ATIS-0300245.2013

DIRECTORY SERVICE FOR TELECOMMUNICATIONS MANAGEMENT NETWORK (TMN) AND SYNCHRONOUS OPTICAL NETWORK (SONET)

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS

As a leading technology and solutions development organization, ATIS brings together the top global ICT companies to advance the industry's most-pressing business priorities. Through ATIS committees and forums, nearly 200 companies address cloud services, device solutions, emergency services, M2M communications, cyber security, ehealth, network evolution, quality of service, billing support, operations, and more. These priorities follow a fast-track development lifecycle — from design and innovation through solutions that include standards, specifications, requirements, business use cases, software toolkits, and interoperability testing.

ATIS is accredited by the American National Standards Institute (ANSI). ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a founding Partner of oneM2M, a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications sectors, and a member of the Inter-American Telecommunication Commission (CITEL). For more information, visit < www.atis.org >.

AMERICAN NATIONAL STANDARD

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OFMERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES THAT ANY AND ALL USE OF OR RELIANCE UPON THE INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith. Please refer to [http://www.atis.org/legal/patentinfo.asp] to determine if any statement has been filed by a patent holder indicating a willingness to grant a license either without compensation or on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain a license.

ATIS-0300245.2013, *Directory Service for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET)*

Is an American National Standard developed by the under the **ATIS Telecom Management and Operations Committee (TMOC)**.

Published by

Alliance for Telecommunications Industry Solutions
1200 G Street, NW, Suite 500
Washington, DC 20005

ATIS-0300245.2013

[Revision of ATIS-0300245.1997(R2008)]

American National Standard for Telecommunications

Directory Service for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET)

Alliance for Telecommunications Industry Solutions

Approved December 30, 2013

American National Standards Institute, Inc.

Abstract

This standard specifies the usage of the X.500 Directory, protocols, and services for communications between Directory Users and Directory Servers. These specifications are for use of the Directory in support of management communications within the Telecommunications Management Network (TMN), and for specific technologies, such as Synchronous Optical Network (SONET).

Foreword

The information contained in this Foreword is not part of this American National Standard (ANS) and has not been processed in accordance with ANSI's requirements for an ANS. As such, this Foreword may contain material that has not been subjected to public review or a consensus process. In addition, it does not contain requirements necessary for conformance to the Standard.

As a leading technology and solutions development organization, the Alliance for Telecommunications Industry Solutions (ATIS) brings together the top global information and communications technology (ICT) companies to advance the industry's most-pressing business priorities. ATIS serves the public through improved understanding between carriers, customers, and manufacturers. The Telecom Management and Operations Committee (TMOC) develops operations, administration, maintenance and provisioning standards, and other documentation related to Operations Support System (OSS) and Network Element (NE) functions and interfaces for communications networks - with an emphasis on standards development related to U.S.A. communication networks in coordination with the development of international standards.

ANSI guidelines specify two categories of requirements: mandatory and recommendation. The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, TMOC, 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time of consensus on this document, TMOC, which was responsible for its development, had the following leadership:

T. Barrett, TMOC Chair (AT&T)
S. Kiewel, TMOC Vice Chair (Ericsson)
T. Barrett, Technical Editor (AT&T)
C. Underkoffler, ATIS Chief Editor

Table of Contents

Table of Figures

Table of Tables

ATIS Standard on –

Directory Service for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET)

1 Scope

Directory Service provides a means for locating essential information about network resources and their attributes. A directory service can avoid the cumbersome task of locating resources and services needed to handle the growth of interconnected networks, which are increasing in size, complexity, and diverse requirements. The ITU-T Recommendation X.500 Directory provides a general information sharing service utilizing a set of communication protocol facilities and a global name management infrastructure.

This standard specifies the usage of the ITU-T Recommendation X.500 Directory Service within the *Telecommunications Management Network* (TMN). The ITU-T Recommendation X.500 Directory operates in an *Open Systems Interconnection* (OSI) environment and uses the *Association Control Service Element* (ACSE) and *Remote Operations Service Element* (ROSE) of the OSI application layer. Additionally, implementations supporting replication use the Reliable Transfer Service Element (RTSE) of the OSI application layer.

This standard identifies the Directory Services needs of the TMN and addresses the Directory Service support for name to address mapping, specifically for an *Application Entity* (AE) title to facilitate association set up. This standard also examines some requirements for the *Synchronous Optical Network* (SONET) as a specific technology example in particular, including SONET transmission network elements (e.g., broadband cross-connects, wideband cross-connects, lightwave systems, add-drop multiplexers, repeaters, digital loop carriers, etc.) and SONET management systems (e.g., operation systems, mediation devices, element managers, etc.), that connect to the SONET *Data Communications Network* (DCN). Additional Directory Services needs for SONET and other specific technologies may be identified in the future. This standard serves as a basis for future extension of the usage of ITU-T Recommendation X.500 Directory Service for sharing information needed in the TMN for management of networks. This standard specifies Directory replication and use of *Back-Up Directory Server* (B-DS) as alternate methods for enhancing the reliability of the Directory Service.

This standard for Directory Service for TMN does not rely on any other profile, as no suitable *International Standardized Profile* (ISP) yet exists for the ITU-T X.500:2012 series of Recommendations upon which this standard is based. The model for the Directory Service for TMN is the model in *International Telecommunication Union – Telecommunication Standardization Sector* (ITU-T) Recommendation X.501, with the addition of a *Registration Manager* (RM) and *Registration Agent* (RA) to handle automatic registration of *Network Element* (NE) entries into the Directory under certain conditions.

The protocols for the TMN Directory Service are the *Directory Access Protocol* (DAP), *Directory System Protocol* (DSP), *Directory Information Shadowing Protocol* (DISP), and *Directory Operational Binding Management Protocol* (DOP) protocols specified in the X.500 series of ITU-T Recommendations. These protocols are to be used in conformance to the ITU-T X.500 Recommendations. Only the DAP is fully

addressed in this standard, and its profile is provided in Annex B.[1] The profiles for DSP, DISP, and DOP are planned to be added to this TMN Directory standard as needed in future expansions. A *Registration Request Protocol* (RRP) has been added for automatic registration. The application and profiling of DISP and DOP are for further study. This standard also does not fully address security.

The Directory object classes and attributes profiled for use in TMN are a subset of those defined in ITU-T Recommendations X.521 and X.520, plus a small number of TMN-specific objects and attributes defined here for the purposes of representing Network Elements and Operations Systems. This standard also identifies a set of structural object classes, their naming rules, and their naming attributes specified in the X.500 Recommendations. This is the minimum that shall be supported within the *Directory Information Tree* (DIT) commonly used within the TMN.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this American National Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ATIS-0300208.2013, *Operations, Administration, Maintenance, and Provisioning (OAM&P) – Upper-Layer Protocols for Telecommunications Management Network (TMN) Interfaces, Q and X Interfaces*.[2]

ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network*.[3]

ITU-T Recommendation M.3100 (2005), *Generic network information model*.[3]

ITU-T Recommendation X.200 (1994), *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. (same as ISO 7498-1: 1994, Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model)*.[3]

ITU-T Recommendation X.247 (1996) | ISO/IEC 8650-2: 1997, *Information technology – Open Systems Interconnection – Protocol specification for the association control service element (ACSE): Protocol Implementation Conformance Statement (PICS) proforma*.[3]

ITU-T Recommendation X.249 (1995) | ISO/IEC 9072-3: 1996, *Information Technology – Open Systems Interconnection – Remote Operations: Protocol Implementation Conformance Statement (PICS) Proforma.*[3]

ITU-T Recommendation X.500 (2008) | ISO/IEC 9594-1: 2008, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.[3]

ITU-T Recommendation X.501 (2008) | ISO/IEC 9594-2: 2008, *Information technology – Open Systems Interconnection – The Directory: Models*.[3]

ITU-T Recommendation X.509 (2008) | ISO/IEC 9594-8: 2008, *Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks*.[3]

ITU-T Recommendation X.511 (2008) | ISO/IEC 9594-3: 2008, *Information Technology – Open Systems Interconnection – The Directory: Abstract service definition*.[3]

———

[1] The profile is based on a draft *Protocol Implementation Conformance Statement* (PICS) for DAP currently being worked on by *OSI Implementor's Workshop* (OIW). When the draft PICS become standardized, this TMN Directory standard is planned to align with the PICS.

[2] This document is available from the Alliance for Telecommunications Industry Solutions (ATIS) at < https://www.atis.org/docstore/product.aspx?id=24609 >.

[3] This document is available from the International Telecommunications Union. < http://www.itu.int/ITU-T/ >

ITU-T Recommendation X.518 (2008) | ISO/IEC 9594-4: 2008, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation*.[3]

ITU-T Recommendation X.519 (2008) | ISO/IEC 9594-5: 2008, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications*.[3]

ITU-T Recommendation X.520 (2008) | ISO/IEC 9594-6: 2008, *Information Technology – Open Systems Interconnection – The Directory: Selected attribute types*.[3]

ITU-T Recommendation X.521 (2008) | ISO/IEC 9594-7: 2008, *Information technology – Open Systems Interconnection – The Directory: Selected object classes*.[3]

ITU-T Recommendation X.525 (2008) | ISO/IEC 9594-9: 2008, *Information technology – Open Systems Interconnection – The Directory: Replication*.[3]

ITU-T Recommendation X.680 (2008) | ISO/IEC 8824-1: 2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.[3]

ITU-T Recommendation X.681 (2008) | ISO/IEC 8824-2: 2008, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.[3]

ITU-T Recommendation X.682 (2008) | ISO/IEC 8824-3: 2008, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.[3]

ITU-T Recommendation X.683 (2008) | ISO/IEC 8824-4: 2008, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.* [3]

ITU-T Recommendation X.701 (1997) | ISO/IEC 10040: 1998, *Information technology – Open Systems Interconnection – Systems management overview*.[3]

ITU-T Recommendation X.721 *Corrigendum 1 (02/94) to Recommendation – Information Technology – Open Systems Interconnection – Structure of Management Information: Definition of Management Information*.[3]

3 Definitions

For the purposes of this standard, the following definitions apply.

3.1 Imported Definitions

This standard makes use of the following terms defined in ITU-T Recommendation X.200 | ISO 7498:

a) Application Entity;

b) Application Process (AP); and

c) Network Service Access Point (NSAP).

This standard makes use of the following term defined in ITU-T Recommendation X.500 | ISO 9594-1:

a) Directory.

This standard makes use of the following terms defined in ITU-T Recommendation X.501 | ISO 9594-2:

a) Alias;

b) Attribute;

c) Attribute type;
d) Attribute value;
e) Directory Information Base (DIB);
f) Directory Information Tree (DIT);
g) (Directory) name;
h) Directory schema;
i) Directory System Agent (DSA);
j) Directory User Agent (DUA);
k) Distinguished Name (DN);
l) DIT structure Rule;
m) Entry;
n) Name;
o) Object class;
p) Relative Distinguished Name (RDN);
q) Structural Object Class;
r) Subordinate; and
s) Superior.

This standard makes use of the following terms defined in ITU-T Recommendation X.518 | ISO 9594-4:

a) Chaining; and
b) Referral.

This standard makes use of the following terms defined in ITU-T Recommendation X.525 | ISO 9594-9:

a) Caching;
b) Master DSA;
c) Replicated area;
d) Replication; and
e) Shadowing.

This standard makes use of the following terms defined in ITU-T Recommendation M.3010:

a) Network Element (NE); and
b) Operations System (OS).

This standard makes use of the following term defined in ITU-T Recommendation X.701 | ISO/IEC 10040:

a) Management Information Base (MIB).

3.2 Additional Definitions

3.2.1 Directory Server (DS): The DS consists of a DSA and possibly other application processes, such as the Registration Manager defined in this standard, to provide the Directory Service with additional features such as automatic registration of a network element in a DIT.

3.2.2 local cache: The (optional) local cache is a copy of a subset of the DIB. Each Network Element may have a local cache containing information on names and addresses of frequently used NEs. This helps reduce the query traffic from the NE with a DUA to the DSA.

3.2.3 MIB/DIB synchronization: MIB/DIB synchronization in Directory Services ensures that two semantically identical attributes in the Management Information Base and the Directory Information Base have consistent values. The MIB/DIB synchronization algorithm ensures that updates to MIB object attributes result in updates to the paired DIB attribute. MIB/DIB synchronization is specified where applicable in the object class definitions of clause 9.

3.2.4 proxy agent: An agent that provides an application (e.g., management communication) on behalf of another system(s). Usage of the Directory Service by the proxy agent conforms to standard DSA access (i.e., via DAP by a DUA).

3.2.5 Registration Agent (RA): An RA is an application process that receives an RRP message from the Registration Manager signaling it to start automatic registration. The result is that the R-NE's DUA adds the NE system administrative information (e.g., name, make, location, etc.) into the DIB via the "addEntry" operation of the DAP. The RA extracts the DS address from the RRP exchange and provides this address to the DUA.

3.2.6 Registration Manager (RM): An RM is an application process responsible for contacting the RA using the RRP to start automatic registration of a newly discovered NE into the DIB. The RM provides the RA with the appropriate DSA address.

3.2.7 Registration Request Protocol (RRP): The RRP is the protocol used between the RM and RA for initiating automatic registration of an NE system administrative information in the DIB.

3.2.8 Remote-Network Element (R-NE): The R-NE is a type of NE that contains a DUA, an optional local cache, and possibly other application processes such as the Registration Agent. The R-NE communicates with the DSA via the DAP.

4 Abbreviations

ACSE	Association Control Service Element
AE	Application Entity
ANSI	American National Standards Institute
AP	Application Process
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
AVA	Attribute Value Assertion
B-DS	Back-up Directory Server
B-RM	Back-up Registration Manager
BLSR	Bidirectional Line Switched Ring
CCITT	The International Telegraph and Telephone Consultative Committee
DAP	Directory Access Protocol
DCN	Data Communications Network

DIB	Directory Information Base
DIS	Draft International Standard
DISP	Directory Information Shadowing Protocol
DIT	Directory Information Tree
DN	Distinguished Name
DOP	Directory Operational Binding Management Protocol
DS	Directory Server
DSA	Directory System Agent
DSP	Directory System Protocol
DUA	Directory User Agent
EM	Element Manager
ES	End System
IEC	International Electrotechnical Commission
IS	Intermediate System
ISO	International Organization for Standardization
ISP	International Standardized Profile
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IUT	Implementation Under Test
LC	Local Cache
LCN	Local Communications Network
MD	Mediation Device
MHS	Message Handling System
MIB	Management Information Base
MF	Mediation Function
NE	Network Element
NEF	Network Element Function
NSAP	Network Service Access Point
OAM&P	Operations, Administration, Maintenance and Provisioning
OIW	OSI Implementor’s Workshop
OS	Operations System
OSF	Operations System Function
OSI	Open Systems Interconnection
P-DS	Primary Directory Server
P-RM	Primary Registration Manager
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
QA	Q-Adapter
QAF	Q-Adapter Function
RA	Registration Agent

RDN	Relative Distinguished Name
RM	Registration Manager
R-NE	Remote Network Element
ROSE	Remote Operations Service Element
RRP	Registration Request Protocol
RTSE	Reliable Transfer Service Element
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Network
TMN	Telecommunications Management Network

5 Conformance

This clause specifies the requirements for conformance to this Directory Service for TMN and SONET standard. Implementations claiming conformance to this Directory Service for TMN and SONET standard shall satisfy the requirements specified in this clause. The PICS proforma for this standard are provided in Annexes A to F in this standard as a mechanism for stating conformance.

5.1 Conformance by DUAs

A DUA implementation claiming conformance to this Directory for TMN and SONET standard shall satisfy the requirements specified in 13.1 of ITU-T Recommendation X.519 on statement requirements, static requirements, and dynamic requirements, with the following additional requirements specific to TMN and SONET application.

5.1.1 Additional Statement Requirements

The following shall be stated:

a) The selected structure rules and name forms specified in clauses 8 and G.4 for which conformance is claimed;
b) The selected object classes specified in clause 9 for which conformance is claimed;
c) The selected attribute types specified in clause 10 for which conformance is claimed; and
d) The selected abstract syntaxes and matching rules specified in clause 11 for which conformance is claimed.

5.2 Conformance by DSAs

A DSA implementation claiming conformance to this Directory Service for TMN and SONET standard shall satisfy the requirements specified in 13.2 of ITU-T Recommendation X.519 on statement requirements, static requirements, and dynamic requirements that are relevant to the DSA functionality specified in clauses 7 through 14 of this standard, with the following additional requirements specific to TMN and SONET application.

5.2.1 Additional Statement Requirements

The following shall be stated:

a) The selected structure rules and name forms specified in clauses 8 and G.4 for which conformance is claimed;
b) The selected object classes specified in clause 9 for which conformance is claimed;
c) The selected attribute types specified in clause 10 for which conformance is claimed; and
d) The selected abstract syntaxes and matching rules specified in clause 11 for which conformance is claimed.

5.2.2 Additional Static Requirements

A DSA shall:

a) Have the capability of at least supporting the directoryAccessAC application-context as defined by its abstract syntax in clause 8 of ITU-T Recommendation X.519;
b) Have the capability of supporting the structure rules and name forms specified in G.4 for which conformance is claimed;
c) Have the capability of supporting the object classes specified in clause 9 for which conformance is claimed;
d) Have the capability of supporting the attribute types specified in clause 10 for which conformance is claimed; and
e) Have the capability of supporting the abstract syntaxes and matching rules specified in clause 11 for which conformance is claimed.

5.3 Conformance by RMs

The optional Directory population feature is described in clause 13. A SONET environment that claims conformance to this standard shall support the Directory population feature. At least one RM implementation is needed per OSI Level 1 routing area to support the Directory population feature. An RM implementation claiming conformance to this Directory Service for TMN and SONET standard shall satisfy the requirements specified below.

a) *Statement Requirements:* The following shall be stated: the operations of the registrationApplicationContext application-context that the RM is capable of invoking for which conformance is claimed.
b) *Static Requirements:* An RM shall have the capability of supporting the registrationApplicationContext application-context for which conformance is claimed as defined by their abstract syntax in G.5.
c) *Dynamic Requirements:* An RM shall conform to the mapping onto ACSE and ROSE services as defined in clause 13.

5.4 Conformance by RAs

The optional Directory population feature is described in clause 13. A SONET NE that claims conformance to this standard shall support the Directory population feature by supporting the RA functionality. An RA implementation claiming conformance to this Directory for TMN and SONET standard shall satisfy the requirements specified below:

a) *Statement Requirements:* The following shall be stated: the operations of the registrationApplicationContext application-context for which conformance is claimed.
b) *Static Requirements:* An RA shall support the registrationApplicationContext application-context for which conformance is claimed as defined by their abstract syntax in clauses 13 and G.5.

c) *Dynamic Requirements:* An RA shall conform to the mapping onto ACSE and ROSE services as defined in clause 13.

5.5 Conformance by a shadow supplier DSA

A DSA implementation claiming conformance to this Directory Service in the role of shadow supplier shall satisfy the requirements specified in 13.3 of ITU-T Recommendation X.519 on statement requirements, static requirements, and dynamic requirements that are relevant to the DSA functionality specified in clauses 7 through 14 of this standard.

5.6 Conformance by a Shadow Consumer DSA

A DSA implementation claiming conformance to this Directory Service in the role of shadow consumer shall satisfy the requirements specified in 13.4 of ITU-T Recommendation X.519 on statement requirements, static requirements, and dynamic requirements that are relevant to the DSA functionality specified in clauses 7 through 14 of this standard.

6 Requirements

Open systems that are participating in the TMN for the purposes of management require specific knowledge in order to determine the peer systems with which to associate, to enable association, and to fulfill their management functions. In addition, human users of the TMN need certain information to perform their functions. Some of this knowledge may be available from a Directory Service.

The following subclauses describe specific requirements that can be satisfied by the Directory Service defined in this standard.

6.1 Naming/Addressing Support

This Directory Service supports the determination of:

a) The TMN elements involved in the management subsystem;
b) The identity of a Network Element or an Operations System based on its Network Address;
c) The identity of an NE or an OS based on technology-, implementation-, network-, or operation-dependent naming (such as M.3100:userLabel) where available;
d) The identity of NEs and OSs based on vendor or locality information, where known;
e) The identity of an NE or an OS from its MIB naming attribute (M.3100:managedElementId);
f) The value of the M.3100:managedElementId of an NE or an OS for use in management protocol exchanges;
g) The identity of an NE or an OS based on its function and role;
h) The identity of a *Bidirectional Line Switched Ring* (BLSR) NE based on its ring node ID information; and
i) The identity of an OS based on its additionally supported functional blocks, as defined in ITU-T Recommendation M.3010.

6.2 Association Resolution

This Directory Service supports the determination of:

a) The Application Entity titles of the entities with which management associations may be established;
b) The presentation addresses of those entities; and
c) Given a managed object and optionally the name of a desired management capability, the identity of one or more management agents capable of providing that management function.[4]

6.3 Shared Management Knowledge

Managed objects may represent both static and dynamic resources. This Directory Service shall not be used to store dynamic state information which is obtainable from the MIB. Furthermore, the information in the DIB is updated only through abstract services.

