INTERNATIONAL STANDARD

ISO/IEC 9594-2

First edition
1990-12-15

Information technology — Open Systems Interconnection — The Directory —

Part 2: Models

Technologies de l'information — Interconnexion de systèmes ouverts — L'annuaire —

Partie 2: Modèles

Reference number
ISO/IEC 9594-2 : 1990 (E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. 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 voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9594-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 9594 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — The Directory*:

- *Part 1: Overview of concepts, models and services*
- *Part 2: Models*
- *Part 3: Abstract service definition*
- *Part 4: Procedures for distributed operation*
- *Part 5: Protocol specifications*
- *Part 6: Selected attribute types*
- *Part 7: Selected object classes*
- *Part 8: Authentication framework*

Annexes B and C form an integral part of this part of ISO/IEC 9594. Annexes A, D, E and F are for information only.

Introduction

0.1 This part of ISO/IEC 9594, together with the other parts, has been produced to facilitate the interconnection of information processing systems to provide directory services. A set of such systems, together with the directory information which they hold, can be viewed as an integrated whole, called the *Directory*. The information held by the Directory, collectively known as the Directory Information Base (DIB), is typically used to facilitate communication between, with or about objects such as application entities, people, terminals and distribution lists.

0.2 The Directory plays a significant role in Open Systems Interconnection, whose aim is to allow, with a minimum of technical agreement outside of the interconnection standards themselves, the interconnection of information processing systems:

- from different manufacturers;
- under different managements;
- of different levels of complexity; and
- of different ages.

0.3 This part of ISO/IEC 9594 provides a number of different models for the Directory as a framework for the other parts of ISO/IEC 9594. The models are the overall (functional) model; the organizational model; the security model; and the information model. The latter describes the manner in which the Directory organizes the information it holds. It describes, for example, how information about objects is grouped to form directory entries for those objects and how that information provides names for objects.

0.4 Annex A summarizes the mathematical terminology associated with tree structures.

0.5 Annex B summarizes the usage of ASN.1 object identifiers in ISO/IEC 9594.

0.6 Annex C provides the ASN.1 module which contains all of the definitions associated with the information framework.

0.7 Annex D lists alphabetically the terms defined in this document.

0.8 Annex E describes some criteria that can be considered in designing names.

0.9 Annex F describes guidelines for access control.

0.10 Annex G describes the derivation of a subclass.

Information technology — Open Systems Interconnection — The Directory —

Part 2: Models

SECTION 1: GENERAL

1 Scope

1.1 The models defined in this part of ISO/IEC 9594 provide a conceptual and terminological framework for the other parts of ISO/IEC 9594 which define various aspects of the Directory.

1.2 The functional and organizational models define ways in which the Directory can be distributed, both functionally and administratively.

1.3 The security model defines the framework within which security features, such as access control, are provided in the Directory.

1.4 The information model describes the logical structure of the DIB. From this viewpoint, the fact that the Directory is distributed, rather than centralized, is not visible. The other parts of ISO/IEC 9594 make use of the concepts of the information framework. Specifically:

a) the service provided by the Directory is described (in ISO/IEC 9594-3) in terms of the concepts of the information framework: this allows the service provided to be somewhat independent of the physical distribution of the DIB;

b) the distributed operation of the Directory is specified (in ISO/IEC 9594-4) so as to provide that service, and therefore maintain that logical information structure, given that the DIB is in fact highly distributed.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9594. 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 9594 are encouraged to investigate the possibility of applying the most recent editions of the standards listed 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/IEC 8824:1990, *Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC 9594-1:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 1: Overview of Concepts, Models and Services.*

ISO/IEC 9594-3:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 3: Abstract Service Definition.*

ISO/IEC 9594-4:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 4: Procedures for Distributed Operation.*

ISO/IEC 9594-5:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 5: Protocol Specifications.*

ISO/IEC 9594-6:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 6: Selected Attribute Types.*

ISO/IEC 9594-7:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 7: Selected Object Classes.*

ISO/IEC 9594-8:1990, *Information Technology — Open Systems Interconnection — The Directory — Part 8: Authentication Framework.*

3 Definitions

Definitions of terms are included at the beginning of individual clauses, as appropriate. An index of these terms is provided in annex D for easy reference.

