INTERNATIONAL STANDARD

ISO/IEC 10031-2

First edition
1991-08-01

Information technology — Text and office systems — Distributed-office-applications model —

Part 2:
Distinguished-object-reference and associated procedures

Technologies de l'information — Bureautique — Modèle d'application pour bureau distribué —

Partie 2: Transfert de données référencées

Reference number
ISO/IEC 10031-2:1991(E)

Contents

Annexes

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for approval before their acceptance as International Standards. They are approved in accordance with procedures requiring at least 75 % approval by the national bodies voting.

International Standard ISO/IEC 10031-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 10031 consists of the following parts, under the general title: *Information technology – Text and office systems – Distributed-office-applications model –*

– *Part 1: General model*

– *Part 2: Distinguished-object-reference and associated procedures.*

Annexes A and B of this part of ISO/IEC 10031 are for information only.

Introduction

This part of ISO/IEC 10031 is one of a set of standards for distributed-office-applications.

Distributed-office-applications are used by an integrated distributed office system, consisting of user nodes and server nodes linked by a network. The user nodes access the server nodes via the network, using access protocols.

In such an environment, data processing applications that within a single host act as a single unit, have been split among the different intelligent components of the system. This splitting has led to the need for standardization of inter-relationships between the different parts of an application.

In this environment the distributed-office-applications should satisfy the following objectives:

a) Make easier the implementation of application-processes developed for a distributed environment based on micro-processors and large or medium sized mainframes interconnected through local area network or wide area network means;

b) Reduce the processing delay time for document related activities such as document filing and retrieval, document distribution, printing, etc., and group communication related activities such as interpersonal messaging, user directory and authentication processes, etc;

c) Allow concurrent processing of different tasks within the distributed office system;

d) Reduce the overall size of an office system and facilitate its modular extension.

Within distributed-office-applications, there will be applications that will act as an Accessee or Accessor of objects whose values are of comparatively large quantities, for example files, documents or body parts.

The transfer of data object values conceptually involves three parties: an Initiator which requests the transfer, an Accessee which produces the data object value and an Accessor which consumes the data object value. To achieve economies in the use of transmission facilities and hence more efficient use of system resources, a mechanism is required whereby an Accessee can provide an Initiator with a reference to a data object value. This reference can then be given by the Initiator to the Accessor,which can directly contact the Accessee to obtain the data object value. This mechanism is known as referenced-object-access (ROA). It provides a mechanism to perform ROA-operations between an open ended list of parties.

CCITT has expressed the intent to publish a recommendation aligned with the content of this part of ISO/IEC 10031.

Information technology – Text and office systems – Distributed-office-applications model – Part 2: Distinguished-object-reference and associated procedures

1 Scope

This part of ISO/IEC 10031 defines the elements used in the specification of the Distinguished-object-reference for use within distributed-office-applications.

Its content covers three major areas:

a) an introductory part in which references, definitions and abbreviations are collected together;
b) a description of Distinguished-object-reference and associated procedures together with a Functional model;
c) distinguished-object-reference structure and Abstract Syntax.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10031. 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 10031 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, *Information processing systems - Open Systems Interconnection - Basic Reference Model.*

ISO 7498-2: 1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model Part 2 : Security Architecture.*

ISO 7498-3: 1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model Part 3 : Naming and addressing.*

ISO 8649 : 1988, *Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element.*

ISO 8650 : 1988, *Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element.*

ISO 8650 : 1988/Cor.1 : 1990, *Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element TECHNICAL CORRIGENDUM 1.*

ISO 8822 : 1988, *Information processing systems - Open Systems Interconnection - Connection oriented presentation service definitions.*

ISO/IEC 8824 : 1990, *Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC 9072-1: 1989,	*Information processing systems - Text communication - Remote Operations - Part 1 : Model, notation and service definition.*
ISO/IEC 9072-2: 1989,	*Information processing systems - Text communication - Remote Operations - Part 2 : Protocol specification.*
ISO/IEC 9594-2: 1990,	*Information processing systems - Open Systems Interconnection - The Directory - Part 2: Models.*
ISO/IEC 9594-3: 1990,	*Information processing systems - Open Systems Interconnection - The Directory -Part 3: Abstract service definition.*
ISO/IEC 10031-1 : 1991,	*Information technology - Text and office systems - Distributed-office-applications model - Part 1: General model.*

3 Definitions

3.1 OSI basic reference model definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 7498:

a) application-entity;
b) open system;
c) presentation-address;
d) protocol;
e) transfer syntax.