This Directory Service supports the determination of: [5]

a) The supported application contexts of a management application entity;
b) The supported functional units of a management application entity;
c) The supported management profiles of a management application entity;
d) The list of managed objects or managed object classes in a managed application entity; and
e) The grouping of managed systems and management systems into management domains.

NOTE – The above requirements are currently not covered in this standard.[6]

6.4 Security Support

This Directory Service supports the determination of:

a) Passwords for management entities to support simple authentication during management association establishment;
b) Certified public keys for management entities to support strong authentication during management association establishment; and
c) Access control information to support access to the Directory.

NOTE – The above requirements are currently not covered in this standard.[6]

6.5 Administrative Usage

This Directory Service allows:

a) A human administrator to search, modify, add, and delete entries and sub-tree structures (within access rights) for the purposes of management of this Directory; and
b) An application to search, modify, add, and delete entries and sub-tree structures (within access rights) for the purposes of administering the Directory.

6.6 Messaging and use by persons

In order to support interpersonal messaging, this Directory Service supports the determination of:

———————

[4] This requirement is currently not fully satisfied in this standard.
[5] The ISO 10164-1 series of standards | ITU Recommendation X.730 contains information on this topic.
[6] These are expected to be covered in future expansions of the standard.

a) The X.400 addresses of roles and persons involved in TMN management; and
b) Other contact information (telephone numbers, facsimile, postal, etc.) of organizations, roles, and persons involved in TMN management.

NOTE – The above requirements are currently not covered in this standard.

7 Architecture Overview

The basic architecture of the Directory Service for the TMN is as described in ITU-T Recommendation X.501. An RM and RA have been added, with the RRP used between them, to effect automatic registration of some entries in the Directory. The Directory Service Architecture is shown in Figure 1, and clause I.2 shows examples of how Directory Servers may be deployed in a SONET subnetwork. Note that the components of the Directory Server (the DSA, DIB fragment, and RM) may actually form a distributed set of applications. Similarly for the NE, the RA and DUA need not be resident on the same physical system.

Access to information in a segment of the global Directory Information Base is provided by the Directory System Agent resident on a Directory Server. Each user, including people and applications, is represented in accessing the Directory by a Directory User Agent. The DUA and DSA are OSI application processes that reside at Layer 7 of the OSI Stack defined in ATIS-0300208. The Directory Access Protocol is used by a DUA to access a DSA.

In general, a DUA resides on a network element other than the Directory Server, although administrators and applications on the DS may also require a DUA for local access. On a remote network element, the DUA may maintain a Local Cache of Directory information; this is a local matter.

Automatic registration of certain entries into the DIB can be provided by a Registration Manager, which uses the Registration Request Protocol to communicate with a Registration Agent. The RRP provides an application-layer message requesting a user agent of the Directory Service to commence registration. The registration is accomplished through the DUA using the DAP to communicate the appropriate information to the DSA. The RM and RA are OSI application processes that reside at Layer 7 of the OSI Stack defined in ATIS-0300208.

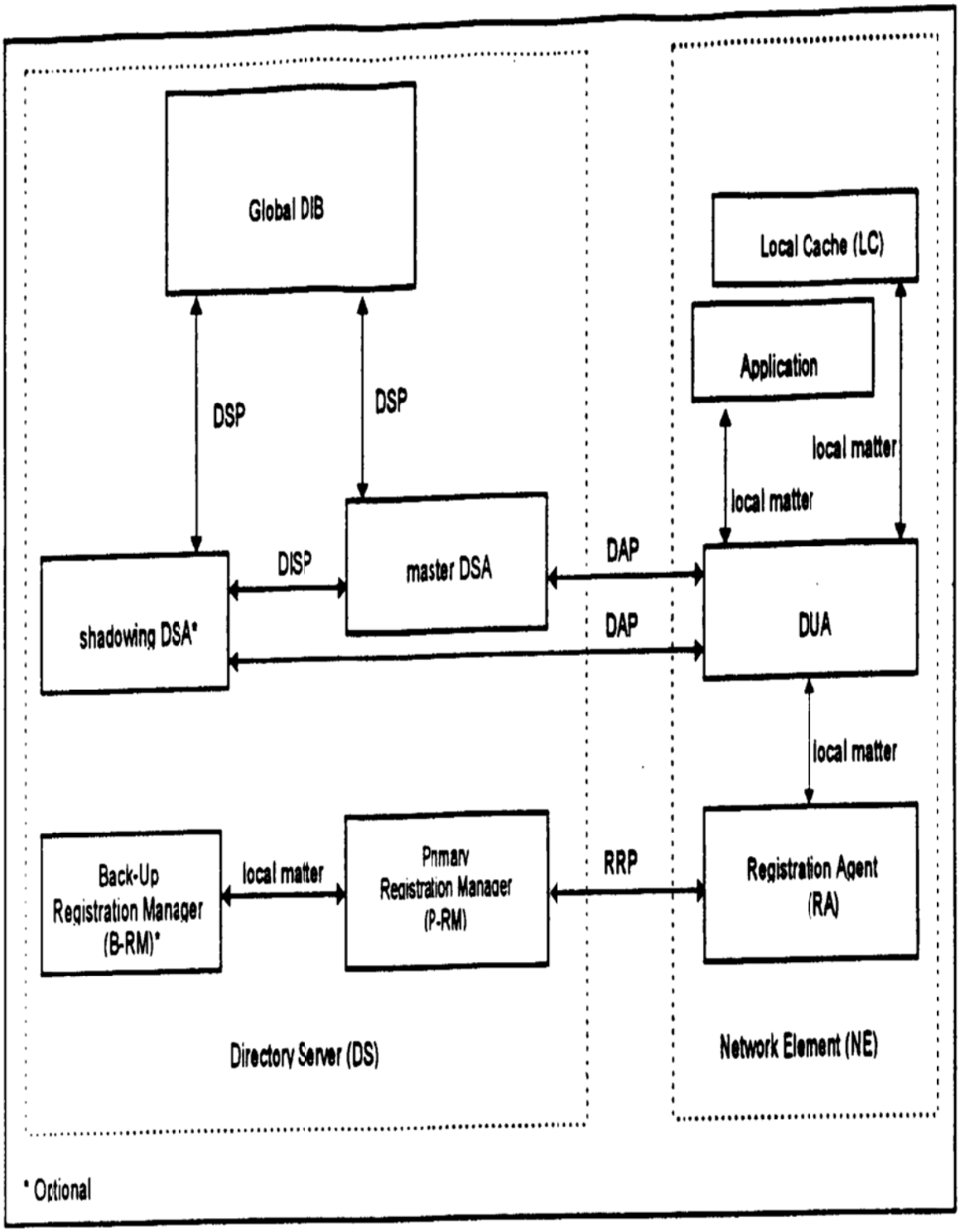


Figure 1 - Directory service architecture

In general, the RM resides on the DS and the RA resides at the NE. Communications between the RA and DUA are not defined by this standard. The manner in which the RM is invoked to communicate with the RA is not specified in this standard, although the situations under which this would occur are discussed.

The Directory Service shall be implemented to provide a high level of availability. Recovery time in the order of minutes is required for this standard. The specific implementation is beyond the scope of this standard. To eliminate susceptibility of the Directory to a single point of failure, it may be desirable to use shadowing and a back-up RM or to have a back-up DS available. Possible solutions include: [7]

a) The use of shadowing to provide an alternate DSA with the same information;

b) The use of the RM to produce a back-up DS for at least a portion of the DIB entries in the event of failure (see H.1);

c) The use of a back-up Registration Manager;

d) The use of fault-tolerant approaches capable of dealing with domain segmentation; and

e) The use of fault-tolerant platforms.

A DSA may use information in its DIB fragment to satisfy an access request, or may interact with other DSAs to obtain information in other parts of the DIB. This function is referred to as chaining. The DSA may require references to superior and subordinate segments of the DIB. The procedures for distributed operation are covered in ITU-T Recommendation X.518.

A DSA shall be able to provide a chained mode-only form of operation – i.e., no referral capability – if the DUA expresses the service control of *"preferChaining"*. This can provide the capability of minimizing the complexity and cost of providing DUA capability in smaller Network Elements.

[7] In this standard, 1988 versions of ITU-T Recommendation X.500 are not precluded and these do not support replication/shadowing. However, replication/shadowing is expected to be the preferred method in future expansions of the standard.

8 DIT Structure

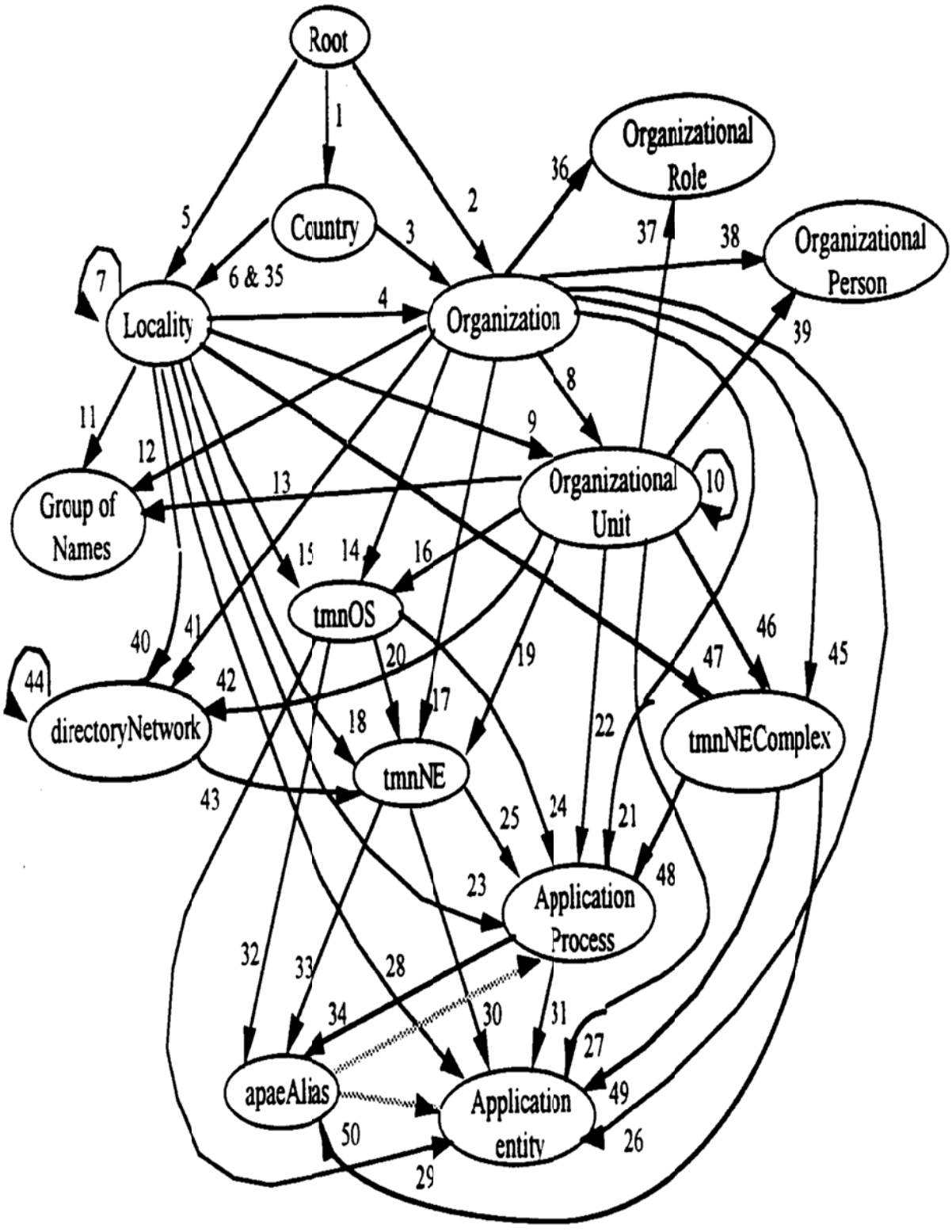
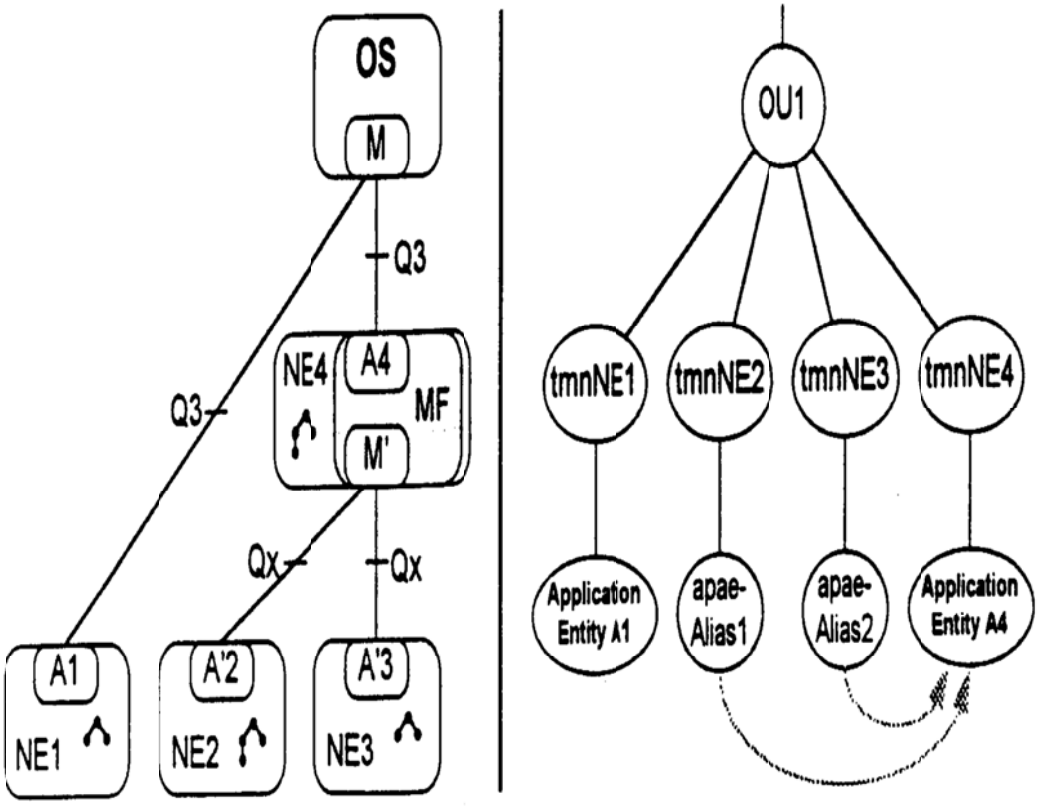


Figure 2 - DIT naming structure

Figure 2 shows the naming structure of the Directory Information Tree for Directory Service for TMN and SONET. Subclause A.6.5.1 further defines each structural object class and its superiors. Clause G.4 provides the corresponding structure rules and name forms to form the DIT structure. The numbers identifying each line in Figure 2 correspond to the structure rules in clause G.4.



a) Example of MF Proxy b) Corresponding Directory Object Instances

Figure 3 - Example of a proxy agent for TMN Nes

An application entity related to an NE or an OS will be represented by an applicationEntity entry (or an apaeAlias entry) which is subordinate to the tmnNE entry or tmnOS entry, respectively. For example, the management agent for an NE is represented via an applicationEntity entry subordinate to the tmnNE entry for that NE. Further, an apaeAlias entry can point to a proxy agent residing in another system. One example of how these entries can be used is shown in Figure 3.

Specification of a DIT structure involves identifying applicable structure rule(s) and the corresponding naming attribute(s) used for a structural object class. The structure rule and name form specifications provide indication of the superior object classes and the naming attribute syntax to be used for that structural object class.

This standard calls for support of entries that comply with A.6.5.1 [this implies their association with the structural object class and its superclass(es)].

9 Object Classes

Standard object classes defined in ITU-T Recommendation X.521 shall be supported as specified in 9.1 and as profiled in A.6.4.1. Conformant DSA implementations shall support the TMN applications listed in clause 6 by supporting those object classes and those defined for this standard as specified in 9.2 to 9.4 and profiled in A.6.4.1.2.

Support of these object classes by a DSA conformant with this standard requires the DSA to be able to store, modify, and retrieve – via Directory operations – an entry in the DIT, if the entry is associated with supported object classes and the following conditions are fulfilled:

a) The entry lies within the DIT structure described in clause 8 and in A.6.5.1;
b) The entry contains all MUST CONTAIN attributes of its object classes and any associated content rules;
c) The entry contains no other than MUST CONTAIN and MAY CONTAIN attributes of its object classes and any associated content rules.

Conformant DSAs shall accept entries that explicitly indicate *top* in their object class attribute. Indication of object class *top* is optional in accordance with ITU-T Recommendation X.501 | ISO/IEC 9594-2, 13.3.2.

Support of a standard object class implies support of its mandatory attribute types, and support of its optional attribute types for which support is claimed for the DSA (see A.6.4.2).

A conformant DSA shall reject requests for creation or modification of entries that as a consequence would not fulfill the condition (b) and shall reply with an update error indicating an object class violation.

A conformant DSA may accept or reject requests for creation or modification of entries that as a consequence would not fulfill one or more of the conditions (a) and (c), depending on whether or not the scope of the schema operated by the DSA exceeds the scope of the schema laid down in Annex A. In particular, a conformant DSA may, but is not required to, accept entries that have additional attributes not identified by any of its object classes or content rules.

Support of these object classes by a DUA conformant with this standard requires the DUA to be able to form query and modification operations, and successfully interpret retrieved information, concerning those objects to which it claims conformance, within the scope of its application.

9.1 Supported Common Object Classes

The following object classes, defined in ITU-T Recommendation X.521 | ISO/IEC 9594-7, are supported:

- Country
- Locality
- Organization
- Organizational Person
- Organizational Role
- Organizational Unit
- Group of Names [8]
- Application process
- Application Entity

[8] It may be desirable to group NEs according to the topology in which they are deployed or by their physical location. For example, all NEs belonging to a ring may be part of an NE group. The subject of NE Grouping is for further study. The use of groupOfNames to describe management domains is for further study.

The TMN Directory Service standard also imposes the following content rule on Application Entity:

```
applicationEntityRule CONTENT-RULE  ::=          {
        STRUCTURAL OBJECT CLASS    applicationEntity
        MUST CONTAIN               { supportedApplicationContext }}
```

9.2 Network Element (tmnNE)

The *Network Element* (tmnNE) object class describes a Network Element as described in ITU-T Recommendation M.3010. The NE contains *Network Element Function* (NEF) and may also contain *Mediation Function* (MF), *Operations System Function* (OSF), or both. There is one tmnNE instance-entry in the DIB for each Network Element in the TMN.

```
tmnNE   OBJECT-CLASS ::= {
        SUBCLASS of             {top}
        MUST CONTAIN { commonName |
                                managedElementId |
                                entityAddress }
        MAY CONTAIN             { proprietaryAddress |
                                vendorName |
                                localityName |
                                neType }
        ID                      id-oc-tmnNE}
```

NOTE – The entityAddress is intended to relate an address to a physical entity and does not necessarily imply that application entities reside at that physical location.

9.2.1 MIB/DIB Synchronization

A tmnNE entry in the DIB will typically reflect an instance of managedElement object class or its subclasses in the Management Information Base. Changes affecting the MIB will cause corresponding changes in the parallel DIB entries. The mechanisms to trigger such changes are based on attribute value changes in the managed objects. The implementation and timing of these mechanisms is a local matter. The DAP may be used by the DUA as the protocol to communicate these changes to the DSA. The MIB is the source of the MIB/DIB synchronized information contained in the DIB.

The following describes the selected attributes in the managedElement object class in the MIB and the corresponding attributes in the tmnNE object class in the DIB that need to be synchronized:

- *commonName*: Values of this multi-valued attribute may be used to provide alternate names for searching. If the "userLabel" conditional package is present in the MIB object, its value shall be replicated as one of the values of the commonName DIB attribute.[9]

[9] Note that the need for alternate names to be unique within some domain is not a Directory-enforced requirement, but would depend upon whatever mechanism is assigning the alternate names (e.g., values replicated from the MIB are assigned and verified within the context of the MIB, and are assumed by the Directory to be valid values within that context).

- *managedElementId*: This attribute replicates the value in the managedElementId attribute of the MIB object managedElement.
- *vendorName*: If the vendorNamePackage conditional package is present in the MIB object, the vendorName DIB attribute replicates the value stored in the vendorName MIB attribute.
- *localityName*: If the locationNamePackage conditional package is present in the MIB object, the localityName DIB attribute replicates the value stored in the locationName MIB attribute.

NOTE – The syntaxes for these attributes in the MIB and DIB are not identical. However, if the MIB values are limited to those characters contained in TeletexString,[10] the comparison procedure described in ITU-T Recommendation X.520 will give the desired result.