4 Abbreviations

ADDMD	Administration Directory Management Domain
AVA	attribute value assertion
DIB	Directory Information Base
DIT	Directory Information Tree
DMD	Directory Management Domain
DSA	Directory System Agent
DUA	Directory User Agent
PRDMD	Private Directory Management Domain
RDN	Relative Distinguished Name

SECTION 2: DIRECTORY MODEL

5 Directory Model

5.1 Definitions

5.1.1 *Access Point* : the point at which an abstract service is obtained

5.1.2 *Administration Directory Management Domain (ADDMD)* : A DMD which is managed by an Administration.

Note — The term Administration denotes a public telecommunications administration or other organization offering public telecommunications services;

5.1.3 *Administrative Authority* : An entity which has administrative control over all entries stored within a single Directory System Agent.

5.1.4 *the Directory* : A repository of information about objects, and which provides directory services to its users which allow access to the information.

5.1.5 *Directory Management Domain (DMD)* : A collection of one or more DSAs and zero or more DUAs which is managed by a single organization.

5.1.6 *Directory System Agent (DSA)* : An OSI application process which is part of the Directory.

5.1.7 *(Directory) user* : The end user of the Directory, i.e., the entity or person which accesses the Directory.

5.1.8 *Directory User Agent (DUA)* : An OSI application process which represents a user in accessing the Directory.

Note — DUAs may also provide a range of local facilities to assist users compose queries and interpret the responses.

5.1.9 *Private Directory Management Domain (PRDMD)* : A DMD which is managed by an organization other than an Administration*.

5.2 The Directory and its Users

5.2.1 A directory user (e.g., a person or an application-process) obtains directory services by accessing the *Directory*. More precisely, it is a *Directory User Agent (DUA)*, which actually accesses the Directory and interacts with it to obtain the service on behalf of a particular user. The Directory provides one or more *access points* at which such accesses can take place. These concepts are illustrated in figure 1.

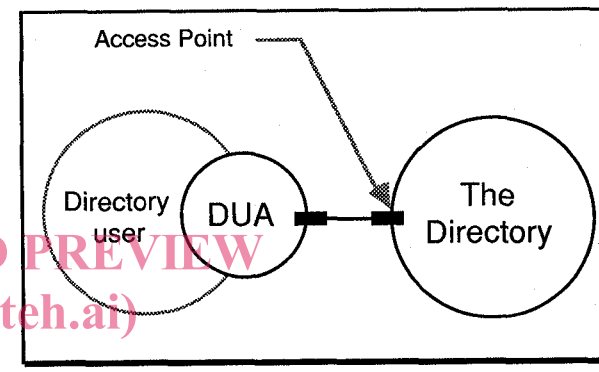

Figure 1 — Access to the Directory

5.2.2 The services provided by the Directory are defined in ISO/IEC 9594-3.

5.2.3 The Directory is a repository of information about objects, and the directory services it provides to its users are concerned with various kinds of access to this information. The information is collectively known as the *Directory Information Base (DIB)*. A model for the DIB is defined in section three of this part of ISO/IEC 9594.

5.2.4 A DUA is manifested as an application-process. Each DUA represents precisely one directory user.

Notes

1. Some open systems may provide a centralized DUA function retrieving information for the actual users (application-processes, persons, etc.). This is transparent to the Directory.

2. The DUA functions and a DSA (see 5.3.1) can be within the same open system, and it is an implementation choice whether to make one or more DUAs visible within the OSI Environment as application-entities.

3. A DUA may exhibit local behavior and structure which is outside the scope of ISO/IEC 9594. For example, a DUA which represents a human directory user may provide a range of local facilities to assist its user to compose queries and interpret the responses.

5.3 Functional Model

5.3.1 The Directory is manifested as a set of one or more application-processes known as *Directory System Agents (DSAs)*, each of which provides zero, one, or more of the access points. This is illustrated in figure 2. Where the Directory is composed of more than one DSA, it is said to be *distributed*. The procedures for the operation of the Directory when it is distributed are specified in ISO/IEC 9594-4.

