER } -- This parameter is only applicable to multiple --
                                -- abstract operations, --
                                -- these are List or Copy group. --

Warning ::= SEQUENCE { -- warning will never be reported if the DFR-User has no read access right --
    entry    [0] DfrEntryName,
    access1  [1] AccessProblem OPTIONAL, -- see 8.3.3 --
    access2  [2] ReferentAccessProblem OPTIONAL } -- see 8.3.5 --
```

When several DFR-Entries need to be accessed during the execution of the DFR abstract operations **List** and **Copy**, different situations can occur:

- a) none of the DFR-Entries involved can in fact be accessed;
- b) all the involved entries can be accessed;
- c) some entries can be accessed, while others can not.

The first situation (a) is that of a DFR error (of access type), and the second (b) that of success (if no other error occurs). The third situation (c) is more complex. For example, when listing a DFR-Group some members of which are accessible to the user while others are not, the DFR-User may however want all accessible members listed, with a warning(s) about inaccessible Members.

The error handling mode **all-or-nothing** means that the abstract operation will proceed to its conclusion unless any error or warning is detected, in which case the abstract operation will have no effect whatever on the DFR-Document-Store.

The error handling mode **until-first-warning** means that the abstract operation will be performed until an inaccessible Member is detected, in which case the abstract operation is stopped. Only the so far accessed entries are copied or listed and that entry causing the termination of the abstract operation is reported as a warning. The result of the abstract operation is in this case DFR-Server implementation dependent.

The error handling mode **report-all-warnings** means that the abstract operation will be performed to its conclusion regardless of the detection of inaccessible Members. Inaccessible Members, if detected, are reported as warnings in the result of the abstract operation. All accessible Members are copied or listed.

The error handling mode **report-n-warnings** means that the abstract operation will be performed to its conclusion regardless of the detection of inaccessible Members. Inaccessible Members, if detected, are reported as warnings in the result of the abstract operation up to the specified number. All accessible Members are copied or listed independent of the limitation specified for warnings. If the limitation specified for warnings is zero, the abstract operation will be performed to its conclusion without any warnings being reported.

The DFR service definition leaves it to the DFR-User to specify the error-handling mode desired on a per abstract operation basis.

Paired with this common argument is the "warnings" common result. It takes the form of a sequence of access type errors (one per DFR-Entry concerned but not accessible), and can appear in the abstract operation result provided that the abstract operation argument had **ErrorHandlingMode** specified as until-first-warning, report-all-warnings or report-n-warnings.

Any problems related to the modifications parameter (see 8.1.5.5) are always handled as if the all-or-nothing mode was specified.

8.1.3.4 Priority

```
Priority ::= ENUMERATED {
    low      (0),
    medium   (1),
    high     (2)}
```

This common parameter shall be specified for any DFR abstract operation. It is useful in the context of a heavily loaded DFR-Server to allow better service to some privileged requests, according to an applied priority policy. A default **Priority** can already have been established at bind time (see 7.1.1); any **Priority** requested here will override the **Priority** given at bind time.

This concept of operational priority is not necessarily related to any communication priority policy. This means that specification of this parameter may, but need not, influence the quality of service of the underlying layers.

The server does not mandatorily grant the abstract operation with the priority requested. Any numeric characterization of the above three priority levels is server-specific.

8.1.3.5 Privileges

```
Privileges ::= SEQUENCE {
    operation-pac [0] PrivilegeAttributeCertificate OPTIONAL,
    proxy-pac     [1] PrivilegeAttributeCertificate OPTIONAL }
```

The **Privileges** parameter enables a **PrivilegeAttributeCertificate** to be associated directly with an abstract operation request, modifying that available from the bind-operation within which the request is being made. The way the operation-pac modifies the bind PAC depends on the security policy applied. The resulting privileges are only applied for this abstract operation.