9.3 Operations System (tmnOS)

The Operations System (tmnOS) object class describes an Operations System (OS) as described in ITU-T Recommendation M.3010. The OS contains the Operations System Function (OSF) and may also contain any combination of MF, QAF, and WSF. There is one tmnOS instance-entry in the DIB for each Operations System in the TMN.

```
tmnOS  OBJECT-CLASS ::= {
       SUBCLASS of              {top}
       MUST CONTAIN {commonName | entityAddress }
       MAY CONTAIN              { proprietaryAddress |
                                  vendorName | localityName | osType |
                                  otherSupportedFunctionalBlocks}
       ID                       id-oc-tmnOS}
```

9.3.1 MIB/DIB Synchronization

The attribute systemId defined for naming System managed objects in ITU-T Recommendation X.721 is used here for synchronizing the MIB and DIB.

NOTE – A managed object to represent an OS or a management system is not currently defined. However, this issue is being addressed in Open Distributed Management Architecture.

9.4 Directory Network (directoryNetwork)

The Directory Network (directoryNetwork) object class describes the network that is supported by this instance of the directory.

```
directoryNetwork OBJECT-CLASS ::= {
       SUBCLASS OF   {top}
       MUST CONTAIN {commonName | networkId | entityAddress}
       ID                       id-oc-network}
```

[10] The TeletexString is defined in ITU-T Recommendation X.680 (2008) | ISO/IEC 8824-1: 2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

9.5 Complex Network Element (tmnNEComplex)

The Complex Network Element (tmnNEComplex) object class describes a collection of network elements. An OS can reference and manage one or more NE(s) belonging to the complex represented by an instance of this object class.

```
tmnNEComplex OBJECT-CLASS ::= {
        SUBCLASS OF   {top}
        MUST CONTAIN {commonName |managedElementComplexId|entityAddress}
        ID                      id-oc-tmnNEComplex}
```

9.6 SDH Network Element Entry (sdhNEEntry)

The *Synchronous Digital Hierarchy* (SDH) Network Element Entry (sdhNEEntry) object class describes an SDH/SONET Network Element.

```
sdhNEEntry      OBJECT-CLASS ::= {
                SUBCLASS of             {top}
                KIND                    auxiliary
                MAY CONTAIN             { nodeIdInfo }
                ID                      id-oc-sdhNEEntry}
```

9.7 Application Process/Application Entity Alias (apaeAlias)

The Application Process/Application Entity Alias object class is used to provide a means of including – under an entry for an NE, an OS, or an Application Process – aliases for Application Processes and Application Entities that are not resident on the NE or OS itself. The commonName attribute is used to provide a means whereby the alias entry can be referenced during administrative operations.

```
apaeAlias       OBJECT CLASS ::= {
        SUBCLASS OF   {alias}
                MUST CONTAIN { commonName } -- used to provide an RDN for
                                        the alias entry itself --
                ID                      id-oc-apaeAlias}
```

10 Attribute Types

The attribute types described in 10.1 and listed in A.6.4.2.1 are defined in ITU-T Recommendation X.520 | ISO/IEC 9594-6. They are used in conjunction with the object classes specified in clause 9, as either mandatory or optional attributes. They shall be supported as specified in A.6.4.2.1. Subclause 10.1 also lists the managedElementId, managedElementComplexId, and networkId from ITU-T Recommendation M.3100, which shall be supported as specified in A.6.4.2.3. Certain additional attributes are defined by this standard for support by conformant implementations as specified in 10.2 to 10.8 and profiled in A.6.4.2.3.

A DSA conformant with this standard shall support an attribute type as follows:

a) The DSA shall perform – on the original inclusion or on a subsequent modification attempt of an attribute – the checking algorithm that is associated with the syntax of the attribute, when required (see clause 11).
b) The DSA shall check that the number of attribute values complies with the multivalued element of the attribute definition.
c) The DSA shall check that the attribute value(s) conform with the bounds defined in ITU-T Recommendation X.520 | ISO/IEC 9594-6, Annex C (which is not an integral part of the Recommendation | standard).
d) The DSA shall support the matching algorithm, if any, that is associated with the syntax of the attribute (see clause 11), and shall execute this matching algorithm in a manner that conforms with ITU-T Recommendation X.500 series | ISO/IEC 9594 requirements as clarified in clause 11.

Support of these attribute types by a DUA conformant with this standard requires the DUA to be able to form query and modification operations, and successfully interpret retrieved information, using the attributes to which it claims conformance, within the scope of its application.

10.1 Supported Common Attribute Types

The attribute types required to implement the common object classes identified in 9.1 are defined in ITU-T Recommendation X.520, and are supported for this standard. Specifically, support for the following attributes defined in ITU-T Recommendation X.520 are required for object classes defined for TMN:

- Common Name
- Locality Name

Support for the following attribute type defined in ITU-T Recommendation M.3100 is required for object classes defined for TMN:

- managedElementId

10.2 Proprietary Address

The *Proprietary Address* attribute type is an optional, single-valued "printable string" for proprietary use.

proprietaryAddress ATTRIBUTE ::= {

WITH SYNTAX	**PrintableString{ub-proprietary-address}**
EQUALITY MATCHING RULE	**caseIgnoreMatch**
SUBSTRINGS MATCHING RULE	**caseIgnoreSubstringsMatch**
SINGLE VALUE	**TRUE**
ID	**id-at-proprietaryAddress}**

10.3 Node ID Information

The *nodeIdInfo* attribute type specifies a set of pairs (ringId, nodeId) that is the node number associated with an NE. A Node ID value is an integer between 0 and 15 which represents a node in a Bidirectional Line Switched Ring topology. NodeIdInfo is this node number that is optionally made unique by the inclusion of its ring identifier.

An attribute value for NodeIdInfo is an integer accompanied by an optional directory string. The nodeIdinfo attribute is optional.

```
nodeIdInfo ATTRIBUTE   ::=              {
        WITH SYNTAX                             NodeNumberUID
        EQUALITY MATCHING RULE                  nodeIdInfoMatch
        ID                                      id-at-nodeIdInfo }

NodeNumberUID           ::=     SET {
        nodeId                  [0]     Integer (0..ub-nodeId) OPTIONAL,
        ringId                  [1]     DirectoryString (SIZE (1..ub-ringId)) OPTIONAL }

ub-nodeId       INTEGER         ::= 15
ub-ringId       INTEGER         ::= 16
```

The *nodeIdInfoMatch* rule compares for equality a presented nodeIdInfo value with an attribute value of type NodeIdInfo.

```
nodeIdInfoMatch MATCHING-RULE       ::=     {
        SYNTAX                  NodeNumberUID
        ID                      id-mr-nodeIdInfoMatch }
```

The rule returns TRUE if and only if the nodeId component is absent from the attribute value or matches the corresponding component from the presented value according to the IntegerMatch rule, and the ringId component is absent from the attribute value or matches the corresponding component from the presented value according to the caseIgnoreMatch rule.

Where the strings being matched are of different ASN.1 syntax, the comparison proceeds as normal so long as the corresponding characters are in both character sets. Otherwise, matching fails.

10.4 NE Type

The NE Type attribute is used to designate one or more roles the NE plays in one or more domains in which it participates. Specification of valid combinations of this multivalued attribute, and uniqueness criteria within any designated domain, is beyond the scope of this standard and beyond the capability of the Directory to enforce.

```
neType ATTRIBUTE ::=  {
        WITH SYNTAX                     OBJECTIDENTIFIER
        EQUALITY MATCHING RULE          objectIdentifierMatch
        ID                              id-at-neType}
```

10.5 OS Type

The OS Type (osType) attribute is used to designate the primary function of the OS. Specification of valid combinations of this multivalued attribute, and uniqueness criteria within any designated domain, is beyond the scope of this standard and beyond the capability of the Directory to enforce.

```
osType ATTRIBUTE ::= {
        WITH SYNTAX                 OBJECT IDENTIFIER
        EQUALITY MATCHING RULE      objectIdentifierMatch
        ID                                  id-at-osType}
```

10.6 Other Supported Functional Blocks

The Other Supported Functional Blocks attribute is used to identify an OS's other supported functional blocks, as defined in ITU-T Recommendation M.3010. This is a multivalued attribute.

```
otherSupportedFunctionalBlocks ATTRIBUTE ::= {
        WITH SYNTAX                 OBJECT IDENTIFIER
        EQUALITY MATCHING RULE      objectIdentifierMatch
        ID                          id-at-otherSupportedFunctionalBlocks}
```

10.7 Vendor Name

The *Vendor Name* attribute type is used to identify the vendor/supplier of the NE.

```
vendorName  ATTRIBUTE       ::= {
        SUBTYPE OF          name
        WITH SYNTAX         DirectoryString {ub-vendor-name}
        ID                  id-at-vendorName}
```

10.8 Network Address (entityAddress)

The *Network Address* (or *entityAddress*) attribute type may contain octets of any value (including zeroes), and is usually not a printable string. The length of the network address is from 1 to 20 octets (if it exists). For a tmnNE that supports an OSI interface, the entityAddress is an NSAP.

```
entityAddress ATTRIBUTE       ::= {
        WITH SYNTAX                         OCTETSTRING (SIZE (0..20))
        EQUALITY MATCHING RULE              caseIgnoreMatch
        SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
        ID                                  id-at-entityAddress}
```

11 Attribute Syntaxes

Implementations conformant with this standard shall support the attribute syntaxes of the attributes listed in A.6.4.2.

Support of an attribute syntax by a DSA implies:

a) The ability to check values of a supported attribute syntax for syntactical correctness in compliance with limitations as defined by ITU-T Recommendation X.520, Annex C, and additional rules stated in ISO/IEC ISP 10616, clause 9.
b) The ability to perform matches for equality, for matching substring, and for relative ordering if required by the attribute syntax definition, according to rules defined in ITU-T Recommendation X.520 and additional rules stated in ISO/IEC ISP 10616, clause 9. Tables in A.6.4 profile specific matching rules that shall be supported.

A DSA shall return a suitable error if an operation requires it to act on or store an attribute or *Attribute Value Assertion* (AVA) of a type unsupported by the DSA.

Support of an attribute syntax by a DUA implies appropriate formatting of attribute values being sent to the DSA and suitable handling of attribute values retrieved.

The NodeNumberUID attribute syntax and nodeIdInfoMatch matching rule for the nodeIdInfo attribute are defined by this standard in 10.3.

12 Directory Access

Table 1 lists the operations, defined in ITU-T Recommendation X.511 | ISO/IEC 9594-3, which are supported for access to the Directory for TMN. The "General DUA Rights" column indicates a default access control policy (DUA to DSA).

Table 1 - Directory access operations

Operation	General DUA rights
Directory Bind	any 'known' DUA
Directory Unbind	binder
Read	any 'known' DUA
Compare	any 'known' DUA
Abandon	binder or bindee
List	any 'known' DUA
Search	any 'known' DUA
AddEntry	an admin DUA or owner of entry
RemoveEntry	an admin DUA or owner of entry
ModifyEntry	an admin DUA or owner of entry
ModifyDN	an admin DUA

The manner in which a DUA is "known" to a DSA may be based on:

- An entry for the DUA AE in the DIB segment;
- An entry for an NE or an OS with the associated Network Address;
- Inclusion in some domain described in the DIB; or
- Other criteria.

Administrative DUAs are identified in similar manners. The X.500 standards include concepts for security, specifically access control to the DIB (ITU-T Recommendation X.501 clause 16). The use of these features is for further study.

13 DIB Population

Entry and maintenance of the Directory information is accomplished by a human administrator and/or application, and may utilize a DUA or other means which is outside the scope of this standard. Portions of DIB population may be accomplished by automatic registration and maintenance in which the Directory modify operations (addEntry, removeEntry, modifyEntry, modifyDN) are used by an appropriate agent AE.

Once an application determines that a Directory modification is required, a DUA *Application Service Element* (ASE) is used to invoke the appropriate operation over the DUA-DSA interface, as described in clause 12.

This standard defines one automatic mechanism that facilitates automatic registration of certain entries under certain conditions.

13.1 DIB Population of NE Entries by Registration Manager & Registration Agents

This clause defines a means to accomplish the initial creation of tmnNE entries (and subordinate entries) only, under the following conditions:

a) A suitable RM exists within an *Intermediate System* (IS) Level 1 Domain;
b) An NE has implemented an RA and followed the conventions for fixed selectors in the presentation address of the RA; and
c) Security is assumed to be based on NE existence in the IS routing tables.

13.1.1 The Model

As noted in clause 7, automatic registration of certain entries into the DIB is optionally provided by a Registration Manager which uses the Registration Request Protocol to communicate with a Registration Agent on remote network elements.

The RM uses information from the *Intermediate System to Intermediate System* (IS-IS) and *End System to Intermediate System* (ES-IS) routing protocols defined in ISO/IEC 10589 and ISO 9542 to detect when a new NE has been added to (or is once again reachable in) its domain of responsibility. Specification of the mechanism by which this detection is accomplished is beyond the scope of this standard. Upon detecting such an event, the RM uses the network address of the newly reachable NE to form a presentation address for the target RA. It sets up an association with the RA in order to invoke the Automatic Registration of the NE's entry into the DIB. The RA then uses a DUA and the Directory Access operations to create its entry in the DIB. The means by which the RA uses the DUA is beyond the scope of this standard.

13.1.2 Protocol Overview

The Registration Request Protocol is used to support Automatic Registration. The RRP operates between a Registration Manager and a Registration Agent usually residing on an R-NE. The RM sends the RA a message containing the addressing information of the primary DSA with which registration is to occur. It may also provide addressing information for other DSAs for alternate use by the NE for Directory Service. The RA then provides this information to the DUA residing on the R-NE, and causes the DUA to perform the add procedure, which adds the R-NE information to the Directory Information Base on the DS.

The RRP uses the services of the Remote Operations Service Element. Four macros are used by the RRP:

- *BIND* (used to establish an association);
- *UNBIND* (used to release an association);
- *OPERATION* (used to specify operations and their arguments); and
- *ERROR* (used to specify negative results of operations).

These macros map to the underlying primitives, messages, and state machines of the ROSE.

The Association Control Service Element is used to establish and release an application-association between a pair of Application Entities. Associations between an RM and an RA may only be initiated by the RM.

13.1.3 Use of Underlying Services

The RRP makes use of underlying services as described below.

13.1.3.1 Use of ROSE Services

The Remote Operations Service Element is defined in ITU-T Recommendation X.219. The ROSE supports the request/reply paradigm of remote operations.

The *registrationElement* Application Service Element is a user of the RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U, and RO-REJECT-P services of the ROSE.

The remote operations of the RRP are Class 1 (synchronous) operations.

RRP uses Association Class 1. This means the RA cannot invoke operations on the RM.

13.1.3.2 Use of ACSE Services

The Association Control Service Element is defined in ITU-T Recommendation X.217. The ACSE provides for the establishment, release, and abort of application-associations between Application Entities.

The RegistrationBind and RegistrationUnbind are the sole users of the A-ASSOCIATE and A-RELEASE services of the ACSE in normal mode. The application process is the user of the A-ABORT and A-P-ABORT services of the ACSE.

13.1.3.3 Use of the Presentation Service

The presentation-service is defined in ITU-T Recommendation X.216. The Presentation Layer coordinates the representation (transfer syntax) of the Application Layer information (as described by ASN.1 notation) to be exchanged.

In normal mode, a different presentation-context is used for each abstract-syntax included in the application context. The abstract-syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules. Other transfer syntaxes may be optionally supported.

The ACSE is the sole user of the P-CONNECT, P-RELEASE, P-U-ABORT, and P-P-ABORT services of the presentation-service.

The ROSE is the sole user of the P-DATA service of the presentation-service.

13.1.3.4 Elements of Procedure

The Registration Manager for the Level 1 routing area sets up an association with the Registration Agent on the R-NE. The presentation address of the RA is constructed from the NSAP of the R-NE (obtained in a manner described earlier in the standard) and the standard RA upper layer selectors defined in this standard. The user data in the A-ASSOCIATE contains the RRP BIND arguments. In the BIND arguments, the RA and RM exchange protocol version information, which is a list of the versions each supports. The highest version supported by both sides is the protocol version used. If there are no versions in common, the RM releases the association and discontinues Automatic Registration.

Once the association is successfully established, the RM uses an RO-INVOKE request to perform the registrationRequestOperation. The OPERATION argument specifies the address of the DSA application entity on the DS. If alternate DSs are deployed, the argument also specifies their addresses.

The argument also may specify an optional namePrefix. This contains the initial naming information, or prefix, of the Distinguished Name of the R-NE. The amount of naming information in the namePrefix is variable. This means the namePrefix may supply the entire Distinguished Name of the R-NE, the DN of the R-NE's containing object, or the DN of an object anywhere on the path from the root to the R-NE in the Directory Information Tree. The RA combines the namePrefix, if present, with local information to construct the complete DN of the R-NE for use in adding the R-NE entry to the Directory.

If the RA determines that the dsaAddress(es) are syntactically valid addresses, it saves these values and returns success (a null RESULT), indicating that it has successfully received the DSA address(es). Otherwise it returns an error and the automatic registration is deemed a failure. In either case, the RM issues an A-RELEASE to perform an UNBIND.

Association by the RM with newly reachable NEs, and invocation of the automatic registration, shall be attempted up to three (3) times at *n* second intervals [locally configurable, with the default equal to sixty (*n* = 60) seconds]. If all three attempts to register an NE fail, then the NE (identified by its network address) shall be deemed "unresponsive" and no further attempt shall be made by the RM to register the NE unless it becomes "newly reachable" once again (i.e., the NE's address disappears and then reappears in the IS routing tables).

The RA then issues an addEntry request via the DUA on the R-NE to add the R-NE's entry to the DIB on the primaryDsa.

13.1.4 Registration Agent Selectors

For Registration PDUs used in the set-up of an association with the Registration Agent, the initial Presentation Selector value shall be set to “FB” (hex).

14 Directory Distribution

A DS maintains a portion of the global DIB. A DSA provides access to this DIB segment, and may also provide access to the Directory information held in other DIB segments by cooperating with other DSAs. In general, access to the Directory information is provided by a set of one or more DSAs that collectively constitute the distributed directory service.

The procedures for distributed operation of the Directory are specified in ITU-T Recommendation X.518 | ISO 9594-4. The Directory System Protocol defines the exchange of requests and outcomes between two DSAs. Within TMN, chaining is preferred over referrals in satisfying access requests requiring distributed operations.

Further refinement of the use of DSP is for further study.

15 Replication

Shadow updates shall be triggered by any change in the shadowed information. The master DSA shall initiate shadow updates.

For a given area to be replicated, all data and knowledge references shall be shadowed.

Participating DSAs shall be capable of supporting both shadow supplier and shadow consumer roles.

If a master DSA fails, a shadowing DSA may be re-configured to support the role of master DSA for the information being shadowed. Knowledge references pertaining to the master and shadow DSAs shall be updated to reflect the role reversal. A standardized mechanism by which this is accomplished is for further study.

Participating DSAs shall be capable of supporting the Reliable Transfer Service Element (RTSE) of the OSI application layer.

Participating DSAs shall be capable of supporting the shadowSupplierInitiatedAC application context in accordance with ITU-T Recommendation X.525.

Annex A
(normative)

A PICS for Information

This annex profiles information that is to be stored within the TMN Directory and that is common to a variety of applications. Information that is specific to certain applications may be profiled by other documents.

The Directory has been designed to support multiple applications, drawn from a wide range of possibilities. The nature of the totality of applications supported will govern which objects are stored in the Directory. This annex specifies the minimum set of structure and naming elements for a conformant DSA and other minimum schema requirements. This standard does not limit DSAs to these minimum capabilities.

The Directory Access Protocol and the Directory System Protocol, as defined by ITU-T Recommendation X.500 series (ITU-T Recommendation X.519) | ISO/IEC 9594 and profiled within other annexes of this standard, can be used to access information stored in a Directory Information Base fragment, which is profiled in this annex.

The supplier of a DSA or DUA implementation that is claimed to conform to this standard is required to complete a copy of the PICS proforma provided in this annex and is required to provide the information necessary to identify both the supplier and the implementation.

This annex is based on a draft version of the Protocol Implementation Conformance Statement Proforma for the Directory Access Protocol 1993 Standard as contained in document ISO/IEC JTC 1/SC 21/WG 4 N 2106 [*Call for Contributions on the Protocol Implementation Conformance Statement (PICS) Proforma for the Directory Access Protocol 1993 Standard*], dated 16 December 1994. It uses only the tables defined in that proforma that pertain to Directory information. The numbering of the common PICS Proforma is retained in order to facilitate completion of the entire proforma by implementors.

A.1 Identification of the Implementation

A.1.1 Identification of PICS

Item No.	Question	Response
1	Date of Statement (DD/MM/YY)	
2	PICS Serial Number	
3	System Conformance Statement Cross Reference	

A.1.2 Identification of the Implementation and/or System

Item No.	Question	Response
1	Implementation Name	
2	Version Number	
3	Machine Name	
4	Machine Version Number	
5	Operating System Name	
6	Operating System Version No.	
7	Special Configuration	Note 1
8	Other information	

NOTE 1 – Please enter one or more of the following:

- DUA for connection to centralized DSAs;
- DUA for connection to cooperating DSAs;
- Centralized DSAs;
- Cooperating DSAs; and
- First-level DSAs.