Note — A DSA will likely exhibit local behavior and structure which is outside the scope of ISO/IEC 9594. For example, a DSA which is responsible for holding some or all of the information in the DIB will normally do so by means of a database, the interface to which is a local matter.

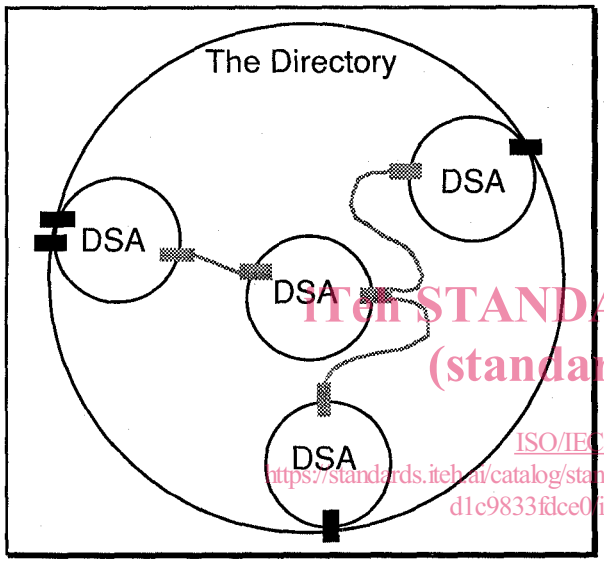

Figure 2 — The Directory Provided by Multiple DSAs

5.3.2 A particular pair of application-processes which need to interact in the provision of directory services (either a DUA and a DSA, or two DSAs) may be located in different open systems. Such an interaction is carried out by means of OSI Directory protocols, as specified in ISO/IEC 9594-5.

5.4 Organizational Model

5.4.1 A set of one or more DSAs and zero or more DUAs managed by a single organization may form a *Directory Management Domain (DMD)*.

Note - The organization which manages a DMD may be an Administration* (i.e., a public telecommunications administration or other organization offering public telecommunications services) in which case the DMD is said to be an Administration DMD (ADDMD); otherwise it is a Private DMD (PRDMD). It should be recognized that the provision of support for private directory systems by CCITT members falls within the framework of national regulations. Thus, the technical possibilities described may or may not be offered by an Administration* which provides directory services. The internal operation and configuration of private DMDs is not within the scope of envisaged CCITT Recommendations.

5.4.2 Management of a DUA by a DMD implies an ongoing responsibility for service to that DUA, e.g., maintenance, or in some cases ownership, by the DMD.

5.4.3 The organization concerned may or may not elect to make use of ISO/IEC 9594 to govern any interactions among DUAs and DSAs which are wholly within the DMD.

5.4.4 Each DSA is administered by an Administrative Authority. This entity has control over all object entries and alias entries stored by that DSA. This includes responsibilities for the Directory schema being used to guide the creation and modification of entries (see clause 9). The structure and allocation of names is the responsibility of a naming authority (see 8.1.6) and the role of the Administrative Authority is to implement these naming structures in the schema.

SECTION 3: INFORMATION MODEL

6 Directory Information Base

6.1 Definitions

6.1.1 *alias entry* : an entry of the class "alias" containing information used to provide an alternative name for an object.

6.1.2 *Directory Information Base (DIB)* : the complete set of information to which the Directory provides access, and which includes all of the pieces of information which can be read or manipulated using the operations of the Directory.

6.1.3 *Directory Information Tree (DIT)*: the DIB considered as a tree, whose vertices (other than the root) are the Directory entries.

Note - The term DIT is used instead of DIB only in contexts where the tree structure of the information is relevant.

6.1.4 *(Directory) entry* : a part of the DIB which contains information about an object.

6.1.5 *immediate superior* (noun) : relative to a particular entry or object (it shall be clear from the context which is intended), the immediately superior entry or object.