3.2 OSI basic reference model Security part definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO 7498-2:

a) security policy.

3.3 OSI basic reference model naming and addressing part definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 7498-3:

a) network address;
b) Presentation selector;
c) Session selector;
d) Transport selector.

3.4 Presentation service definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO 8822:

a) abstract syntax.

3.5 Association Control Service Element (ACSE) definition

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 8649:

a) application context;
b) AE-title.

3.6 Abstract Syntax Notation definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO 8824:

a) ASN.1 ;
b) Generalized Time ;
c) object Identifier.

3.7 Remote Operations definition

This part of ISO/IEC 10031 makes use of the following term defined in ISO/IEC 9072:

a) operation.

3.8 Directory definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 9594:

a) directory entry;
b) distinguished name.

3.9 Distributed-office-applications model part 1 definitions

This part of ISO/IEC 10031 makes use of the following terms defined in ISO/IEC 10031-1:

a) accessee;
b) accessor;
c) consume-operation;
d) data object;
e) data object value;
f) distinguished-object-reference;
g) distributed-office-applications;
h) initiator;
i) produce-operation;
j) referenced-object-access;
k) ROA-operation;
l) ROA-protocol.

3.10 Distributed-office-applications model, part 2 definitions

For the purposes of this part of ISO/IEC 10031, the following definitions apply:

3.10.1 access-time: Time of execution of ROA-operation.

3.10.2 fidelity-time: Time up to which the Accessee performing ROA-operation guarantees the DOR to be valid.

3.10.3 produce-time: Time of execution of the produce-operation.

3.10.4 token: An authorization mechanism whereby the Accessee can validate that the Accessor is authorized to request the access of the referenced data object value.

4 Abbreviations

ASN.1	Abstract Syntax Notation One
AE	Application Entity
DOR	Distinguished Object Reference
QoS	Quality of Service
ROA	Referenced Object Access

5 Conventions

This part of ISO/IEC 10031 uses the descriptive convention listed below:

a) ASN.1 to specify the abstract syntax of information objects.

6 Functional model

Application protocols normally define their own methods for selecting and referring to data objects. Within the distributed office environment there is a need for a more general, application-independent referencing convention to allow different applications to interwork. The "Distinguished Object Reference (DOR)" provides this mechanism.

The DOR has two main functions:

a) It describes how to contact the relevant application entity, using the "AE-identifier" field;
b) It contains a "Local-reference" which gives instructions to the application entity about what subset of its state is to be considered in the response. For example, if the AE is a document store, DORs may indicate a particular document within the store.

The DOR also carries supporting information about the data object to which it refers :

a) The type of that data-object;
b) Some Quality-of-service information;
c) Security and access-control information.

In principle, DORs may be used wherever a reference to a data-object-value is required, although some protocols may choose to limit the usage of the DOR option.

There is also a class of protocols which always refer to data-objects by using DORs. This class is called "Referenced-object-access Protocols", which may be shortened to "ROA-protocols". (See ISO/IEC 10031-1, 5.3.) Protocol elements from some ROA-protocols may also be included within more specialized Application Protocols.

7 Distinguished-object-reference

7.1 Structure

The DOR is a collection of information providing a globally unique reference to a data object value for a period of time. The components of the DOR are described in the following clauses. (See also table 1.).

Table 1 – Components of DOR

Component	Presence
AE-identifier	C*
Locational-identifier	O
Presentation-address	M
AE-title	O
Application-contexts	M
Direct-logical-identifier	O
Indirect-logical-identifier	O
Local-reference	M
Application	O
Specific-reference	M
Data-object-type	M
Quality-of-service	O
Qos-level	O
Produce-time	M
Fidelity-time	O
Usage-of-reference	D
Token	O

* Mandatory in case of produce-operations and consume-operations
C conditional
O optional
M mandatory
D defaultable

7.1.1 AE-identifier

7.1.1.1 Overview

The AE-identifier identifies the AE which is the part of the Accessee performing the ROA-operation. This parameter is classified as conditional, but it is mandatory in produce-operations and consume-operations.

The AE-identifier has three optional components the locational-identifier, the direct-logical-identifier and the indirect-logical-identifier, however at least one component shall be present. If more than one component are present the present identifiers are used in decreasing precedence, locational-identifier

first indirect-logical-identifier last. If one attempt to gain access to the referenced data object value fails the next present identifier is used to gain access.