A.1.3 Identification of the System Supplier and/or Test Laboratory Client

Item No.	Question	Response
1	Organization Name	
2	Contact Name(s)	
3	Address	
4	Telephone Number	
5	Telex Number	
6	Fax Number	
7	E-Mail Address	
8	Other information	

A.2 (not used)

A.3 Global statement of Conformance

If the supplied implementation is a DSA implementation, A.3.1 is required to be answered by the supplier.

If the supplied implementation is a DUA implementation, A.3.2 is required to be answered by the supplier.

Answering "No" to A.3.1.1 or A.3.2.1 indicates nonconformance to the specification. Nonsupported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is nonconformant. Such information shall be provided in A.6.5 "Other Information".

A.3.1 DSA Implementation and/or System

Item No.	Question		Status	Support	Predicate Name
1	Are all mandatory general capabilities for the DSA implemented?		m		
2	Are all mandatory First-level DSA requirements (ISO/IEC 9594-4) implemented?		c0		
3	Are minimum knowledge requirements (ISO/IEC 9594-2) implemented?		m		
4	Other Supported Reference(s)	Cross Reference	o		
		Nonspecific Subordinate Reference	o		
		Immediate Superior Reference	o		
5	Supported Security Level(s)	none	o.1		
		simple	o.1		Simple-DSA
		strong	o.1		Strong-DSA
		external	i		
6	Is asynchronous (ROSE class 2) mode of operation supported?		m		
7	Does the DSA follow the rules of extensibility as defined in 7.5 of ISO/IEC 9594-5?		m		
8	Is the alias mechanism implemented?		m		
9	Does the DSA support the directoryAccessAC application-context?		m		
10	Supported Access Controls	Simplified Access Control	o		SimpleAC-DSA
		Basic Access control	o		BasicAC-DSA
		Other	i		
11	Is the DSA capable of supporting collective attributes?		o		
12	Is the DSA capable of supporting operational attributes?		m		
13	Is the DSA capable of supporting hierarchical attributes?		o		
14	Is the DSA capable of supporting auxiliary object classes?		oc:m		A.3.1/16
15	Is the DSA capable of supporting the subschema for its portion of the DIT?		o		
16	Is the DSA capable of supporting SONET-specific extensions?		o		sdhUsage-DSA

o.1 The DSA shall support at least one security level.

c0: If the special configuration in item A.1.2/7 is First-level DSA, then m; else i.

A.3.2 DUA Implementation and/or System

Item No.	Question		Status	Support	Predicate Name
1	Are all mandatory general capabilities for the DUA implemented?		m		
2	Is the directoryAccessAC application-context supported?		m		
3	Supported Security Level(s)	none	o.2		
		simple	o.2		Simple-DUA
		strong	o.2		Strong-DUA
		external	i		
4	Is asynchronous (ROSE class 2) mode of operation supported?		o		
5	Supported Access Controls	Simplified Access Control	o		SimpleAC-DUA
		Basic Access Control	o		BasicAC-DUA
		Other	i		
5	Does the DUA follow the rules of extensibility as defined in 7.5 of ISO/IEC 9594-5?		m		
6	Does the DUA support SONET-specific extensions?		o		sdhUsage-DUA

o.2 The DUA shall support at least one security level.

A.4 Instruction for Completing the PICS Proforma

A.4.1 Definition of Support

A capability is said to be supported if the *Implementation Under Test* (IUT) is able:

- To generate the corresponding operation parameters (either automatically or because the end user requires that capability explicitly).
- To interpret, handle, and – when required – make available to the end user the corresponding error or result.

An object class is said to be supported if the IUT is able to construct entries of that object class. Support of an object class also requires support of the object identifier(s) of its superclass(es) of that object class.

An attribute type is said to be supported by a DUA implementation if the DUA supports those aspects of the attribute syntax that are pertinent to encoding, decoding, or both of the attribute.

An attribute type is said to be supported by a DSA implementation if the DSA supports a subset or all aspects of the attribute syntax of the attribute and stores the attribute value(s) where appropriate.

A.4.2 Status Column

This column indicates the level of support required for conformance to the ISO/IEC standard.

The Status (DUA) indicates that the implementation under test is a DUA and the Status (DSA) indicates that the implementation under the IUT is a DSA.

The values are as follows:

m Mandatory support is required.

o Optional support is permitted for conformance to the standard. If implemented, it shall conform to the specifications and restrictions contained in the standard. These restrictions may affect the optionality of other items.

c The item is conditional (support of the capability is subject to a predicate).

c: m The item is mandatory if the predicate is true; optional otherwise.

- The item is not applicable.

i The item is outside the scope of this PICS.

Wherever this profile differs from the base standard (as specified by the 1993 DAP PICS Proforma), two characters are used in the status column, with the first character indicating the level of support required by the base standard, and the second character indicating the level of support required by this profile for the TMN Directory Service. Thus, "om" indicates the feature is optional in general but mandatory for TMN; "oc:m" indicates the feature is optional in general but mandatory under the stated condition for TMN (e.g., for SONET only if A.3.1/16 or A.3.2/6 are cited as predicates).

A.4.3 Support Column

This column shall be completed by the supplier or implementor, to indicate the level of implementation of each item. The proforma has designed such that values required are:

Y Yes, the item has been implemented.

N No, the item has not been implemented.

- The item is not applicable;

In the PICS proforma tables, every leading item marked “m” shall be supported by the IUT. Subitems marked “m” shall be supported if the corresponding leading item is supported by the IUT.

A.4.4 Note Column

This column indicates the following:

notexx Refers to Note xx.

d(xx) A default value xx within ( ) is defined in the standard. When absent in the *Protocol Data Unit* (PDU), both sender and receiver shall interpret it as having the default value specified in the standard.

A.4.5 Predicate Column

The item number contained in the predicate column, if any, means that the status in the "Status" column applies only when the PICS states that one or more features identified by the predicate item is supported.

A.4.6 Item Reference Numbers

Each line within the PICS proforma that requires implementation details to be entered is numbered at the left-hand edge of the line. This numbering is included as a means of uniquely identifying all possible

implementation details within the PICS proforma. This referencing is used both inside the PICS proforma, and for references from other test specification documents.

The means of referencing individual responses is done by the following sequence:

- A reference to the smallest subsection enclosing the relevant item.
- A solidus character, ”/”.
- The reference number of the row in which the response appears.
- If, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labeled a, b, c, etc.. from left to right, and this letter is appended to the sequence.

An example of the use of this notation would be A.6.4.1.1/4, which refers to the support for the locality object class.

A.5 (not used)

A.6 Capabilities & Options

A.6.1 (not used)

A.6.2 (not used)

A.6.3 (not used)

A.6.4 Directory Schema

A.6.4.1 Supported Object Classes

A.6.4.1.1 Standard Object Classes

The supplier of the implementation shall indicate, in the table below, the selected object classes defined in ITU-T Recommendation X.521 | ISO/IEC 9594-7 for which conformance is claimed.

Item No.	Object class	Status	Support	Note
1	top	m		
2	alias	m		
3	country	om		
4	locality	om		
5	organization	om		
6	organizationalUnit	om		
7	person	o		
8	organizationalPerson	o		
9	organizationalRole	o		
10	groupOfNames	om		
11	groupOfUniqueNames	o		
12	residentialPerson	o		
13	applicationProcess	om		
14	applicationEntity	om		Note 1
15	dSA	o		
16	device	o		
17	strongAuthenticationUser	o		
18	certificationAuthority	o		

NOTE 1 – Shall contain supportedApplicationContexts attribute.

A.6.4.1.2 TMN-Defined Object Classes

The supplier of the implementation shall indicate, in the table below, the selected object classes defined in this standard (Directory Service for TMN) for which conformance is claimed.

Item No.	Object class	Status	Support	Note/Predicate
1	tmnNE	-m		
2	sdhNEEntry	-c:m		[[sdhUsage-DUA (A.3.2/6)] or [sdhUsage-DSA (A.3.1/16)]]
3	apaeAlias	-m		
4	tmnOS	-m		
5	directoryNetwork	-m		
6	tmnComplex	-m		

A.6.4.1.3 Other Supported Object Classes

The supplier of the implementation is required to list, in the following table, any other object classes provided for which conformance is claimed:

Index	Supported object classes

A.6.4.2 Supported Attribute Types

A.6.4.2.1 Standard Attribute Types

The supplier of the implementation shall indicate, in the following table, the selected attribute types defined in ITU-T Recommendation X.520 | ISO/IEC 9594-6 for which conformance is claimed:

Item No.	Attribute Type	Upperbound	Status	Support	Note
0	objectClass		m		
1	aliasedEntryName		om		
2	knowledgeInformation		o		
3	commonName	64	om		
4	surName	64	o		
5	givenName		o		
6	initials		o		
7	generationQualifier		o		
8	uniqueIdentifier		o		
9	dnQualifier		o		
10	serialNumber	64	o		
11	countryName		om		size = 2
12	localityName	128	om		
13	stateOrProvinceName	128	om		
14	streetAddress	128	o		
15	houseIdentifier	64	o		
16	organizationName	64	om		
17	organizationalUnitName	64	om		
18	title	64	o		
19	description	1024	om		
20	searchGuide		o		
21	enhancedSearchGuide		o		
22	businessCategory	128	o		
23	postalAddress	6(lines) x 30(chs)	o		
24	postalCode	40	o		
25	postOfficeBox	40	o		
26	physicalDeliveryOfficeName	128	o		
27	telephoneNumber	32	o		
28	telexNumber	14, 4, 8	o		
29	teletexTerminalIdentifier	1024	o		
30	fasimileTelephoneNumber	32	o		
31	X.121 Address	15	o		
32	internationalISDNNumber	16	o		

(continued)

(concluded)

Item No.	Attribute Type	Upperbound	Status	Support	Note
33	registeredAddress	6(lines) x 30(chs)	o		
34	destinationIndicator	128	o		
35	preferredDeliveryMethod		o		
36	presentationAddress		om		
37	supportedApplicationContext		om		
38	protocolInformation		o		
39	distinguishedName		om		
40	member		om		
41	uniqueMember		o		
42	owner		om		
43	roleOccupant		o		
44	seeAlso		om		
45	userPassword	128	o		
46	userCertificate		o		
47	cACertificate		o		
48	authorityRevocationList		o		
49	certificateRevocationList		o		
50	crossCertificatePair		o		

A.6.4.2.2 Collective Standard Attribute Types

The supplier of the implementation shall indicate, in the following table, the selected collective attribute types defined in ITU-T Recommendation X.520 | ISO/IEC 9594-6 for which conformance is claimed:

Item No.	Attribute Types	Upperbound	Status	Support	Note
1	collectiveLocalityName	128	o		
2	collectiveStateOrProvinceName	128	o		
3	collectiveStreetAddress	128	o		
4	collectiveOrganizationName	64	o		
5	collectiveOrganizationalUnitName	64	o		
6	collectivePostalAddress	6(lines) x 30(chs)	o		
7	collectivePostalCode	40	o		
8	collectivePostOfficeBox	40	o		
9	collecivePhysicalDeliveryOfficeName	128	o		
10	collectiveTelephoneNumber	32	o		
11	collectiveTelexNumber	14,4,8	o		
12	collectiveTeletexTerminalIdentifier	1024	o		
13	collectiveFacsimileTelephoneNumber	32	o		
14	collectiveInternationalISDNNumber	16	o		

A.6.4.2.3 TMN-Defined Attribute Types

The supplier of the implementation shall indicate, in the following table, the selected attribute types defined in this standard (Directory Service for TMN) for which conformance is claimed:

Item No.	Attribute Types	Upper bound	Status	Support	Note/Predicate
1	proprietaryAddress	128	-m		
2	nodeIdInfo		-c:m		[[sdhUsage-DUA (A.3.2/6)] or [sdhUsage-DSA (A.3.1/16)]]
3	vendorName	256	-m		
4	neType		-m		
5	entityAddress		-m		
6	M.3100::managedElementId		-m		
7	osType		-m		
8	otherSupportedFunctionalBlocks		-m		
9	X.721:systemId		-m		
10	M.3100: networkId		-m		
11	M.3100: managedElementComplexId		-m		

A.6.4.2.4 Other Supported Attribute Types

The supplier of the implementation is required to list, in the following table, any other attribute types provided for which conformance is claimed:

Index	Attribute types

A.6.4.3 Supported Matching Rules

A.6.4.3.1 Standard Matching Rules

The supplier of the implementation shall indicate, in the following table, the matching rules defined in ITU-T Recommendation X.520 | ISO/IEC 9594-6 for which support is claimed:

Item No.	Matching Rule	Status	Support	Note
1	caseIgnoreMatch	om		
2	caseIgnoreOrderingMatch	om		
3	caseIgnoreSubstringsMatch	om		
4	SubstringAssertion	om		
5	caseExactMatch	om		
6	caseExactSubstringsMatch	om		
7	numericStringMatch	om		
8	numericStringOrderingMatch	om		
9	numericStringSubstringsMatch	om		
10	caseIgnoreListMatch	om		
11	caseIgnoreListSubstringsMatch	om		
12	booleanMatch	om		
13	integerMatch	om		
14	integerOrderingMatch	om		
15	bitStringMatch	om		
16	octetStringMatch	om		
17	octetStringOrderingMatch	om		
18	octetStringSubstringsMatch	om		
19	octetSubstringAssertion	om		
20	telephoneNumberMatch	o		
21	presentationAddressMatch	om		
22	uniqueMemberMatch	o		
23	protocolInformationMatch	om		
24	uTCTimeMatch	o		
25	uTCTimeOrderingMatch	o		
26	generalizedTimeMatch	o		
27	generalizedTimeOrderingMatch	o		
28	integerFirstComponentMatch	o		
29	objectIdentifierFirstComponentMatch	om		
30	directoryStringFirstComponentMatch	om		
31	wordMatch	o		
32	keywordMatch	o		

A.6.4.3.2 TMN-Defined Matching Rules

The supplier of the implementation shall indicate, in the following table, the matching rules defined in this standard (Directory Service for TMN) for which support is claimed:

Item No.	Matching Rule	Status	Support	Note/Predicate
1	nodeIdInfoMatch	-c:m		[[sdhUsage-DUA (A.3.2/6)] or [sdhUsage-DSA (A.3.1/16)]]

A.6.4.4 Information Framework

The supplier of the implementation shall indicate, in the following table, the object class, attributes, and matching rules defined in ITU-T Recommendation X.501 | ISO/IEC 9594-2, Information Framework for which support is claimed.

A.6.4.4.1 Information Framework Object Classes

Item No.	Object class	Status	Support	Predicate	Note
1	subentry	c:m		subentries extension supported	
2	accessControlSubentry	c9			
3	collectiveAttributeSubentry	c10			

c9: if [subentries extension supported and [SimpleAC-DSA or BasicAC-DSA]], then m; else n/a.

c10: if [subentries extension supported and A.3.1/11], then m; else n/a.

A.6.4.4.2 Information Framework Attributes

Item No.	Attribute	Status	Support	Predicate	Note
1	createTimestamp	mo			
2	modifyTimestamp	mo			
3	creatorsName	o			
4	modifiersName	o			
5	administrativeRole	mo			
6	subtreeSpecification	c:m		subentries extension supported	
7	collectiveExclusions	c:m		A.3.1/11	

A.6.4.4.3 Information Framework Matching Rules

Item No.	Matching Rule	Status	Support	Predicate	Note
1	objectIdentifierMatch	m			
2	distinguishedNameMatch	m			

A.6.4.5 Subschema Administration

If the supplied implementation supports the subschema for its portion of the DSA claimed in item A.3.1/15, then A.6.4.5.1, A.6.4.5.2, and A.6.4.5.3 is required to be answered by the supplier.

A.6.4.5.1 Subschema Administration Object Classes

Item No.	Object class	Status	Support	Predicate	Note
1	subschema	m			

A.6.4.5.2 Subschema Administration Attributes

Item No.	Attribute	Status	Support	Predicate	Note
1	dITStructureRules	m			
2	dITContentRules	m			
3	matchingRules	m			
4	attributeTypes	m			
5	objectClasses	m			
6	nameForms	m			
7	matchingRuleUse	m			
8	structuralObjectClass	m			
9	governingStructureRule	m			

A.6.4.5.3 Subschema Administration Matching Rules

None.

A.6.4.6 Access Control

A.6.4.6.1 Access Control Object Classes

None.

A.6.4.6.2 Access Control Attributes

Item No.	Attribute	Status	Support	Predicate	Note
1	accessControlScheme	m			
2	prescriptiveACI	c:m		SimpleAC-DSA or BasicAC-DSA	
3	entryACI	c:m		BasicAC-DSA	
4	subentryACI	c:m		SimpleAC-DSA or BasicAC-DSA	

A.6.4.6.3 Access Control Matching Rules

None.

A.6.4.7 DSA Operational Attributes

A.6.4.7.1 DSA Operational Attribute Object Classes

None.

A.6.4.7.2 DSA Operational Attributes

Item No.	Attribute Types	Status	Support	Predicate	Note
1	dseType	m			
2	myAccessPoint	m			
3	superiorKnowledge	c: m		A.3.1/3	
4	specificKnowledge	c: m		A.3.1/3	
5	nonSpecificKnowledge	c: m		A.3.1/4b	
6	supplierKnowledge	c: m		A.3.1/3	
7	consumerKnowledge	c: m		A.3.1/3	
8	secondaryShadows	o			

A.6.4.7.3 DSA Operational Matching Rules

Item No.	Matching Rule	Status	Support	Predicate	Note
1	accessPointMatch	m			
2	masterAndShadowAccessPointsMatch	o			
3	supplierOrConsumerInformationMatch	o			
4	supplierAndConsumerMatch	c: m		A.3.1/3	

A.6.4.8 Message Handling System (MHS) Attribute Types

Item No.	Attribute	Status	Support	Note
1	mhsDeliverableContentLength	o		
2	mhsDeliverableContentTypes	o		
3	mhsDeliverableEits	o		
4	mhsDLMembers	o		
5	mhsDLSubmitPermissions	o		
6	mhsDLMessageStoreName	o		
7	mhsORAddresses	o		
8	mhsPreferredDeliveryMethods	o		
9	mhsSupportedAutomaticActions	o		
10	mhsSupportedContentTypes	o		
11	mhsSupportedOptionalAttributes	o		

A.6.5 Other Information

A.6.5.1 Minimum set of Structure & Naming Elements

In order to specify the minimum set of structure and naming elements, this subclause lists a set of structure elements for selected object classes corresponding to the structure shown in Figure 2. Entries that comply with one of these elements form the minimum structure capability of the DIT.

Specification of a structure element involves:

- Assigning an identifier to it;
- Indicating the structural object class to which it applies;
- Indicating its superior structure elements; and
- Indicating the naming attributes of the structural object class to which it applies.

An entry complies with a structure element if:

- Its superior entry complies with at least one of the superior structure elements;
- It is associated with the structural object class; and
- Its Relative Distinguished Name is formed using the naming attribute(s).

The form of an entry's name is determined by the structure element with which the entry complies: the attribute(s) for the entry's RDN are provided by the naming attribute(s) indicated by this DIT structure rule and the form of the superior's name (which is part of the entry's name) is determined by the superior structure element. If there are multiple structure elements for the same object class, multiple forms for naming entries of this object class are assigned by these structure elements.

Table A.1 below defines the structure and naming elements for the DIT for TMN. A conformant DSA shall support this minimum set.

The first row is for descriptive purposes only; there is no root entry, and thus no object class and no structure element for the root.

For the purposes of this standard, the corresponding structural object class of an entry (i.e., the one to which the structure elements described above relate) shall be taken as that of the lower-most registered structural object class for the entry.