6.1.6 *immediately superior*

(*entry*): relative to a particular entry - an entry which is at the initial vertex of an arc in the DIT whose final vertex is that of the particular entry;

(*object*): relative to a particular object - an object whose *object* entry is the immediate superior of *any* of the entries (object or alias) for the second object.

6.1.7 *object* (*of interest*) : anything in some 'world', generally the world of telecommunications and information processing or some part thereof, which is identifiable (can be named), and which it is of interest to hold information on in the DIB.

6.1.8 *object class* : an identified family of objects (or conceivable objects) which share certain characteristics.

6.1.9 *object entry* : an entry which is the primary collection of information in the DIB about an object, and which can therefore be said to represent that object in the DIB.

6.1.10 *subclass*: relative to a superclass -- an object class derived from a superclass. The members of the subclass share all the characteristics of another object class (the superclass) and additional characteristics possessed by none of the members of that object class (the superclass).

6.1.11 *subordinate /inferior*: the converse of superior.

6.1.12 *superclass*: relative to a subclass -- an object class from which a subclass is derived.

6.1.13 *superior* : (applying to entry or object) immediately superior, or superior to one which is immediately superior (recursively).

6.2 Objects

6.2.1 The purpose of the Directory is to hold, and provide access to, information about *objects of interest* (*objects*) which exist in some 'world'. An object can be anything in that world which is identifiable (can be named).

Notes

1. The 'world' is generally that of telecommunications and information processing or some part thereof.

2. The objects known to the Directory may not correspond exactly with the set of 'real' things in the world. For example, a real-world person may be regarded as two different objects, a business person and a residential person, as far as the Directory is concerned. The mapping is not defined in ISO/IEC 9594, but is a matter for the users and providers of the Directory in the context of their applications.

6.2.2 The complete set of information to which the Directory provides access is known as the *Directory Information Base* (DIB). All of the pieces of information which can be read or manipulated by the operations of the Directory are considered to be included in the DIB.

6.2.3 An *object class* is an identified family of objects (or conceivable objects) which share certain characteristics. Every object belongs to at least one class. An object class may be a *subclass* of another object class, in which case the members of the former class (the subclass) are also considered to be members of the latter (the superclass). There may be subclasses of subclasses, etc., to an arbitrary depth.

6.3 Directory Entries

6.3.1 The DIB is composed of *Directory entries* (*entries*) each containing information about (describing) a single object.

6.3.2 For any particular object there is precisely one *object entry*, this being the primary collection of information in the DIB about that object. The object entry is said to represent the object.

6.3.3 For any particular object there may, in addition to the object entry, be one or more *alias entries* for that object, which are used to provide alternative names (see 8.5).

6.3.4 The structure of directory entries is depicted in figure 3 and described in 7.2.

6.3.5 Each entry contains an indication of the object class and the superclasses of that object class with which the entry is associated. In the case of an object entry, this indicates the class(es) to which the object belongs. In the case of an alias entry, this indicates (by means of a special object class, "alias" (defined in 9.4.8.2)) that it is in fact an alias entry, and may also indicate to which subclass(es) of the alias object class the entry belongs.

6.4 The Directory Information Tree (DIT)

6.4.1 In order to satisfy the requirements for the distribution and management of a potentially very large DIB, and to ensure that objects can be unambiguously named (see clause 8), and their entries found, a flat structure of entries is not likely to be feasible. Accordingly, the hierarchical relationship commonly found among objects (e.g., a person works for a department, which belongs to an organization, which is headquartered in a country) can be exploited, by the arrangement of the entries into a tree, known as the *Directory Information Tree* (*DIT*).

Note — An introduction to the concepts and terminology of tree structures can be found in annex A.