7.1.1.2 Locational-identifier

Locates the application-entity for ROA-operations by means of its presentation-address. The presentation-address is mandatory.

a) Presentation-address comprises:

1)	Presentation selector	OPTIONAL;
2)	Session selector	OPTIONAL;
3)	Transport selector	OPTIONAL;
4)	Network-address;	

b) AE-title — identifies an Application-entity;

c) Application-contexts — contains one or more names of the application contexts performing referenced-object-access.

7.1.1.3 Direct-logical-identifier

Identifies the application-entity which is part of the Accessee-application-process. This structure can only be used if a Directory server is available to allow resolution of the location of the AE, and to list the available application contexts for the purpose of transfer. (See ISO/IEC 9594.)

7.1.1.4 Indirect-logical-identifier

Identifies a primary Directory entry which refers directly or indirectly to the Directory entry of the application-entity. How this application-entity is referred to is defined by the object class of the identified primary Directory entry.

NOTE – This field is intended to allow more flexibility in the information returned from Directory entries. For example, if several named servers are indirectly defined to reside on a named open system, the presentation address of the open system can easily be changed without altering the directory entry of all the individual servers.

7.1.2 Local-reference

This parameter is mandatory.

The Application component is optional. It is used where an open system has more than one generator of specific references serviced by the AE defined by the AE-identifier. (See 7.1.1.)

The specific-reference is the application-specific reference to the data object value.

The internal structure and semantics of the local-reference components are local to the Accessee.

7.1.3 Data-object-type

The type of the referenced data object is identical in produce-operations and consume- operations. This type shall be a self-identifying type using the "direct reference" mechanism of the ASN.1 external type.

The Object Identifier within the external type is the value of the Data-object-type parameter in the DOR. This Object Identifier identifies the abstract syntax of the data object, in combination with a specific transfer syntax.

7.1.4 Quality-of-service

If this parameter is absent, level 1 and single use of the DOR is assumed.

7.1.4.1 QoS-level

If this parameter is absent, level 1 is assumed. Produce-time is mandatory for level 2 and level 3, fidelity-time is mandatory for level 3.

Produce-time is the time of execution of the produce-operation.

Fidelity-time is the time up to which the Accessee performing the ROA-operation guarantees the DOR to be valid.

Access time is the time of execution of the ROA-operation.

Time is represented in "Generalized Time", defined in ISO 8824.

The referenced data object value as existing at produce-time may be the same at access-time, might be changed in between, or may not be available at all at access-time.

This situation can be modeled by three levels of QoS:

a) Level 1: the referenced data object value might be changed or might not be available at access-time ;

b) Level 2: the referenced data object value might be changed or might not be available at access-time. However a change which has occurred since the produce-time is indicated to the Accessor and it is a decision of the Accessor to process the changed data object value ;

c) Level 3: as level 2 but now the fidelity of the DOR is guaranteed up to a certain point in time (fidelity-time) (i.e. the referenced data object value is identical at produce-time and access-time). The referenced data value is reserved independently of the original data object value, which may be changed by other operations.

The Initiator may request a certain level of QoS in the produce-operation, but the Accessee may respond with a level of QoS equal to or less than the requested.

7.1.4.2 Usage-of-reference

The existence of an DOR requires resources in the Accessee. An Accessee may provide two levels of use for an DOR:

a) single use;

b) multiple use.

In the case of single use the resources required for an DOR are freed after the execution of the ROA-operation. No more ROA-operations using that DOR are possible.

In the case of multiple use of an DOR, the resources required for an DOR are kept after the ROA-operation to allow multiple ROA-operations. The lifetime of the DOR is determined by the Accessee. However in the case of QoS level 3 the lifetime is greater or equal to the fidelity-time.

The user may request multiple use of DOR in the produce-operation and the Accessee responds with a usage-of-reference he determines.

7.1.5 Token

The token is used as an authorization mechanism whereby the Accessee can validate that the Accessor is authorized to request the access of the referenced data object value.

The mechanism by which the Initiator authenticates itself to the Accessee is outside the scope of this part of ISO/IEC 10031. Assuming that this authentication has taken place, and that the Accessee is satisfied that this Initiator has the right to access the specified data object value, the Accessee includes in the DOR a token, which must be used by a Accessor when it accesses the Accessee.

The structure of the simple-token can be as simple or complex as required by the security policy applicable to the Accessee and is conveyed transparently by the DOR. Alternatively an externally defined token may be taken from a future Security Standard.

7.2 Matching Rules

If the configuration of AE-identifier and Local-reference match in two DORs, they should refer to the same object.