Table A.1 - Structure & Naming Elements for the DIT for TMN

Ref. no.	Structure Element	Structural Object Class	Superior Struct. Element	Naming Attribute	Note
	0	Root			
1	1	Country	0	Country Name	
2	2	Organization	0	Organization Name	
3	3	Organization	1	Organization Name	
4	4	Organization	5, 6, 7, 35	Organization Name	
5	5	Locality	0	Locality Name	
6	6	Locality	1	Locality Name	
7	7	Locality	5, 6, 7, 35	Locality Name	Notes 1,3
8	8	Organizational Unit	2, 3, 4	Organizational Unit Name	
9	9	Organizational Unit	5, 6, 7, 35	Locality Name	
10	10	Organizational Unit	8, 9, 10	Locality Name, Organizational Unit Name	Note 4
11	11	Group of Names	5, 6, 7, 35	Common Name	
12	12	Group of Names	2, 3, 4	Common Name	
13	13	Group of Names	8, 9, 10	Common Name	
14	14	TMN Operations System	2, 3, 4	Common Name	
15	15	TMN Operations System	5, 6, 7, 35	Common Name	
16	16	TMN Operations System	8, 9, 10	Common Name	
17	17	TMN Network Element	2, 3, 4	Common Name	Note 5
18	18	TMN Network Element	5, 6, 7, 35	Common Name	Note 5
19	19	TMN Network Element	8, 9, 10	Common Name	Note 5
20	20	TMN Network Element	14, 15, 16	Common Name	Note 5
21	21	Application Process	2, 3, 4	Common Name	
22	22	Application Process	8, 9, 10	Common Name	
23	23	Application Process	5, 6, 7, 35	Common Name	
24	24	Application Process	14, 15, 16	Common Name	
25	25	Application Process	17, 18, 19, 20, 43	Common Name	
26	26	Application Entity	2, 3, 4	Common Name	Note 2
27	27	Application Entity	8, 9, 10	Common Name	Note 2
28	28	Application Entity	5, 6, 7, 35	Common Name	Note 2
29	29	Application Entity	14, 15, 16	Common Name	Note 2
30	30	Application Entity	17, 18, 19, 20, 43	Common Name	Note 2
31	31	Application Entity	21, 22, 23, 24, 25, 48	Common Name	
32	32	Application Process/ Application Entity Alias	14, 15, 16	Common Name	

(continued)

Table A.1– Structure & Naming Elements for the DIT for TMN *(concluded)*

Ref. no.	Structure Element	Structural Object Class	Superior Struct. Element	Naming Attribute	Note
33	33	Application Process/ Application Entity Alias	17, 18, 19, 20, 43	Common Name	
34	34	Application Process/ Application Entity Alias	21, 22, 23, 24, 25, 48	Common Name	
35	35	Locality	1	State or Province Name	
36	36	Organizational Role	2, 3, 4	Common Name	
37	37	Organizational Role	8, 9, 10	Common Name	
38	38	Organizational Person	2, 3, 4	Common Name	
39	39	Organizational Person	8, 9, 10	Common Name	
40	40	Directory Network	5, 6, 7, 35	Common Name	
41	41	Directory Network	2, 3, 4	Common Name	
42	42	Directory Network	8, 9, 10	Common Name	
43	43	TMN Network Element	40, 41, 42, 44	Common Name	Note 5
44	44	Directory Network	40, 41, 42, 44	Common Name	
45	45	TMN Complex Network Element	2, 3, 4	Common Name	
46	46	TMN Complex Network Element	8, 9, 10	Common Name	
47	47	TMN Complex Network Element	5, 6, 7, 35	Common Name	
48	48	Application Process	45, 46, 47	Common Name	
49	49	Application Entity	45, 46, 47	Common Name	Note 2
50	50	Application Process/ Application Entity Alias	45, 46, 47	Common Name	

NOTES

1 ITU-T Recommendation X.500 series | ISO/IEC 9594 suggest that entries of object class locality can be immediate subordinates of organization entries. It is suggested that the Directory Service for TMN use organizational unit entries named by locality names instead, as this seems to be the appropriate semantic of such entries. Even if the organizational unit is identified by a locality name, it is in fact an organizational unit.

2 The inclusion of superior structural elements other than 21 through 25, or 48, is not within the scope of ITU-T Recommendation X.521 | ISO/IEC 9594-7, Annex B; it does, however, reflect the merging of Application Process and Application Entity implied by ITU-T Recommendation X.521 | ISO/IEC 9594-7 in describing these objects classes.

3 Although the recursive nature of structure element 7 specifies a chain with an arbitrary number of locality entries, the minimum set of structure and naming elements contains not more than 5 locality entries in a chain – i.e., up to 5 localities in a chain shall be supported. This does not prevent conformant DSAs from supporting more than 5 localities in a chain.

4 Although the recursive nature of structure element 10 specifies a chain with an arbitrary number of organizational unit entries, the minimum set of structure and naming elements contains not more than 8 organizational unit entries in a chain – i.e., up to 8 organizational units in a chain shall be supported. This does not prevent conformant DSAs from supporting more than 8 organizational units in a chain.

5 Entries of the tmnNE Structural class may also be members of the sdhNEEntry auxiliary class.

A.6.5.2 Other Implementation Information

The following table can be used to provide any other relevant information:

Index	Other information

Annex B
(normative)

B PICS for Directory Access Protocol (DAP)

A conformant DSA shall support the DirectoryAccessAC application context in accordance with the profile defined herein. A variety of conformant DUAs are possible, depending on the requirements for service by the user(s) of the DUA, be they human or application. For any required service, the necessary elements of protocol as specified below shall be implemented.

The supplier of a DAP implementation that is claimed to conform to the Directory Service for TMN standard is required to complete a copy of the PICS proforma provided below and is required to provide the information necessary to identify both the supplier and the implementation.

This annex is based on a draft version of the Protocol Implementation Conformance Statement Proforma for the Directory Access Protocol 1993 Standard as contained in document ISO/IEC JTC 1/SC 21/WG 4 N 2106 *(Call for Contributions on the Protocol Implementation Conformance Statement (PICS) Proforma for the Directory Access Protocol 1993 Standard)*, dated 16 December 1994. It uses only the tables defined in that proforma that pertain to protocol. The numbering of the PICS Proforma is retained in order to facilitate completion of the entire Proforma by implementors.

B.1 Identification of the Implementation

B.1.1 Identification of PICS

Item No.	Question	Response
1	Date of Statement (DD/MM/YY)	
2	PICS Serial Number	
3	System Conformance Statement Cross Reference	

B.1.2 Identification of the Implementation and/or System

Item No.	Question	Response
1	Implementation Name	
2	Version Number	
3	Machine Name	
4	Machine Version Number	
5	Operating System Name	
6	Operating System Version No.	
7	Special Configuration	Note 1
8	Other information	

NOTE 1 – Please enter one or more of the following:

- DUA for connection to centralized DSAs;
- DUA for connection to cooperating DSAs;

- Centralized DSAs;
- Cooperating DSAs; and
- First-level DSAs.

B.1.3 Identification of the System Supplier and/or Test Laboratory Client

Item No.	Question	Response
1	Organization Name	
2	Contact Name(s)	
3	Address	
4	Telephone Number	
5	Telex Number	
6	Fax Number	
7	E-Mail Address	
8	Other information	

B.2 Identification of the Protocol

Item No.	Question	Response
1	Title, Reference, No., publication date of the protocol standard	
2	Protocol Version Number	
3	Implemented Addenda	
4	Implemented Defect Reports (Reference No.)	

B.3 Global Statement of Conformance

If the supplied implementation is a DSA implementation, B.3.1 is required to be answered by the supplier.

If the supplied implementation is a DUA implementation, B.3.2 is required to be answered by the supplier.

Answering "No" to B.3.1.1 or B.3.2.1 indicates non-conformance to the protocol specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conformant. Such information shall be provided in B.6.5 "Other Information".

B.3.1 DSA Implementation and/or System

Item No.	Question		Status	Support	Predicate Name
1	Are all mandatory general capabilities for the DSA implemented?		m		
2	Are all mandatory First-level DSA requirements (ISO/IEC 9594-4) implemented?		c0		
3	Are minimum knowledge requirements (ISO/IEC 9594-2) implemented?		m		
4	Other Supported Reference(s)	Cross Reference	o		
		Nonspecific Subordinate Reference	o		
		Immediate Superior Reference	o		
5	Supported Security Level(s)	none	o.1		
		simple	o.1		Simple-DSA
		strong	o.1		Strong-DSA
		external	i		
6	Is asynchronous (ROSE class 2) mode of operation supported?		m		
7	Does the DSA follow the rules of extensibility as defined in 7.5 of ISO/IEC 9594-5?		m		
8	Is the alias mechanism implemented?		m		
9	Does the DSA support the directoryAccessAC application-context?		m		
10	Supported Access Controls	Simplified Access Control	o		SimpleAC-DSA
		Basic Access control	o		BasicAC-DSA
		Other	i		
11	Is the DSA capable of supporting collective attributes?		o		
12	Is the DSA capable of supporting operational attributes?		m		
13	Is the DSA capable of supporting hierarchical attributes?		o		
14	Is the DSA capable of supporting auxiliary object classes?		o		
15	Is the DSA capable of supporting the subschema for its portion of the DIT?		o		
16	Is the DSA capable of supporting SONET-specific extensions?		o		sdhUsage-DSA

o.1 The DSA shall support at least one security level.

c0: If the special configuration in item B.1.2/7 is First-level DSA, then m; else i.

B.3.2 DUA Implementation and/or System

Item No.	Question		Status	Support	Predicate Name
1	Are all mandatory general capabilities for the DUA implemented?		m		
2	Is the directoryAccessAC application-context supported?		m		
3	Supported Security Level(s)	none	o.2		
		simple	o.2		Simple-DUA
		strong	o.2		Strong-DUA
		external	i		
4	Is asynchronous (ROSE class 2) mode of operation supported?		o		
5	Supported Access Controls	Simplified Access Control	o		SimpleAC-DUA
		Basic Access Control	o		BasicAC-DUA
		Other	i		
5	Does the DUA follow the rules of extensibility as defined in 7.5 of ISO/IEC 9594-5?		m		
6	Does the DUA support SONET-specific extensions?		o		sdhUsage-DUA

o.2 The DUA shall support at least one security level.

B.4 Instruction for Completing the PICS Proforma

B.4.1 Definition of Support

A capability is said to be supported if the Implementation Under Test is able:

- To generate the corresponding operation parameters (either automatically or because the end user requires that capability explicitly).
- To interpret, handle, and – when required – make available to the end user the corresponding error or result.

A protocol element is said to be supported for a sending implementation if it is able to generate it under some circumstances (either automatically or because the end user requires relevant services explicitly).

A protocol element is said to be supported for a receiving implementation if it is correctly interpreted and handled and also, when appropriate, made available to the end user.

B.4.2 Status Column

This column indicates the level of support required for conformance to the TMN Directory Service standard.

The Status (DUA) indicates that the implementation under the IUT is a DUA and the Status (DSA) indicates that the implementation under the IUT is a DSA.

The values are as follows:

m Mandatory support is required.

- o Optional support is permitted for conformance to the standard. If implemented, it shall conform to the specifications and restrictions contained in the standard. These restrictions may affect the optionality of other items.
- c The item is conditional (support of the capability is subject to a predicate).
- c: m The item is mandatory if the predicate is true; optional otherwise.
- - The item is not applicable.
- i The item is outside the scope of this PICS.

Where this profile differs from the base standard (as specified by the 1993 DAP PICS Proforma), two characters are used in the status column, with the first character indicating the level of support required by the base standard, and the second character indicating the level of support required by this profile for the TMN Directory Service. Thus, "om" indicates the feature is optional in general but mandatory for TMN; "oc:m" indicates the feature is optional in general but mandatory under the stated predicate for TMN (e.g., for SONET only if B.3.1/16 or B.3.2/6 are the predicates).

B.4.3 Support Column

This column shall be completed by the supplier or implementor, to indicate the level of implementation of each item. The proforma has been designed such that values required are:

- Y Yes, the item has been implemented.
- N No, the item has not been implemented.
- - The item is not applicable.

In the PICS proforma tables, every leading item marked “m” shall be supported by the IUT. Subitems marked “m” shall be supported if the corresponding leading item is supported by the IUT.

B.4.4 Note Column

This column indicates the following:

- notexx Refers to Note xx.
- d(xx) A default value xx within ( ) is defined in the standard. When absent in the PDU, both sender and receiver shall interpret it as having the default value specified in the standard.

B.4.5 Predicate Column

The item number contained in the predicate column, if any, means that the status in the "Status" column applies only when the PICS states that one or more features identified by the item is supported.

B.4.6 Item Reference Numbers

Each line within the PICS proforma that requires implementation details to be entered is numbered at the left-hand edge of the line. This numbering is included as a means of uniquely identifying all possible implementation details within the PICS proforma. This referencing is used both inside the PICS proforma, and for references from other test specification documents.

The means of referencing individual responses is done by the following sequence:

- A reference to the smallest subsection enclosing the relevant item.
- A solidus character, ”/”.
- The reference number of the row in which the response appears.
- If, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labeled a, b, c, etc., from left to right, and this letter is appended to the sequence.

An example of the use of this notation would be B.6.3.1.1/2, which refers to the support for credentials in a DirectoryBind protocol data unit.

B.5 (not used)

B.6 Capabilities & options

This part of the PICS proforma identifies the supported application context, the PDUs, and operations.

Finally, the operation arguments and PDU parameters are identified.

B.6.1 Supported Application Context

The only application context supported by this PICS proforma is Directory Access application context.

B.6.2 Operations & Extensibility

B.6.2.1 Operations [11]

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	DirectoryBind	m	m				
2	DirectoryUnbind	m	m				
3	Read	o	m				
4	Compare	o	m				
5	Abandon	o	m	B.3.1/6, B.3.2/4	Note 2		
6	List	o	m				
7	Search	o	m				
8	AddEntry	o	m				
9	RemoveEntry	o	m				
10	ModifyEntry	o	m				
11	ModifyDN	o	m				

NOTE 2 – The Abandon operation can only be supported if the asynchronous mode (ROSE class 2) of operation is supported in Items B.3.1/6 or B.3.2/4 for DSA and DUA, respectively.

B.6.2.2 Extensibility

This table defines a number of extensions that are available in the 1993 edition of the Directory. The supplier of the implementation shall indicate, in the following table, which extensions for which conformance is claimed.

---

[11] For a TMN NE, the following operations, in addition to *DirectoryBind* and *DirectoryUnbind*, are considered a minimal set (though they may not all reside in a single DUA or a single system):

- *Read* - For retrieving specific parameters for a known entry (e.g., self).
- *Search* - For resolving Event Forward Discriminator.
- *AddEntry* - For participation in Automatic Registration.
- *ModifyEntry* - For MIB/DIB synchronization.

Item No.	Extension	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	subentries	o	o				
2	copyShallDo	o	o				
3	attributeSizeLimit	o	o				
4	extraAttributes	o	o				
5	modifyRightsRequest	o	o				
6	pagedResultsRequest	o	o				
7	matchValuesOnly	o	o				
8	extendedFilter	o	o				
9	targetSystem	o	o				
10	useAliasOnUpdate	o	o				
11	newSuperior	o	o				

B.6.3 Protocol Elements

B.6.3.1 Directory Bind Elements

B.6.3.1.1 Directory Bind Arguments

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	DirectoryBindArg	c: m	c: m	B.6.2.1/1			
2	credentials	c1	c1				
3	simple	c: m	c: m	Simple-DUA, Simple-DSA			
4	name	m	m				
5	validity	c: m	c: m	B.6.3.1.1/8			
6	password	o	o				
7	unprotected	o.3	o.3				
8	protected	o.3	o.3				
9	strong	c: m	c: m	Strong-DUA, Strong-DSA			
10	certification-path	o	o		Note 3		
11	bind-token SIGNED	m	m				
12	algorithm	m	m				
13	name	m	m				
14	time	m	m				
15	random	m	m				
16	externalProcedure	i	i				
17	versions	m	m		d(v1)		

c1 If [Simple-DSA or Simple-DUA or Strong-DUA or Strong-DSA], then support of this feature is m; else support is o.

o.3 The password for the DUA and DSA may be unprotected or protected as described in clause 18 of ITU-T Recommendation X.509 | ISO/IEC 9594-8.

NOTE 3 – Reference Table B.6.3.23.

B.6.3.1.2 Directory Bind Result

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	DirectoryBindResult	c: m	c: m	B.6.2.1/1			
2	credentials	c1	c1				
3	simple	c: m	c: m	Simple-DUA, Simple-DSA			
4	name	m	m				
5	validity	m	m	B.6.3.1.2/8			
6	password	m	m				
7	unprotected	o.3	o.3				
8	protected	o.3	o.3				
9	strong	c: m	c: m	Strong-DUA, Strong-DSA			
10	certification-path	o	o		Note 3		
11	bind-token SIGNED	m	m				
12	algorithm	m	m				
13	name	m	m				
14	time	m	m				
15	random	m	m				
16	externalProcedure	o	o				
17	versions	m	m		d(v1)		

c1 If [Simple-DUA or Simple-DSA or Strong-DUA or Strong-DSA], then support of this feature is m; else support is o.

o.3 The password for the DUA and DSA may be unprotected or protected as described in clause 18 of ITU-T Recommendation X.509 | ISO/IEC 9594-8.

Note 3 – Reference Table B.6.3.23.

B.6.3.1.3 Directory Bind Error

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	DirectoryBindError	c: m	c: m	B.6.2.1/1			
2	versions	m	m		d(v1)		
3	ServiceError	m	m				
4	SecurityError	m	m				

B.6.3.2 Directory Unbind Elements

DirectoryUnbind has no arguments

B.6.3.3 Read Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	read	c: m	c: m	B.6.2.1/3			
2	ReadArgument	m	m				
3	SIGNED ReadArgument	c2	c3				
4	object	m	m				
5	selection	m	m		d({})		
6	modifyRightsRequest	c: m	c: m	B.6.2.2/5	d(false)		
7	CommonArguments	o	m				
8	ReadResult	m	m				
9	SIGNED ReadResult	c2	c3				
10	entry	m	m				
11	modifyRights	c: m	c: m	B.6.3.3/6			
12	CommonResults	m	m				

c2 : if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.4 Compare Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	compare	c: m	c: m	B.6.2.1/4			
2	CompareArgument	m	m				
3	SIGNED CompareArgument	c2	c3				
4	object	m	m				
5	purported	m	m				
6	CommonArguments	o	m				
7	CompareResult	m	m				
8	SIGNED CompareResult	c2	c3				
9	name	m	m				
10	matched	m	m				
11	fromEntry	m	m		d(true)		
12	matchedSubtype	m	m				
13	CommonResults	m	m				

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.5 Abandon Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	abandon	c: m	c: m	B.6.2.1/5			
2	AbandonArgument	m	m				
3	invokeID	m	m				
4	AbandonResult	m	m				

B.6.3.6 List Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	list	c: m	c: m	B.6.2.1/6			
2	ListArgument	m	m				
3	SIGNED ListArgument	c2	c3				
4	object	m	m				
5	pagedResults	c: m	c: m	B.6.2.2/6			
6	CommonArguments	o	m				
7	ListResult	m	m				
8	SIGNED ListResult	c2	c3				
9	listInfo	m	m				
10	name	m	m				
11	subordinates	m	m				
12	RDN	m	m				
13	aliasEntry	m	m		d(false)		
14	fromEntry	m	m		d(true)		
15	PartialOutcomeQualifier	o	m				
16	limitProblem	o	m				
17	unexplored	c4	c4				
18	unavailableCriticalExt	m	m		d(false)		
19	unknownErrors	m	m				
20	queryReference	c: m	c: m	B.6.3.6/5			
21	CommonResults	m	m				
22	uncorrelatedListInfo	o	c4				

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

c4: If item B.1.2/7 indicates that the DSA is a "Cooperating" DSA or that the DUA connects to "Cooperating" DSAs, then support of this feature is m; else support is o.

B.6.3.7 Search Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	search	c: m	c: m	B.6.2.1/7			
2	SearchArgument	m	m				
3	SIGNED SearchArgument	c2	c3				
4	baseObject	m	m				
5	subset	m	m		d(0)		
6	filter	o	m		d(and{})		
7	searchAlias	m	m		d(true)		
8	selection	o	m		d({})		
9	pagedResults	c: m	c: m	B.6.2.2/6			
10	matchedValuesOnly	c: m	c: m	B.6.2.2/7	d(false)		
11	extendedFilter	c: m	c: m	B.6.2.2/8			
12	CommonArguments	o	m				
13	SearchResult	m	m				
14	SIGNED SearchResult	c2	c3				
15	searchInfo	m	m				
16	name	o	m				
17	entries	m	m				
18	PartialOutcomeQualifier	o	m				
19	limitProblem	o	m				
20	unexplored	c4	c4				
21	unavailableCriticalExt	m	m		d(false)		
22	unknownErrors	m	m				
23	queryReference	c: m	c: m	B.6.3.7/9			
24	CommonResults	m	m				
25	uncorrelatedSearchInfo	o	c4				

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

c4: If item B.1.2/7 indicates that the DSA is a "Cooperating" DSA or that the DUA connects to "Cooperating" DSAs, then support of this feature is m; else support is o.

B.6.3.8 Add Entry Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	addEntry	c: m	c: m	B.6.2.1/8			
2	AddEntryArgument	m	m				
3	SIGNED AddEntryArgument	c2	c3				
4	object	m	m				
5	entry	m	m				
6	targetSystem	c: m	c: m	B.6.2.2/9			
7	CommonArguments	o	m				
8	AddEntryResult	m	m		= NULL		

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.9 Remove Entry Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	removeEntry	c: m	c: m	B.6.2.1/9			
2	RemoveEntryArgument	m	m				
3	SIGNED RemoveEntryArgument	c2	c3				
4	object	m	m				
5	CommonArguments	o	m				
6	RemoveEntryResult	m	m		= NULL		

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.10 Modify Entry Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	modifyEntry	c: m	c: m	B.6.2.1/10			
2	ModifyEntryArgument	m	m				
3	SIGNED ModifyEntryArgument	c2	c3				
4	object	m	m				
5	changes	m	m				
6	addAttribute	m	m				
7	removeAttribute	m	m				
8	addValues	m	m				
9	removeValues	m	m				
10	CommonArguments	o	m				
11	ModifyEntryResult	m	m		= NULL		

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.11 ModifyDN Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	modifyDN	c: m	c: m	B.6.2.1/11	Note 4		
2	ModifyDNArgument	m	m				
3	SIGNED ModifyDNArgument	c2	c3				
4	object	m	m				
5	newRDN	m	m				
6	deleteOldRDN	m	m		d(false)		
7	newSuperior	c: m	c: m	B.6.2.2/11			
8	CommonArguments	o	m				
9	ModifyDNResult	m	m		= NULL		

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

NOTE 4 – 1988-edition systems may use the operation only to change the Relative Distinguished Name of a leaf entry.