6.4.2 The component parts of the DIT have the following interpretations:

a) the vertices are the entries. Object entries may be either leaf or non-leaf vertices, whereas alias entries

are always leaf vertices. The root is not an entry as such, but can, where convenient to do so (e.g., in the definitions of (b) and (c) below), be viewed as a null object entry (see (d) below);

b) the arcs define the relationship between vertices (and hence entries). An arc from vertex A to vertex B means that that the entry at A is the *immediately superior entry* (*immediate superior*) of the entry at B, and conversely, that the entry at B is an *immediately subordinate entry* (*immediate subordinate*) of the entry at A. The *superior entries* (*superiors*) of a particular entry are its immediate superior together with its superiors (recursively). The *subordinate entries* (*subordinates*) of a particular entry are its immediate subordinates together with their subordinates (recursively);

c) the object represented by an entry is, or is closely associated with, the naming authority (see clause 8) for its subordinates;

d) the root represents the highest level of naming authority for the DIB.

6.4.3 A superior/subordinate relationship between objects can be derived from that between entries. An object is an *immediately superior object* (*immediate superior*) of another object if and only if the object entry for the first object is the immediate superior of any of the entries for the second object. The terms *immediately subordinate object*, *immediate subordinate*, *superior* and *subordinate* (applied to objects) have their analogous meanings.

6.4.4 Permitted superior/subordinate relationships among objects are governed by the DIT structure definitions (see 9.2).

7 Directory Entries

7.1 Definitions

7.1.1 *attribute* : the information of a particular type concerning an object and appearing in an entry describing that object in the DIB.

7.1.2 *attribute type* : that component of an attribute which indicates the class of information given by that attribute.

7.1.3 *attribute value* : a particular instance of the class of information indicated by an attribute type.

7.1.4 *attribute value assertion* : a proposition, which may be true, false, or undefined, concerning the values (or perhaps only the distinguished values) of an entry.

Note - In this document the notation "string1 = string2" is used to write down examples of attribute value assertions. In this notation, "string1" is an abbreviation for the 'name' of the attribute type, and "string2" is a textual representation of a suitable value. Although the attribute types in the examples are often based upon real types, such as those defined in ISO/IEC 9594-7 (e.g., "C" stands for "Country", CN for "Common Name"), this is not strictly necessary for the purposes of this document, as the Directory is usually unaware of the meanings of the attribute types in use.

7.1.5 *distinguished value* : an attribute value in an entry which has been designated to appear in the relative distinguished name of the entry.

7.2 Overall Structure

7.2.1 As depicted in figure 3, an entry consists of a set of *attributes*.

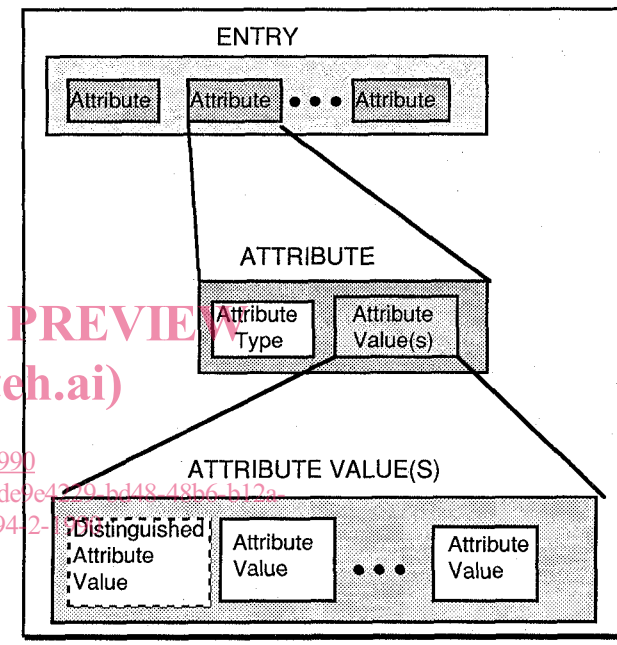

Figure 3 — Structure of an Entry

7.2.2 Each attribute provides a piece of information about, or describes a particular characteristic of, the object to which the entry corresponds.

Note - Examples of attributes which might be present in an entry include naming information such as the object's personal name, and addressing information, such as its telephone number.