7.3 Abstract Syntax

The following is the formal definition of the DOR, using the Abstract Syntax Notation identified in ISO 8824.

```
DOR-definition {joint-iso-ccitt dor(??) reference-definition (0)}
DEFINITIONS::= BEGIN
EXPORTS   AE-identifier, Altered-value, DOR, dor-abstract-syntax, dor-syntax-asn1, dorx,
          Extend-QoS, Local-reference, Locational-identifier, Produce-QoS, QoS-level,
          Quality-of-Service, Requested-QoS-level, Single-use-of-reference, Token;
IMPORTS   DistinguishedName     FROM  InformationFramework
                                      {joint-iso-ccitt ds(5) modules(1)
                                      informationFramework(1)}
          PresentationAddress   FROM  SelectedAttributeTypes
                                      {joint-iso-ccitt ds(5) modules(1)
                                      selectedAttributeTypes(5)};

-- Defined Object Identifiers
dorx OBJECT IDENTIFIER :: = {joint-iso-ccitt dor(11)}
```

```
-- Object identifier for abstract syntax of DOR
dor-abstract-syntax OBJECT IDENTIFIER :: =     {dorx reference-abstract-syntax (1)}

-- Object identifier for abstract syntax of DOR with basic ASN.1 encodings in EXTERNAL
dor-syntax-asn1 OBJECT IDENTIFIER :: =     {dorx reference-syntax(2) asn1(0)}

-- Definition of DOR type
DOR:: = SEQUENCE {
    ae-identifier              [0]  AE-Identifier OPTIONAL,
                                    -- mandatory in case of produce-operations and consume-operations
    local-reference            [1]  Local-reference,
    data-object-type           OBJECT IDENTIFIER,
                                    -- identifying the abstract syntax and  the transfer syntax of the
                                    -- referenced data value
    quality-of-service         [2]  Quality-of-Service DEFAULT{},
    token                      [3]  Token OPTIONAL}

AE-Identifier :: = SEQUENCE { -- at least one component shall be present
    locational-identifier      [0]  Locational-identifier OPTIONAL,
    direct-logical-identifier  [1]  DistinguishedName OPTIONAL,
    indirect-logical-identifier [2]  DistinguishedName OPTIONAL}

Locational-identifier :: = SEQUENCE {
    presentation-address       [0]  PresentationAddress,
    ae-title                   [1]  AE-title -- as defined in ISO 8650:1988/Cor.1:1990 --
                                    OPTIONAL,
    application-contexts       SET OF OBJECT IDENTIFIER}

Local-reference:: = SEQUENCE{
    application                [0]  OCTET STRING OPTIONAL,
    specific-reference         [1]  OCTET STRING}

Quality-of-Service:: = SEQUENCE{
    qoS-level                  [0]  QoS-level DEFAULT level-1 NULL,
    usage-of-reference              Single-use-of-reference DEFAULT TRUE}

QoS-level:: = CHOICE{
    level-1                    [1]  IMPLICIT NULL,
    level-2                    [2]  IMPLICIT GeneralizedTime  -- specifying the produce time -- ,
    level-3                    [3]  IMPLICIT SEQUENCE{
                                    produce-time   GeneralizedTime,
                                    fidelity-time  GeneralizedTime}}

Single-use-of-reference:: = BOOLEAN

Token:: = CHOICE{
    simpletoken                OCTET STRING,     -- used to validate an access which use this DOR
    externaltoken              EXTERNAL          -- for future proxy mechanism
                }
```

```
-- Data types for produce-operations

Produce-QoS      :: = SEQUENCE{
     qoS-level                  [0]  Requested-QoS-level DEFAULT level-1 NULL,
     usage-of-reference              Single-use-of-reference DEFAULT TRUE}

Requested-QoS-level:: = CHOICE{
     level-1         [1]  IMPLICIT NULL,
     level-2         [2]  IMPLICIT NULL,
     level-3         [3]  IMPLICIT GeneralizedTime -- specifying the requested fidelity-time
                     }

-- Data types for extending a specific QoS

Extend-QoS     :: = SEQUENCE{
   qoS-level                    [0]  Requested-QoS-level OPTIONAL,
                                       -- if omitted, no change required
   usage-of-reference                Single-use-of-reference  OPTIONAL
                                       -- if omitted, no change required
                                }

-- Data types for requesting / indicating value alteration in produce-operation or access-operation

Altered-value :: =  ENUMERATED   {   value-not-altered (1),
                                     value-altered (2),
                                     undefined (3) }

END  -- of DOR-definition
```