B.6.3.12 Errors & Parameters

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	Abandoned	c: m	m	B.6.2.1/5			
2	AbandonFailed	c: m	m	B.6.2.1/5			
3	problem	m	m				
4	operation	m	m				
5	AttributeError	m	m				
6	object	m	m				
7	problems	m	m				
8	problem	m	m				
9	type	m	m				
10	value	m	m				
11	NameError	m	m				
12	problem	m	m				
13	matched	m	m				
14	Referral	m	m				
15	candidate	m	m				
16	SecurityError	m	m				
17	problem	m	m				
18	ServiceError	m	m				
19	problem	m	m				
20	UpdateError	c5	m				
21	problem	m	m				

c5: If at least one of the following operations are supported, then support of this feature is m: AddEntry, RemoveEntry, ModifyEntry, ModifyDN; else support is o.

B.6.3.13 Common Arguments Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	serviceControls	o	m		d({})		
2	securityParameters	o	c: m	Strong-DSA	d({})		
3	requestor	o	o		Note 5		
4	operationProgress	o	m		d(notStarted)		
5	nameResolutionPhase	o	m				
6	nextRDNToBeResolved	o	m				
7	aliasedRDNs	o	o		Note 6		
8	criticalExtensions	m	m				
9	referenceType	o	m				
10	entryOnly	o	m		d(true)		
11	exclusions	o	m				
12	nameResolveOnMaster	o	m		d(false)		

NOTE 5 – This parameter may be ignored unless the request is signed.

NOTE 6 – This parameter is provided for compatibility with the 1988 edition of the Directory. DUAs (and DSAs) implemented in accordance with later editions shall always omit this parameter.

B.6.3.14 Common Results Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	securityParameters	c2	c3				
2	performer	m	m				
3	aliasDereferenced	m	m		d(false)		

c2: if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

B.6.3.15 Service Controls

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	ServiceControls	o	m				
2	options	o	m		d({})		
3	priority	o	m		d(medium)		
4	timeLimit	o	m				
5	sizeLimit	o	m				
6	scopeOfReferral	o	c4				
7	attributeSizeLimit	c: m	c: m	B.6.2.2/3			

c4: If item B.1.2/7 indicates that the DSA is a "Cooperating" DSA or that the DUA connects to "Cooperating" DSAs, then support of this feature is m; else support is o.

B.6.3.16 Entry Information Selection

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	EntryInformationSelection	o	m				
2	allUserAttributes	o	m				
3	select	o	m				
4	infoTypes	o	m		d(attr types & values)		
5	allOperationalAttributes	o	m				
6	select	o	m				

B.6.3.17 Entry Information

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	EntryInformation	m	m				
2	name	m	m				
3	fromEntry	m	m		d(true)		
4	information	m	m				
5	AttributeType	m	m				
6	Attribute	m	m				
7	incompleteEntry	m	m		d(false)		

B.6.3.18 Filter Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	item	o	m				
2	and	o	m				
3	or	o	m				
4	not	o	m				

B.6.3.19 Filter Item Elements

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	equality	o	m				
2	substrings	o	m				
3	type	o	m				
4	strings	o	m				
5	initial	o	m				
6	any	o	m				
7	final	o	m				
8	greaterOrEqual	o	m				
9	lessOrEqual	o	m				
10	present	o	m				
11	approximateMatch	o	m				
12	extensibleMatch	o	m				

B.6.3.20 Paged Results

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	PagedResultsRequest	c: m	c: m	B.6.2.2/6			
2	newRequest	o	o				
3	pageSize	o	o				
4	sortkeys	o	o				
5	reverse	o	o				
6	unmerged	o	o				
7	queryReference	o	o				

B.6.3.21 Continuation Reference

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	continuationReference	c4	c4				
2	targetObject	o	m				
3	aliasedRDNs	o	m				
4	operationProgress	o	m				
5	nameResolutionPhase	o	m				
6	nextRDNToBeResolved	o	m				
7	rdnsResolved	o	m				
8	referenceType	o	m				
9	accessPoints	o	m				
10	MasterOrShadowAccessPoint	o	m				
11	category	o	m				
12	AccessPoint	o	m				
13	ae-title	m	m				
14	address	m	m				
15	pSelector	m	m				
16	sSelector	m	m				
17	tSelector	m	m				
18	nSelector	m	m				
19	protocolInformation	o	o				
20	entryOnly	m	m		d(false)		
21	exclusions	o	m				
22	returnToDUA	m	m		d(false)		
23	nameResolveOnMaster	m	m		d(false)		

c4: If item B.1.2/7 indicates that the DSA is a "Cooperating" DSA or that the DUA connects to "Cooperating" DSAs, then support of this feature is m; else support is o.

B.6.3.22 Security Parameters

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	certification-path	o	c: m	Strong-DSA	Note 3		
2	name	o	m				
3	time	o	m				
4	random	o	m				
5	target	o	o				

NOTE 3 – Reference Table B.6.3.23.

B.6.3.23 Certification Path

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	CertificationPath	c2	c3				
2	Certificate	m	m				
3	version	m	m		d(v1)		
4	serialNumber	m	m				
5	signature	m	m				
6	issuer	m	m				
7	validity	m	m				
8	subject	m	m				
9	subjectPublicKey	m	m				
10	issuerUniqueIdentifier	o	o		Note 7		
11	subjectUniqueIdentifier	o	o		Note 7		
12	CertificatePair	o	o				
13	forward	o	o				
14	reverse	o	o				

c2 : if [Strong-DUA], then m; else o.

c3: if [Strong-DSA], then m; else o.

NOTE 7 – If present, version shall be v2.

B.6.3.24 Access Control

B.6.3.24.1 Access Control Information

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	ACIItem	c7	c8				
2	identificationTag	m	m				
3	precedence	m	m				
4	authenticationLevel	m	m				
5	basicLevels	m	m				
6	level	m	m				
7	localQualifier	o	o				
8	other	i	i				
9	itemOrUserFirst	m	m				
10	itemFirst	m	m				
11	protectedItems	m	m				
12	itemPermissions	m	m				
13	precedence	o	o				
14	userClasses	o	o				
15	grantsAndDenials	m	m				
16	userFirst	m	m				
17	userClasses	o	o				
18	userPermissions	m	m				
19	precedence	o	o				
20	protectItems	m	m				
21	grantsAndDenials	m	m				

c7 : If [SimpleAC-DUA or BasicAC-DUA], then m; else n/a.

c8: if [SimpleAC-DSA or BasicAC-DSA], then m; else n/a

B.6.3.24.2 Protected Items

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	ProtectedItems	m	m				
2	entry	o	o				
3	allUserAttributesTypes	o	o				
4	attributeType	o	o				
5	allAttributeValues	o	o				
6	allUserAttributeTypes&Values	o	o				
7	attributeValue	o	o				
8	selfValue	o	o				

B.6.3.24.3 User Classes

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	UserClasses	m	m				
2	allUsers	o	o				
3	thisEntry	o	o				
4	name	o	o				
5	userGroup	o	o				
6	subtree	o	o				

B.6.3.25 Subtrees

Item No.	Protocol Element	Status (DUA)	Status (DSA)	Predicate	Note	Support (DUA)	Support (DSA)
1	SubtreeSpecification	c: m	c: m	B.6.2.2/1			
2	base	m	m				
3	LocalName	m	m		d({})		
4	ChopSpecification	m	m				
5	specificExclusions	m	m				
6	chopBefore	m	m				
7	chopAfter	o	o				
8	minimum	m	m		d(0)		
9	maximum	o	o				
10	specificationFilter	o	o				
11	item	m	m				
12	and	m	m				
13	or	m	m				
14	not	m	m				

B.6.4 Directory Schema

This is covered in Annex A of the Directory Service for TMN standard.

B.6.5 Other Information

The following table can be used to provide any other relevant information:

Index	Other information

Annex C
(normative)

C PICS for Directory System Protocol (DSP)

For the purposes of participating in distributed operations, a conformant DSA shall support the DirectorySystemAC application context according to the base standard ITU-T Recommendation X.518.[12]

[12] The PICS Proforma for the base standard is currently being drafted.

Although there are differences in DSP between the 1993 and 1988 versions of the standard, the following profiles for the 1988 version of the protocol are largely applicable:

- ISO/IEC PDISP 10615-3, *Information Technology – International Standardized Profiles ADInn – OSI Directory – Part 4: ADI22 – DSA Initiator Role.*
- ISO/IEC PDISP 10615-4, *Information Technology – International Standardized Profiles ADInn – OSI Directory – Part 3: ADI21 – DSA Responder Role.*

Annex D
(normative)

D PICS for Directory Information Shadowing Protocol (DISP)

For the purposes of participating in shadowing, a conformant DSA shall support the shadowSupplierInitiatedAC application context according to the ITU-T Recommendation X.525.[13]

[13] The PICS Performa and ISPs for the base standard are currently being drafted.

Annex E
(normative)

E PICS for Directory Operational Binding Management Protocol (DOP)

The PICS for the Directory Operational Binding Management Protocol are deferred until future expansions of this standard.

Annex F
(normative)

F PICS for Registration Request Protocol (RRP)

The implementation of a Registration Manager or a Registration Agent is optional. Where these are implemented, they shall support the TMNRegistrationRequestAC application context according to the following profile.

F.1 Identification of the Implementation

F.1.1 Identification of PICS

Item No.	Question	Response
1	Date of Statement (DD/MM/YY)	
2	PICS Serial Number	
3	System Conformance Statement Cross Reference	

F.1.2 Identification of the Implementation and/or System

Item No.	Question	Response
1	Implementation Name	
2	Version Number	
3	Machine Name	
4	Machine Version Number	
5	Operating System Name	
6	Operating System Version No.	
7	Special Configuration	Note 1
8	Other information	

NOTE 1 – Please enter one or more of the following:

- DUA for connection to centralized DSAs.
- DUA for connection to cooperating DSAs.
- Centralized DSAs.
- Cooperating DSAs.
- First-level DSAs.

F.1.3 Identification of the System Supplier and/or Test Laboratory Client

Item No.	Question	Response
1	Organization Name	
2	Contact Name(s)	
3	Address	
4	Telephone Number	
5	Telex Number	
6	Fax Number	
7	E-Mail Address	
8	Other information	

F.2 Identification of the Protocol

Item No.	Question	Response
1	Title, Reference, No., publication date of the protocol standard	
2	Protocol Version Number	
3	Implemented Addenda	
4	Implemented Defect Reports (Reference No.)	

F.3 Global Statement of Conformance

If the supplied implementation is an RM implementation, F.3.1 is required to be answered by the supplier.

If the supplied implementation is an RA implementation, F.3.2 is required to be answered by the supplier.

Answering "No" to F.3.1.1 or F.3.2.1 indicates nonconformance to the protocol specification. Nonsupported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is nonconformant. Such information shall be provided in F.6.5 "Other Information".

F.3.1 RM Implementation and/or System

Item No.	Question	Status	Support
1	Are all mandatory general capabilities for the RM implemented?	m	
2	Is the TMNRegistrationRequestAC application context supported?	m	

F.3.2 RA Implementation and/or System

Item No.	Question	Status	Support
1	Are all mandatory general capabilities for the RA implemented?	m	
2	Is the TMNRegistrationRequestAC application context supported?	m	

F.4 Instruction for Completing the PICS Proforma

F.4.1 Definition of Support

A *capability* is said to be supported if the Implementation Under Test is able:

- To generate the corresponding operation parameters (either automatically or because the invoker requires that capability explicitly).
- To interpret, handle, and – when required – make available to the invoker the corresponding error or result.

A *protocol element* is said to be supported *for a sending implementation* if the IUT is able to generate it under some circumstances (either automatically or because the invoker requires relevant services explicitly).

A *protocol element* is said to be supported *for a receiving implementation* if it is correctly interpreted and handled and also – when appropriate – made available to the invoker.

F.4.2 Status Column

This column indicates the level of support required for conformance to the TMN Directory Service standard.

The Status (RA) indicates that the implementation under the IUT is an RA and the Status (RM) indicates that the implementation under the IUT is an RM.

The values are as follows:

m Mandatory support is required.

o Optional support is permitted for conformance to the standard. If implemented, it shall conform to the specifications and restrictions contained in the standard. These restrictions may affect the optionality of other items.

c The item is conditional (support of the capability is subject to a predicate).

c: m The item is mandatory if the predicate is true, optional otherwise.

- The item is not applicable.

i The item is outside the scope of this PICS.

F.4.3 Support Column

This column shall be completed by the supplier or implementor, to indicate the level of implementation of each item. The proforma has designed such that values required are:

Y Yes, the item has been implemented.

N No, the item has not been implemented.

- The item is not applicable.

In the PICS proforma tables, every leading item marked “m” shall be supported by the IUT. Subitems marked “m” shall be supported if the corresponding leading item is supported by the IUT.

F.4.4 Note Column

This column indicates the following:

notexx Refers to Note xx.

d(xx) A default value xx within ( ) is defined in the standard. When absent in the PDU, both sender and receiver shall interpret it as having the default value specified in the standard.

F.4.5 Predicate Column

The item number contained in the predicate column, if any, means that the status in the "Status" column applies only when the PICS states that one or more features identified by the item is supported.

F.4.6 Item Reference Numbers

Each line within the PICS proforma that requires implementation details to be entered is numbered at the left-hand edge of the line. This numbering is included as a means of uniquely identifying all possible implementation details within the PICS proforma. This referencing is used both inside the PICS proforma, and for references from other test specification documents.

The means of referencing individual responses is done by the following sequence:

- A reference to the smallest subsection enclosing the relevant item.
- A solidus character, ”/”.
- The reference number of the row in which the response appears.
- If, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labeled a, b, c, etc., from left to right, and this letter is appended to the sequence.

An example of the use of this notation would be B.6.3.1.1/2, which refers to the support for credentials in a DirectoryBind protocol data unit.

F.5 (not used)

F.6 Capabilities & Options

This part of the PICS proforma identifies the supported application context, the PDUs, and operations.

Finally, the operation arguments and PDU parameters are identified.

F.6.1 Supported Application Context

The only application context supported by this PICS proforma is TMNRegistrationRequestAC context.

F.6.2 Operations

Item No.	Protocol Element	Status (RA)	Status (RM)	Predicate	Note	Support (RA)	Support (RM)
1	RegistrationBind	m	m				
2	RegistrationUnbind	m	m				
3	RegistrationRequestOperation	m	m				

F.6.3 Protocol Elements

F.6.3.1 Registration Bind Elements

F.6.3.1.1 Registration Bind Arguments

Item No.	Protocol Element	Status (RA)	Status (RM)	Predicate	Note	Support (RA)	Support (RM)
1	RegistrationBindArgument	m	m				
2	versions	m	m		d (v1)		

F.6.3.1.2 Registration Bind Result

Item No.	Protocol Element	Status (RA)	Status (RM)	Predicate	Note	Support (RA)	Support (RM)
1	RegistrationBindResult	m	m				
2	versions	m	m		d(v1)		

F.6.3.1.3 Registration Bind Error

Item No.	Protocol Element	Status (RA)	Status (RM)	Predicate	Note	Support (RA)	Support (RM)
1	RegistrationBindError	m	m				

F.6.3.2 Registration Unbind Elements

RegistrationUnbind has no arguments.

F.6.3.3 Registration Request Operation Elements

Item No.	Protocol Element	Status (RA)	Status (RM)	Predicate	Note	Support (RA)	Support (RM)
1	RegistrationRequestArgument	m	m				
2	primaryDsaAddress	m	m				
3	alternateDsaAddress	m	m				
4	namePrefix	m	m				

F.6.4 Other Information

The following table can be used to provide any other relevant information:

Index	Other information

Annex G
(normative)

G ASN.1 for Definitions

G.1 Useful Definitions

This module presents an ASN.1 definition of the upper reaches of the ASN.1 object identifier tree used to register the information objects in the other modules.

```
TMNDirUsefulDefinitions {iso org ansi(840) t1245(10041) tmnDirModule(1)
tmnDirUsefulDefinitions(1)}

DEFINITIONS ::=
BEGIN

-- EXPORTS All

-- object identifiers for major arcs

tmnDirModule      OBJECT IDENTIFIER      ::= {iso org ansi(840) t1245(10041) tmnDirModule(1)}
tmnDir            OBJECT IDENTIFIER      ::= {iso org ansi(840) t1245(10041) tmnDirectoryObjects(2)}
tmnRRS            OBJECT IDENTIFIER      ::= {iso org ansi(840) t1245(10041)
tmnRegistrationRequestService(3)}

-- categories of Directory information object

tmnDirObjectClass          OBJECT IDENTIFIER          ::= { tmnDir 6}
tmnDirAttributeSet         OBJECT IDENTIFIER          ::= { tmnDir 7}
tmnDirAttribute            OBJECT IDENTIFIER          ::= { tmnDir 4}
tmnDirAttributeSyntax      OBJECT IDENTIFIER          ::= { tmnDir 5}
tmnDirMatchingRule         OBJECT IDENTIFIER          ::= { tmnDir 13}
tmnDirNameForm             OBJECT IDENTIFIER          ::= { tmnDir 15}

-- modules defined in this standard

tmnDirUsefulDefinitions                OBJECT IDENTIFIER      ::= {tmnDirModule 1}
tmnDirAttributesAndMatchingRules       OBJECT IDENTIFIER      ::= {tmnDirModule 2}
tmnDirObjectClasses                    OBJECT IDENTIFIER      ::= {tmnDirModule 3}
tmnDirDIT                              OBJECT IDENTIFIER      ::= {tmnDirModule 4}
tmnDirRegistrationRequestService       OBJECT IDENTIFIER      ::= {tmnDirModule 5}

-- specific object identifiers

id-at-proprietaryAddress               OBJECT IDENTIFIER      ::= {tmnDirAttribute 1}
id-at-nodeIdInfo                       OBJECT IDENTIFIER      ::= {tmnDirAttribute 2}
id-at-neType                           OBJECT IDENTIFIER      ::= {tmnDirAttribute 3}
id-at-vendorName                       OBJECT IDENTIFIER      ::= {tmnDirAttribute 4}
id-at-entityAddress                    OBJECT IDENTIFIER      ::= {tmnDirAttribute 5}
id-at-osType                           OBJECT IDENTIFIER      ::= {tmnDirAttribute 6}
id-at-otherSupportedFunctionalBlocks   OBJECT IDENTIFIER      ::= {tmnDirAttribute 7}
```

```
id-mr-nodeIdInfoMatch          OBJECT IDENTIFIER          ::= {tmnDirMatchingRule 1}

id-oc-tmnNE                    OBJECT IDENTIFIER          ::= {tmnDirObjectClass 1}
id-oc-sdhNEEntry               OBJECT IDENTIFIER          ::= {tmnDirObjectClass 2}
id-oc-apaeAlias                OBJECT IDENTIFIER          ::= {tmnDirObjectClass 3}
id-oc-tmnOS                    OBJECT IDENTIFIER          ::= {tmnDirObjectClass 4}
id-oc-network                  OBJECT IDENTIFIER          ::= {tmnDirObjectClass 5}
id-oc-tmnNEComplex             OBJECT IDENTIFIER          ::= {tmnDirObjectClass 6}


id-nf-tmnNENameForm            OBJECT IDENTIFIER          ::= {tmnDirNameForm 1}
id-nf-apaeAliasNameForm        OBJECT IDENTIFIER          ::= {tmnDirNameForm 2}
id-nf-networkNameForm          OBJECT IDENTIFIER          ::= {tmnDirNameForm 3}
id-nf-tmnNEComplexNameForm     OBJECT IDENTIFIER          ::= {tmnDirNameForm 4}
id-nf-tmnOSNameForm            OBJECT IDENTIFIER          ::= {tmnDirNameForm 5}

id-rrp-registrationAC          OBJECT IDENTIFIER          ::= {tmnRRS 1}
id-rrp-registrationAS          OBJECT IDENTIFIER          ::= {tmnRRS 2}
id-rrp-registrationRequest     OBJECT IDENTIFIER          ::= {tmnRRS 3}
id-rrp-registrationError       OBJECT IDENTIFIER          ::= {tmnRRS 4}
id-rrp-rosObject-rm            OBJECT IDENTIFIER          ::= {tmnRRS 5}
id-rrp-rosObject-registration  OBJECT IDENTIFIER          ::= {tmnRRS 6}
id-rrp-rosObject-rrpRA         OBJECT IDENTIFIER          ::= {tmnRRS 7}
id-rrp-contract-rrp            OBJECT IDENTIFIER          ::= {tmnRRS 8}
id-rrp-package-rrpConnection   OBJECT IDENTIFIER          ::= {tmnRRS 9}

END
```