7.2.3 An attribute consists of an *attribute type*, which identifies the class of information given by an attribute, and the corresponding *attribute value(s)*, which are the particular instances of that class appearing in the entry

```
Attribute ::=
    SEQUENCE {
        type            AttributeType,
        values      SET OF AttributeValue
                    -- at least one value is required -- }
```

7.3 Attribute Types

7.3.1 Some attribute types will be internationally standardized. Other attribute types will be defined by

national administrative authorities and private organizations. This implies that a number of separate authorities will be responsible for assigning types in a way that ensures that each is distinct from all other assigned types. This is accomplished by identifying each attribute type with an object identifier when the type is defined (as described in 9.5):

AttributeType ::= OBJECT IDENTIFIER

7.3.2 All attributes in an entry shall be of distinct attribute types.

7.3.3 There are a number of attribute types which the Directory knows about and uses for its own purposes. They include:

a) **ObjectClass** — An attribute of this type appears in every entry, and indicates the object class and superclass(es) to which the object belongs.

b) **AliasedObjectName** — An attribute of this type appears in every alias entry, and holds the distinguished name (see 8.5) of the object which this alias entry describes.

These attributes are (partially) defined in 9.5.4.

7.3.4 The types of attributes which shall or which may appear within an entry (other than as mentioned in 7.3.3) are governed by rules applying to the indicated object class(es).

7.4 Attribute Values

7.4.1 Defining an attribute type (see 9.5) also involves specifying the syntax, and hence data type, to which every value in such attributes shall conform. This could be any data type:

AttributeValue ::= ANY

7.4.2 At most one of the values of an attribute may be designated as a *distinguished value*, in which case the attribute value appears in the relative distinguished name (see 8.3) of the entry.

7.4.3 An *attribute value assertion* (*AVA*) is a proposition, which may be true, false, or undefined, concerning the values (or perhaps only the distinguished values) of an entry. It involves an attribute type and an attribute value

AttributeValueAssertion ::=
SEQUENCE {AttributeType, AttributeValue}

and is:

a) undefined, if any of the following holds:

1) the attribute type is unknown,

2) the attribute syntax for the type has no equality matching rule,

3) the value does not conform to the data type of the attribute syntax;

Note — 2) and 3) normally indicate a faulty AVA; 1), however, may occur as a local situation (e.g., a particular DSA has not registered that particular attribute type).

b) true, if the entry contains an attribute of that type, one of whose values matches that value (if the assertion is concerned only with distinguished values, then the matched value shall be the distinguished one);

Note — The matching of values is for equality, and involves the matching rule associated with the attribute syntax.

c) false, otherwise.

8 Names

8.1 Definitions

8.1.1 *alias, alias name*: a name for an object, provided by the use of one or more alias entries in the DIT.

8.1.2 *dereferencing* : replacing the alias name for an object by the object's distinguished name.

8.1.3 *distinguished name* (of an object): one of the names of the object, formed from the sequence of the RDNs of the object entry and each of its superior entries.

8.1.4 (*directory*) *name* : a construct that singles out a particular object from all other objects. A name shall be unambiguous (that is, denote just one object), however it need not be unique (that is, be the only name which unambiguously denotes the object).

8.1.5 *purported name* : a construct which is syntactically a name, but which has not (yet) been shown to be a valid name.

8.1.6 *naming authority* : an authority responsible for the allocation of names. Each object whose object entry is located at a non-leaf vertex in the DIT is, or is closely associated with, a naming-authority.

8.1.7 *relative distinguished name* (*RDN*): a set of attribute value assertions, each of which is true, concerning the distinguished values of a particular entry.

8.2 Names in General

8.2.1 A (*directory*) *name* is a construct that identifies a particular object from among the set of all objects. A name shall be unambiguous, that is, denote just one object. However, a name need not be unique, that is be the only name that unambiguously denotes the object.

8.2.2 Syntactically, each name for an object is an ordered sequence of relative distinguished names (see 8.3).

```
Name ::=
        CHOICE {        -- only one possibility for now --
                        RDNSequence }

RDNSequence ::=         SEQUENCE OF
                                RelativeDistinguishedName

DistinguishedName ::=           RDNSequence
```

Note- Names which are formed in other ways than as described herein are a possible future extension.