G.2 Attributes & Matching Rules

This module presents an ASN.1 definition of the attributes, attribute sets, and matching rules defined in the body of this document. Referenced definitions are imported from other modules where appropriate.

```
TMNDirAttributesAndMatchingRules {iso org ansi(840) t1245(10041) tmnDirModule(1)
tmnDirAttributesAndMatchingRules(2)}

DEFINITIONS ::=
BEGIN
IMPORTS
   id-at-proprietaryAddress, id-at-nodeIdInfo, id-at-neType, id-at-vendorName, id-at-entityAddress, id-at-
   osType, id-at-otherSupportedFunctionalBlocks
   FROM TMNDirUSefulDefinitions {iso org ansi(840) t1245(10041) tmnDirModule(1)
   tmnDirUsefulDefinitions(1)},

   informationFramework, selectedAttributeTypes
   FROM UsefulDefinitions {joint-iso-ccitt(2) ds(5) modules(1) usefulDefinitions(0) 2},

   ATTRIBUTE, MATCHING RULE, AttributeType, objectIdentifierMatch, Name
   FROM InformationFramework informationFramework,

   caseIgnoreMatch, caseIgnoreSubstringsMatch, NumericStringMatch, DirectoryString, name
```

```
   FROM SelectedAttributeTypes selectedAttributeTypes

-- EXPORTS All

-- attributes

        proprietaryAddress ATTRIBUTE  ::= {
                WITH SYNTAX                     PrintableString {ub-proprietary-Address}
                EQUALITY MATCHING RULE          caseIgnoreMatch
                SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
                SINGLE VALUE                TRUE
                ID                              id-at-proprietary-Address}

        nodeIdInfo ATTRIBUTE  ::=           {
                WITH SYNTAX                     NodeNumberUID
                EQUALITY MATCHING RULE          nodeIdInfoMatch
                ID                              id-at-nodeIdInfo }

        neType ATTRIBUTE      ::=  {
                WITH SYNTAX                 OBJECT IDENTIFIER
                EQUALITY MATCHING RULE      objectIdentifierMatch
                ID                          id-at-neType}

        vendorName  ATTRIBUTE       ::= {
                SUBTYPE OF          name
                WITH SYNTAX         DirectoryString {ub-vendor-name}
                ID                  id-at-vendorName}

        entityAddress ATTRIBUTE     ::= {
                WITH SYNTAX                     OCTETSTRING (SIZE (0..20))
                EQUALITY MATCHING RULE          caseIgnoreMatch
                SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
                ID                              id-at-entityAddress}

        osType ATTRIBUTE ::= {
                WITH SYNTAX                 OBJECT IDENTIFIER
                EQUALITY MATCHING RULE      objectIdentifierMatch
                ID                              id-at-osType}
```

```
otherSupportedFunctionalBlocks ATTRIBUTE ::= {
        WITH SYNTAX                     OBJECT IDENTIFIER
        EQUALITY MATCHING RULE          objectIdentifierMatch
        ID                              id-at-otherSupportedFunctionalBlocks}

-- attribute syntaxes

NodeNumberUID           ::=     SET {
        nodeId                  [0]     Integer (0..ub-nodeId) OPTIONAL,
        ringId                  [1]     DirectoryString (SIZE (1..ub-ringId)) OPTIONAL }

-- matching rules

nodeIdInfoMatch MATCHING-RULE       ::=     {
        SYNTAX                  NodeNumberUID
        ID                      id-mr-nodeIdInfoMatch }

-- upper bounds

ub-proprietary-Address          INTEGER         ::= 256
ub-nodeId                       INTEGER         ::= 15
ub-ringId                       INTEGER         ::= 16
ub-vendor-name          INTEGER         ::= 128


END
```

G.3 Object Classes

This module presents an ASN.1 definition of the object classes and alias object classes defined in the body of this document. Referenced definitions are imported from other modules where appropriate.

```
TMNDirObjectClasses {iso org ansi(840) t1245(10041) tmnDirModule(1) tmnDirObjectClasses(3)}

DEFINITIONS ::=
BEGIN
IMPORTS
   tmnDirAttributesAndMatchingRules,
   id-oc-tmnNE, id-oc-sdhNEEntry, id-oc-apaeAlias, id-oc-tmnOS, id-oc-network, id-oc-tmnNEComplex
   FROM TMNDirUSefulDefinitions {iso org ansi(840) t1245(10041) tmnDirModule(1)
   tmnDirUsefulDefinitions(1)},

   proprietaryAddress, nodeIdInfo, neType, vendorName, entityAddress, osType,
   otherSupportedFunctionalBlocks
```

```
    FROM TMNDirAttributesAndMatchingRules tmnDirAttributesAndMatchingRules,

    informationFramework, selectedAttributeTypes, selectedObjectClasses,
    FROM UsefulDefinitions {joint-iso-ccitt(2) ds(5) modules(1) usefulDefinitions(0) 2},

    commonName, localityName
    FROM SelectedAttributeTypes selectedAttributeTypes,

    OBJECT CLASS, top, alias
    FROM InformationFramework informationFramework

    managedElementId, managedElementComplexId, networkId
    FROM M.3100 {ccitt(0) recommendation m(13) gnm(3100) informationModel(0)}

-- EXPORTS All

-- object classes

        tmnNE   OBJECT-CLASS ::= {
                SUBCLASS of             {top}
                MUST CONTAIN { commonName |
                                        managedElementId |
                                        entityAddress }
                MAY CONTAIN             { proprietaryAddress |
                                        vendorName |
                                        localityName |
                                        neType }
                ID                      id-oc-tmnNE}

        sdhNEEntry      OBJECT-CLASS ::= {
                SUBCLASS of             {top}
                KIND                    auxiliary
                MAY CONTAIN             { nodeIdInfo }
                ID                      id-oc-sdhNEEntry}

        apaeAlias       OBJECT CLASS ::= {
                        SUBCLASS OF     {alias}
                        MUST CONTAIN { commonName } -- used to provide an RDN
                                                for the alias entry itself --
                        ID                      id-oc-apaeAlias}

        tmnOS   OBJECT-CLASS ::= {
                SUBCLASS of             {top}
```

```
        MUST CONTAIN {commonName | entityAddress }
        MAY CONTAIN          { proprietaryAddress |
                               vendorName | localityName | osType |
                               otherSupportedFunctionalBlocks}
        ID                   id-oc-tmnOS}

directoryNetwork OBJECT-CLASS ::= {
        SUBCLASS OF   {top}
        MUST CONTAIN {commonName | networkId | entityAddress}
        ID                   id-oc-network}

tmnNEComplex OBJECT-CLASS ::= {
        SUBCLASS OF   {top}
        MUST CONTAIN {commonName | managedElementComplexId | entityAddress}
        ID                   id-oc-tmnNEComplex}

END
```

G.4 DIT Structure

This module presents an ASN.1 definition of the schema described in the body of this document. It also illustrates how the overall schema can be constructed using attributes and object classes defined in a number of different modules, and how this schema may be combined with schema defined for other reasons by ITU-T and ISO.

```
TMNDirDIT {iso org ansi(840) t1245(10041) tmnDirModule(1) tmnDirDIT(4)}

DEFINITIONS ::=
BEGIN
IMPORTS
   tmnDirObjectClasses, tmnDirAttributesAndMatchingRules,
   id-nf-tmnNENameForm, id-nf-apaeAliasNameForm,
   id-nf-tmnOSNameForm, id-nf-networkNameForm, id-nf-tmnNEComplexNameForm
   FROM TMNDirUSefulDefinitions {iso org ansi(840) t1245(10041) tmnDirModule(1)
   tmnDirUsefulDefinitions(1)},
```

```
    proprietaryAddress, nodeIdInfo, neType, vendorName, entityAddress, osType,
        otherSupportedFunctionalBlocks
    FROM TMNDirAttributesAndMatchingRules tmnDirAttributesAndMatchingRules,

    tmnNE, sdhNEEntry, apaeAlias, tmnOS, directoryNetwork, tmnNEComplex
    FROM TMNDirObjectClasses tmnDirObjectClasses,

    informationFramework, selectedObjectClasses,
    FROM UsefulDefinitions {joint-iso-ccitt(2) ds(5) modules(1) usefulDefinitions(0) 2},

    country, locality, organization, organizationalUnit, applicationProcess, applicationEntity, dSA,
    groupOfNames, countryNameForm, locNameForm, sOPNameForm, orgNameForm, orgUnitNameForm,
    orgPersonNameForm, orgRoleNameForm, applProcessNameForm, applEntityNameForm, dSANameForm,
    gonNameForm
    FROM SelectedObjectClasses selectedObjectClasses,

    NAME-FORM, DITStructureRule, STRUCTURE-RULE, CONTENT-RULE, DITContentRule
    FROM InformationFramework informationFramework

-- EXPORTS All

-- name forms

        tmnNENameForm       NAME-FORM ::= {
                                    NAMES                   tmnNE
                                    WITH ATTRIBUTES         {commonName}
                                    ID                      id-nf-tmnNENameForm}

        apaeAliasNameForm           NAME-FORM ::= {
                                    NAMES                   apaeAlias
                                    WITH ATTRIBUTES         {commonName}
                                    ID                      id-nf-
        apaeAliasNameForm}

        tmnOSNameForm NAME-FORM ::= {
                        NAMES               tmnOS
                        WITH ATTRIBUTES     {commonName}
                        ID                  id-nf-tmnOSNameForm}

        networkNameForm NAME-FORM ::= {
                                    NAMES               directoryNetwork
                                    WITH ATTRIBUTES     {commonName}
                                    ID                  id-nf-directoryNameForm}
```

```
tmnNEComplexNameForm NAME-FORM ::= {
                                        NAMES                   tmnNEComplex
                                        WITH ATTRIBUTES         {commonName}
                                        ID                      id-nf-
tmnNEComplexNameForm}
```

-- content rules

```
applicationEntityRule CONTENT-RULE  ::=         {
        STRUCTURAL OBJECT CLASS     applicationEntity
        MUST CONTAIN                { supportedApplicationContext }}

networkElementRule CONTENT-RULE  ::={
        STRUCTURAL OBJECT CLASS     tmnNE
        AUXILIARY OBJECT CLASSES    { sdhNEEntry }}
```

-- structure rules

```
sr1     STRUCTURE-RULE      ::= {
        NAME FORM           countryNameForm
        ID                  1}

sr2     STRUCTURE-RULE      ::= {
        NAME FORM           orgNameForm
        ID                  2}

sr3     STRUCTURE-RULE      ::= {
        NAME FORM           orgNameForm
        SUPERIOR RULES      {sr1}
        ID                  3}

sr4     STRUCTURE-RULE      ::= {
        NAME FORM           orgNameForm
        SUPERIOR RULES      {sr5, sr6, sr7, sr35}
        ID                  4}

sr5     STRUCTURE-RULE      ::= {
        NAME FORM           locNameForm
```

	ID	5}
sr6	STRUCTURE-RULE	::= {
	NAME FORM	locNameForm
	SUPERIOR RULES	{sr1}
	ID	6}
sr7	STRUCTURE-RULE	::= {
	NAME FORM	locNameForm
	SUPERIOR RULES	{sr5, sr6, sr7, sr35}
	ID	7}
sr8	STRUCTURE-RULE	::= {
	NAME FORM	orgUnitNameForm
	SUPERIOR RULES	{sr2, sr3, sr4}
	ID	8}
sr9	STRUCTURE-RULE	::= {
	NAME FORM	orgUnitNameForm
	SUPERIOR RULES	{sr5, sr6, sr7, sr35}
	ID	9}
sr10	STRUCTURE-RULE	::= {
	NAME FORM	orgUnitNameForm
	SUPERIOR RULES	{sr8, sr9, sr10}
	ID	10}

```
sr11    STRUCTURE-RULE      ::= {
        NAME FORM           gONNameForm
        SUPERIOR RULES      {sr5, sr6, sr7, sr35}
        ID                  11}

sr12    STRUCTURE-RULE      ::= {
        NAME FORM           gONNameForm
        SUPERIOR RULES      {sr2, sr3, sr4}
        ID                  12}

sr13    STRUCTURE-RULE      ::= {
        NAME FORM           gONNameForm
        SUPERIOR RULES      {sr8, sr9, sr10}
        ID                  13}

sr14    STRUCTURE-RULE      ::= {
        NAME FORM           tmnOSNameForm
        SUPERIOR RULES      {sr2, sr3, sr4}
        ID                  14}

sr15    STRUCTURE-RULE      ::= {
        NAME FORM           tmnOSNameForm
        SUPERIOR RULES      {sr5, sr6, sr7, sr35}
        ID                  15}

sr16    STRUCTURE-RULE      ::= {
        NAME FORM           tmnOSNameForm
        SUPERIOR RULES      {sr8, sr9, sr10}
        ID                  16}

sr17    STRUCTURE-RULE      ::= {
        NAME FORM           tmnNENameForm
        SUPERIOR RULES      {sr2, sr3, sr4}
        ID                  17}

sr18    STRUCTURE-RULE      ::= {
```

```
        NAME FORM           tmnNENameForm
        SUPERIOR RULES      {sr5, sr6, sr7, sr35}
        ID                  18}

sr19    STRUCTURE-RULE      ::= {
        NAME FORM           tmnNENameForm
        SUPERIOR RULES      {sr8, sr9, sr10}
        ID                  19}

sr20    STRUCTURE-RULE      ::= {
        NAME FORM           tmnNENameForm
        SUPERIOR RULES      {sr14, sr15, sr16}
        ID                  20}

sr21    STRUCTURE-RULE      ::= {
        NAME FORM           applProcessNameForm
        SUPERIOR RULES      {sr2, sr3, sr4}
        ID                  21}

sr22    STRUCTURE-RULE      ::= {
        NAME FORM           applProcessNameForm
        SUPERIOR RULES      {sr8, sr9, sr10}
        ID                  22}

sr23    STRUCTURE-RULE      ::= {
        NAME FORM           applProcessNameForm
        SUPERIOR RULES      {sr5, sr6, sr7, sr35}
        ID                  23}

sr24    STRUCTURE-RULE      ::= {
        NAME FORM           applProcessNameForm
        SUPERIOR RULES      {sr14, sr15, sr16}
        ID                  24}

sr25    STRUCTURE-RULE      ::= {
        NAME FORM           applProcessNameForm
```

	SUPERIOR RULES	{sr17, sr18, sr19, sr20, sr43
	ID	25}
sr26	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr2, sr3, sr4}
	ID	26}
sr27	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr8, sr9, sr10}
	ID	27}
sr28	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr5, sr6, sr7, sr35}
	ID	28}
sr29	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr14, sr15, sr16}
	ID	29}
sr30	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr17, sr18, sr19, sr20, sr43}
	ID	30}
sr31	STRUCTURE-RULE	::= {
	NAME FORM	applEntityNameForm
	SUPERIOR RULES	{sr21, sr22, sr23, sr24, sr25, sr48}
	ID	31}

sr32 STRUCTURE-RULE ::= {
NAME FORM apaeAliasNameForm
SUPERIOR RULES {sr14, sr15, sr16}
ID 32}

sr33 STRUCTURE-RULE ::= {
NAME FORM apaeAliasNameForm
SUPERIOR RULES {sr17, sr18, sr19, sr20, sr43}
ID 33}

sr34 STRUCTURE-RULE ::= {
NAME FORM apaeAliasNameForm
SUPERIOR RULES {sr21, sr22, sr23, sr24, sr25, sr48}
ID 34}

sr35 STRUCTURE-RULE ::= {
NAME FORM sOPNameForm
SUPERIOR RULES {sr1}
ID 35}

sr36 STRUCTURE-RULE ::= {
NAME FORM orgRoleNameForm
SUPERIOR RULES {sr2,sr3, sr4}
ID 36}

sr37 STRUCTURE-RULE ::= {
NAME FORM orgRoleNameForm
SUPERIOR RULES {sr8,sr9, sr10}
ID 37}

sr38 STRUCTURE-RULE ::= {
NAME FORM orgPersonNameForm
SUPERIOR RULES {sr2,sr3, sr4}
ID 38}

sr39 STRUCTURE-RULE ::= {

	NAME FORM	orgPersonNameForm
	SUPERIOR RULES	{sr8,sr9, sr10}
	ID	39}
sr40	STRUCTURE-RULE	::= {
	NAME FORM	networkNameForm
	SUPERIOR RULES	{sr5, sr6, sr7, sr35}
	ID	40}
sr41	STRUCTURE-RULE	::= {
	NAME FORM	networkNameForm
	SUPERIOR RULES	{sr2, sr3, sr4}
	ID	41}
sr42	STRUCTURE-RULE	::= {
	NAME FORM	networkNameForm
	SUPERIOR RULES	{sr8, sr9, sr10 }
	ID	42}
sr43	STRUCTURE-RULE	::= {
	NAME FORM	tmnNENameForm
	SUPERIOR RULES	{sr40, sr41, sr42, sr44}
	ID	43}
sr44	STRUCTURE-RULE	::= {
	NAME FORM	networkNameForm
	SUPERIOR RULES	{sr40, sr41, sr42, sr44 }
	ID	44}
sr45	STRUCTURE-RULE	::= {
	NAME FORM	tmnNEComplexNameForm
	SUPERIOR RULES	{sr2, sr3, sr4}
	ID	45}
sr46	STRUCTURE-RULE	::= {
	NAME FORM	tmnNEComplexNameForm

```
                SUPERIOR RULES      {sr8, sr9, sr10}
                ID                  46}

sr47    STRUCTURE-RULE      ::= {
                NAME FORM           tmnNEComplexNameForm
                SUPERIOR RULES      {sr5, sr6, sr7, sr35}
                ID                  47}

sr48    STRUCTURE-RULE      ::= {
                NAME FORM           applProcessNameForm
                SUPERIOR RULES      {sr45, sr46, sr47}
                ID                  48}

sr49    STRUCTURE-RULE      ::= {
                NAME FORM           applEntityNameForm
                SUPERIOR RULES      {sr45, sr46, sr47}
                ID                  49}

sr50    STRUCTURE-RULE      ::= {
                NAME FORM           apaeAliasNameForm
                SUPERIOR RULES      {sr45, sr46, sr47}
                ID                  50}


END
```

G.5 RRP Abstract Service

This clause includes all of the ASN.1 type and value definitions for the Registration Request Service in the form of the ASN.1 module "RegistrationRequestService". [14]

```
TMNRegistrationRequestService {iso org ansi(840) t1245(10041)
tmnDirmodule(1) tmnRegistrationRequestService(5)}

DEFINITIONS ::=
```

[14] Although the registrationBindError (defined in this clause) includes securityError, security of the TMN Directory is for further study.

```
BEGIN

-- EXPORTS All

IMPORTS

id-rrp-registrationAC, id-rrp-registrationAS,
id-rrp-opcode-registrationRequest, id-rrp-registrationError,
id-rrp-rosObject-rm, id-rrp-rosObject-registration,
id-rrp-rosObject-rrpRA, id-rrp-contract, id-rrp-package-rrpConnection
        FROM {iso org ansi(840) t1245(10041) tmnDirModule(1)
        tmnDirUsefulDefinitions(1)}

ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE,
CONNECTION-PACKAGE, OPERATION, ERROR
        FROM Remote-Operations-Information-Objects
        {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0)}

ROS{}, Bind{}, Unbind{}, InvokeID
        FROM Remote-Operations-Generic-ROS-PDUs
        {joint-iso-ccitt remote-operations(4) generic-ROS-PDUs(6) version1(0)}

APPLICATION-CONTEXT
        FROM Remote-Operations-Information-Objects-extension
        {joint-iso-ccitt remote-operations(4) informationObjects-extension(8) version1(0)}

acse, pData
        FROM Remote-Operations-Realisations
        {joint-iso-ccitt remote-operations(4) realisations(8) version1(0)}

acse-abstract-syntax
        FROM Remote-OperationsAbstractSyntaxes
        {joint-iso-ccitt remote-operations(4) remoteOperationsAbstractSyntaxes(12)
        version1(0)}

emptyUnbind
        FROM Remote-Operations-Useful-Definitions
```

```
        {joint-iso-ccitt remote-operations(4) useful-definitions(7) version1(0)}

    serviceError, securityError, ServiceProblem, SecurityProblem, Credentials
        FROM DirectoryAbstractService
        {joint-iso-ccitt ds(5) module(1) directoryAbstractService(2) 2}

    DistinguishedName
        FROM InformationFramework
        {joint-iso-ccitt(2) ds(5) modules(1) informationFramework(1) 2}

    PresentationAddress
        FROM SelectedAttributeTypes
        {joint-iso-ccitt(2) ds(5) modules(1) selectedAttributeTypes(5) 2};

--application contexts--

    registrationAC  APPLICATION-CONTEXT  ::= {
        CONTRACT                        rrpContract
        ESTABLISHED BY                  acse
        INFORMATION TRANSFER BY         pData
        ABSTRACT SYNTAXES   {acse-abstract-syntax | registration-abstract-syntax}
        APPLICATION CONTEXT NAME    id-rrp-registrationAC}

--ROS objects--

    rm  ROS-OBJECT-CLASS  ::= {
        INITIATES       {rrpContract}
        ID              id-rrp-rosObject-rm}

    registration  ROS-OBJECT-CLASS  ::= {
        RESPONDS        {rrpContract}
        ID              id-rrp-rosObject-registration}

    rrp-ra  ROS-OBJECT-CLASS  ::= {
        RESPONDS        {rrpContract}
        ID              id-rrp-rosObject-rrpRA}
```

```
--contracts--

        rrpContract  CONTRACT  ::= {
                CONNECTION                      rrpConnectionPackage
                INITIATOR CONSUMER OF           {registrationRequestOperation}
                ID                                      id-rrp-contract-rrp}

--connection package--

        rrpConnectionPackage  ::= {
                BIND            registrationBind
                UNBIND registrationUnbind
                ID              id-rrp-package-rrpConnection}

--abstract syntaxes--

        registrationAbstractSyntax  ABSTRACT-SYNTAX  ::= {
                RRP-PDU
                IDENTIFIED BY   id-rrp-registrationAS}

        RRP-PDU  ::=  CHOICE {
                basicRos        ROS {{RRP-InvokeIDSet},{RRP-Invokable},{RRP-Returnable}},
                bind            BIND {registrationBind},
                unbind          Unbind {registrationUnbind}}

        RRP-InvokeIDSet         ::=  InvokeID (ALL EXCEPT absent:NULL)
        RRP-Invokable           OPERATION       ::=  {registrationRequestOperation}
        RRP-Returnable OPERATION        ::=  {registrationRequestOperation}

--Bind and unbind operations--

        registrationBind  OPERATION  ::= {
                ARGUMENT        RegistrationBindArgument
                RESULT          RegistrationBindResult
                ERROR           registrationBindError}

        RegistrationBindArgument  ::=  SET {
```

```
        credentials     [0]     Credentials OPTIONAL,
        versions        [1]     Versions DEFAULT {v1995}}

RegistrationBindResult ::= RegistrationBindArgument

RegistrationBindError           ERROR ::= {
        PARAMETER               SET {
                versions        [0]     Versions DEFAULT {v1995},
                error                   CHOICE {
                serviceError            [1]     ServiceProblem,
                securityError           [2]     SecurityProblem,
                registrationError       [3]     ServiceDiagnostic }}}

registrationUnbind OPERATION ::= emptyUnbind

Versions ::= BIT STRING {v1995(0)}
```