8.2.3 The null sequence is the name for the root of the tree.

8.2.4 Each initial subsequence of the name of an object is also the name of an object. The sequence of objects so identified, starting with the root and ending with the object being named, is such that each is the immediate superior of that which follows it in the sequence.

8.2.5 A *purported name* is a construct which is syntactically a name, but which has not (yet) been shown to be a valid name.

8.3 Relative Distinguished Names

8.3.1 Each entry has a unique *relative distinguished name (RDN)*. An RDN consists of a set of attribute value assertions, each of which is true, concerning the distinguished values of the entry.

```
RelativeDistinguishedName ::=
        SET OF AttributeValueAssertion
```

The set contains exactly one assertion about each distinguished value in the entry.

8.3.2 The RDNs of all of the entries with a particular immediate superior are distinct. It is the responsibility of the relevant naming authority for that entry to ensure that this is so by appropriately assigning distinguished attribute values.

Note - Frequently, an entry will contain a single distinguished value (and the RDN will thus comprise a single AVA); however, under certain circumstances (in order to differentiate), additional values (and hence AVAs) may be used.

8.3.3 The RDN for an entry is chosen when the entry is created. A single value instance of any attribute type may form part of the RDN, depending on the nature of the object class denoted. Allocation of RDNs is considered an administrative undertaking that may or may not require some negotiation between involved organizations or administrations. This part of ISO/IEC 9594 does not provide such a negotiation mechanism, and makes no assumption as to how it is performed. The RDN may be modified if necessary by complete replacement.

Note — RDNs are intended to be long-lived so that the users of the Directory can store the distinguished names of objects (e.g., in the Directory itself) without concerns for their obsolescence. Thus RDNs should be changed cautiously.

8.4 Distinguished Names

8.4.1 The *distinguished name* of a given object is defined as being the sequence of the RDNs of the entry which represents the object and those of all of its superior entries (in descending order). Because of the one to one correspondence between objects and object entries, the distinguished name of an object can be considered to also identify the object entry.

Notes

1 It is preferable that the distinguished names of objects which humans have to deal with be user-friendly.

2 ISO 7498-3 defines the concept of a primitive name. A distinguished name can be used as a primitive name for the object it identifies because: (a) it is unambiguous, (b) it is unique, and (c) the semantics of its internal structure (a sequence of RDNs) need not (but of course may) be understood by the user of the Directory.

3 Because only the object entry and its superiors are involved, distinguished names of objects can never involve alias entries.

8.4.2 It proves convenient to define the 'distinguished name' of the root and of an alias entry, although in neither case is the name also the distinguished name of an object. The distinguished name of the root is defined to be the null sequence. The distinguished name of an alias entry is defined to be the sequence of RDNs of the alias entry and those of all of its superior entries (in descending order).

8.4.3 An example which illustrates the concepts of RDN and distinguished name appears in figure 4.

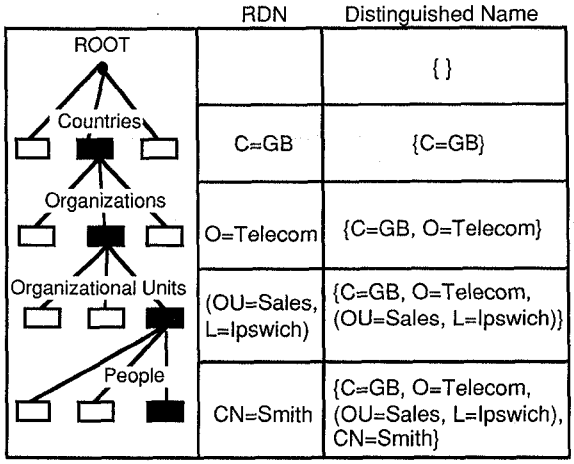

Figure 4 — Determination of Distinguished Names

8.5 Alias Names

8.5.1 An *alias, or an alias name*, for an object is a name at least one of whose RDNs is that of an alias entry. Aliases permit object entries to achieve the effect of having multiple immediate superiors. Therefore, aliases provide a basis for alternative names.