--Operations, arguments, and results--

```
registrationRequestOperation OPERATION ::= {
        ARGUMENT                RegistrationRequestArgument
        RETURN RESULT           FALSE
        ERRORS                  {registrationError | serviceError | securityError}
        CODE                    id-rrp-registrationRequest}

RegistrationRequestArgument ::= SEQUENCE {
                                        primaryDsaAddress               AEAddressInfo,
                                        alternateDsaAddress             SET OF AEAddressInfo
OPTIONAL,
                                        namePrefix                      DistinguishedName
OPTIONAL}

AEAddressInfo ::= SEQUENCE {
        aeTitle   [0] DistinguishedName OPTIONAL,
        pAddress        PresentationAddress}
```

```
registrationError  ERROR  ::= {
                                PARAMETER               ServiceDiagnostic
                                CODE                    id-rrp-registrationError}

ServiceDiagnostic  ::=  SERVICEDIAGNOSTIC.&serviceDiagnosticType

SERVICEDIAGNOSTIC  ::=  CLASS
                {
                        &serviceDiagnosticType
                }
        WITH SYNTAX
                {
                        SERVICEDIAGNOSTIC  &serviceDiagnosticType
                }

END
```

Annex H
(normative)

H Additional Normative Specifications

H.1 Back-up Directory Server

The Directory Server that is active under normal conditions is designated as the *Primary Directory Server* (or P-DS). To eliminate susceptibility of the Directory to a single point of failure, it may be desirable to have a *Back-up Directory Server* (or B-DS).

The simple reliability scheme below allows for a back-up DSA such that catastrophic failures of the P-DS are protected against. Recovery times may be in the range of minutes.

The B-DS uses information from the IS–IS/ES–IS routing protocols to detect when communications with the P-DS are lost. At this point, the B-DS becomes the "active" Directory Server and proceeds to populate its DIB via registration of all reachable NEs in the manner described in 13.1.

Once the backup DSA has determined that the active DSA is no longer reachable, it should not begin re-registration of the reachable NEs until a reasonable timeout (in the order of a minute) has expired.

The B-DS will continue in the "active" Directory Server role until it regains communications with the P-DS, at which time the B-DS will yield the "active" Directory Server role to the P-DS. The P-DS then uses the automatic registration procedure described in 13.1 to (re-)populate its DIB.

To verify proper operation of the Directory Service function in the B-DS, the P-DS (only) is allowed to perform the DIB access procedures on the B-DS while the B-DS is in the stand-by mode. This permits the P-DS to perform routine tests on the B-DS and to report any problems it may encounter. Upon assuming the "active" Directory Server role, the B-DS shall flush from its DIB any data acquired through P-DS testing.

Annex I
(informative)

I Tutorial Information

I.1 Factors Affecting Directory Service Response Time

This informative clause lists the steps needed to perform a Directory Service information retrieval operation under a number of scenarios and provides a time estimate for each step to arrive at a total retrieval time for each case. The scenarios shown include: retrieval from local cache, retrieval from local domain DSA, and retrieval from a DSA outside the local domain via referral and via chaining. The estimated times are only intended to convey the order-of-magnitude of time that the step would take; the numeral "2" is arbitrarily used with the order-of-magnitude figure (i.e., 2 secs, 200 ms, 20 ms, etc.) since it best fits the largest factor affecting response time (association set-up @ ~2 secs).

I.1.1 Retrieving Information from Local Cache

Retrieving information from Local Cache only involves a local retrieval by the DUA and therefore is fastest; however, Local Caches are not required and are not likely to contain all the information a DUA would ever need. In practice, a local cache may be needed due to performance considerations. The retrieval time for this scenario is:

Table I.1

Steps	Est. Time
Search through cache	~2 ms
Total Estimated Retrieval Time	**~2 ms**

I.1.2 Retrieving Information from the Local DSA

Retrieving information from a Local DSA (L-DSA) requires that the DUA establish an association with the L-DSA and that a request/response exchange take place over this association. The retrieval time for this scenario is:

Table I.2

Steps	Est. Time
Establish association with DSA	~2 secs
Send Retrieval request to DSA	~200 ms
DSA searches through DIB	~20 ms
DSA returns retrieval response	~200 ms
Total Estimated Retrieval Time	**~2.42 secs**

I.1.3 Retrieving Information from a DSA Outside of the Local Domain

Figure I.1 illustrates a Directory. The user of the Directory (DUA) connects to a Local DSA (L-DSA), which may be part of a larger Directory. In this example, there are three Remote DSAs (R-DSAs). In many cases, DSAs will reside in adjacent areas and therefore inter-domain messages will only need to go through one Level 2 IS routing node. However, in some cases (e.g., to DSA B), these messages may have to go through many Level 2 IS nodes to reach their destination. The time it takes to go through a Level 2 IS node is estimated to be in the order of hundreds of milliseconds; the number of Level 2 IS routing nodes that a message shall go through is shown as "N".

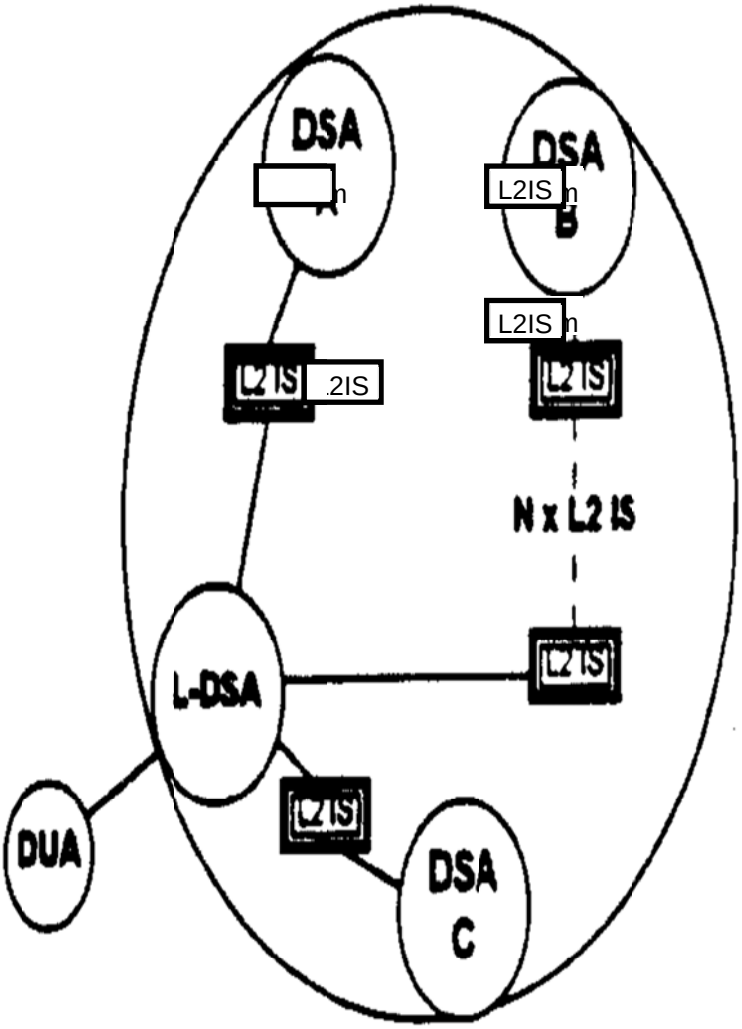


Figure I.1 - The Directory

I.1.4 Referrals

Figure I.2 illustrates how referrals work. In order to retrieve information from a DSA outside the local domain via referrals, the DUA shall first establish an association with its Local DSA (L-DSA) and request a referral. The L-DSA's response will consist of one or more remote DSAs (R-DSAs) addresses. The DUA then establishes an association with the R-DSA(s) and a request/response exchange takes place over this association. If the referral consisted of a list of addresses, the DUA may contact the R-DSAs sequentially or simultaneously, the latter being faster but requiring more resources on the DUA. The retrieval time for a single remote referral scenario, where the messages to the R-DSA shall go through three Level 2 IS routing nodes is:

Table I.3

Domain	Steps	Est. Time
Local	Setup association with Local DSA (L-DSA)	~2 secs
	Send retrieval/referral request to L-DSA	~200 ms
	L-DSA searches through its DIB	~20 ms
	L-DSA returns referral address/list	~200 ms
Remote (by DUA)	Setup association with Remote DSA (R-DSA)	~2 secs + 2*N* x 200 ms
	Send retrieval request to R-DSA	(*N*+1) x ~200 ms
	R-DSA searches through its DIB	~20 ms
	R-DSA returns retrieval response	(*N*+1) x ~200 ms
	Total Estimated Retrieval Time (single remote referral, *N* = 3)	**~7.24 secs**

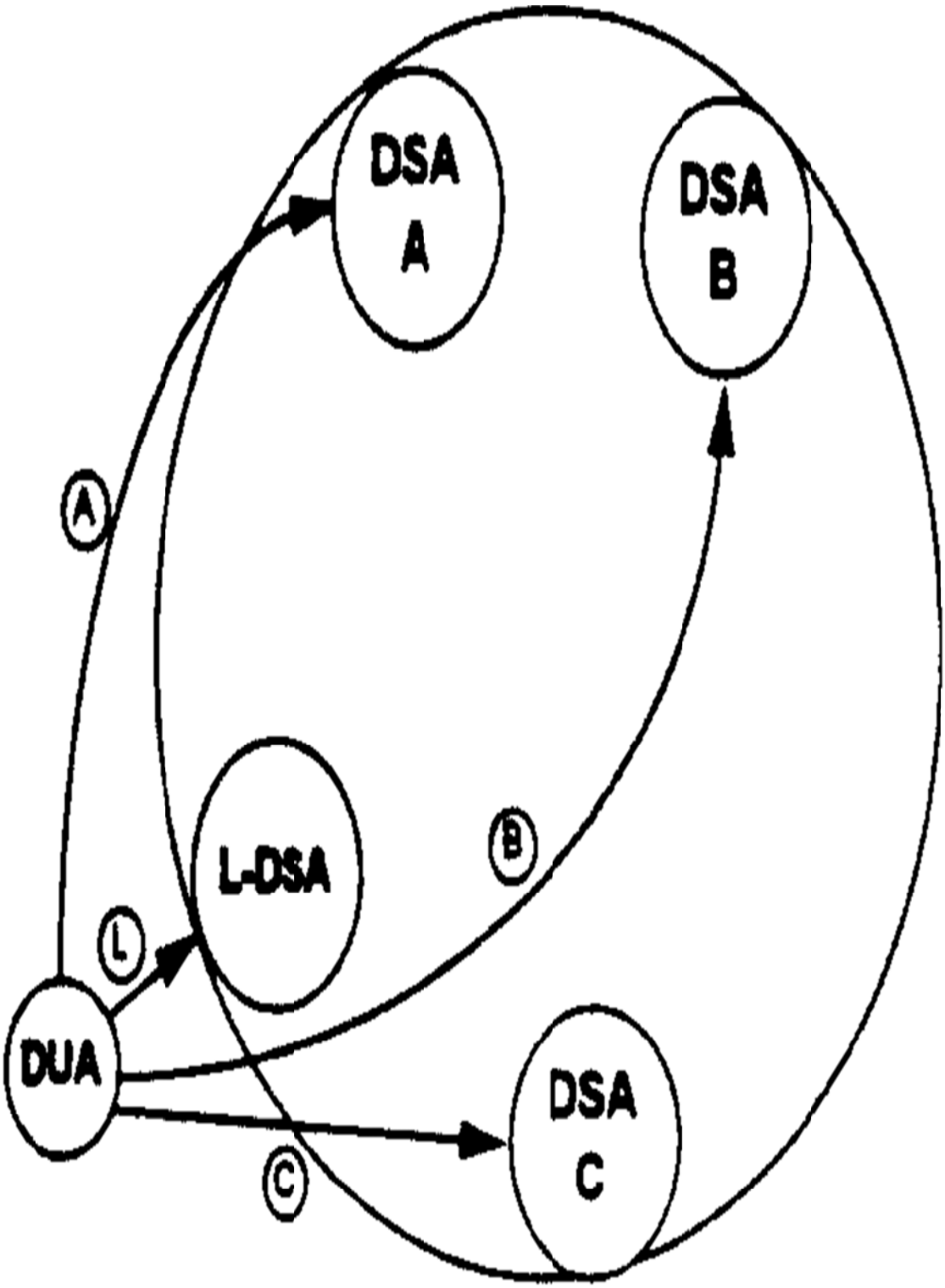


Figure I.2 - The Directory – Referrals

I.1.5 Chaining

Figure I.3 illustrates how chaining works. In order to retrieve information from a DSA outside the local domain via chaining, the DUA interacts with its Local DSA (L-DSA) in the same manner as it does when retrieving local information; the L-DSA handles all the interactions with the Remote DSAs (R-DSAs). The L-DSA establishes associations with the R-DSAs and conducts request/response exchanges over these associations. The L-DSA may contact the R-DSAs sequentially or simultaneously, the latter being faster but requiring more resources on the L-DSA. Once it acquires the requested information, the L-DSA issues a response to the DUA. The retrieval time for a single remote chaining scenario, where the messages to the R-DSA shall go through three Level-2 IS routing nodes is:

Table I.4

Domain	Steps	Est. Time
Local	Setup association with Local DSA (L-DSA)	~2 secs
	Send retrieval request to L-DSA	~200 ms
	L-DSA searches through DIB	~20 ms
Remote (From L-DSA)	L-DSA sets-up association with Remote DSA (R-DSA)	~2 secs + 2*N* x ~200 ms
	L-DSA sends retrieval request to R-DSA	(*N*+1) x ~200 ms
	R-DSA searches through its DIB	~20 ms
	R-DSA returns retrieval response to L-DSA	(*N*+1) x ~200 ms
Local	L-DSA returns retrieval response to DUA	~200 ms
	Total Estimated Retrieval Time (single remote chaining, *N* = 3)	**~7.24 secs**

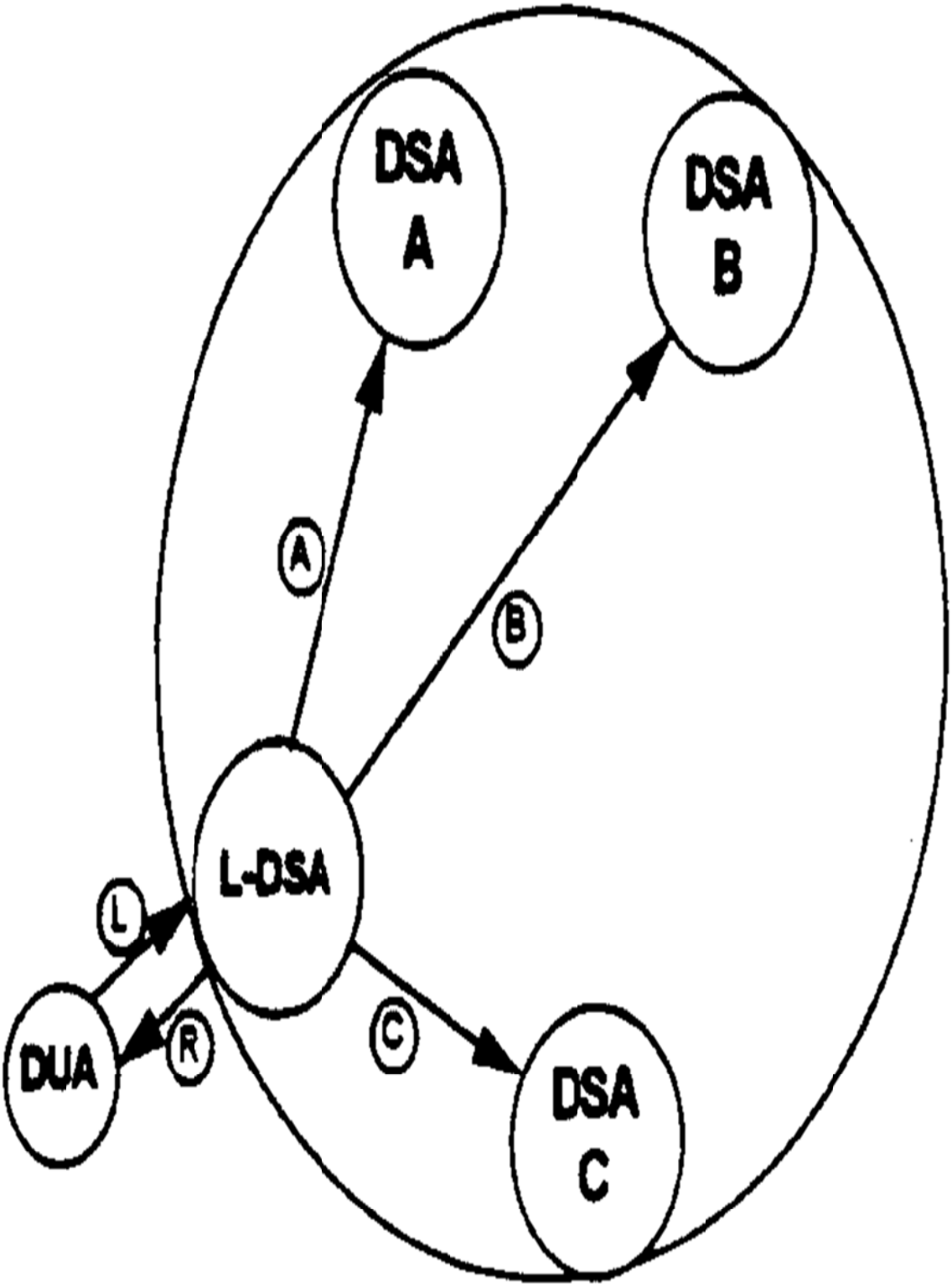


Figure I.3 - The Directory – Chaining

To minimize the time it takes to retrieve information from DSAs outside of the local domain, it may be possible to keep associations between DSAs up permanently in the chaining case. This eliminates a time consuming step and reduces the retrieval time for this scenario as follows:

Table I.5

Domain	Steps	Est. Time
Local	Setup association with Local DSA (L-DSA)	~2 secs
	Send retrieval request to L-DSA	~200 ms
	L-DSA searches through DIB	~20 ms
Remote (From L-DSA)	L-DSA sends retrieval request to R-DSA	(*N*+1) x ~200 ms
	R-DSA searches through its DIB	~20 ms
	R-DSA returns retrieval response to L-DSA	(*N*+1) x ~200 ms
Local	L-DSA returns retrieval response to DUA	~200 ms
	Total Estimated Retrieval Time (single remote chaining, *N* = 3)	**~4.04 secs**

I.2 Directory Server SONET Deployment Examples

The Directory Server used for SONET systems operates over the Q3 interface. Figures I.4 and I.5 show examples of how a DS may be deployed in a SONET subnetwork. Figure I.4 shows an abstract view of the DS in a SONET subnetwork, while Figure I.5 shows an example of where a DS may be deployed in a SONET subnetwork.

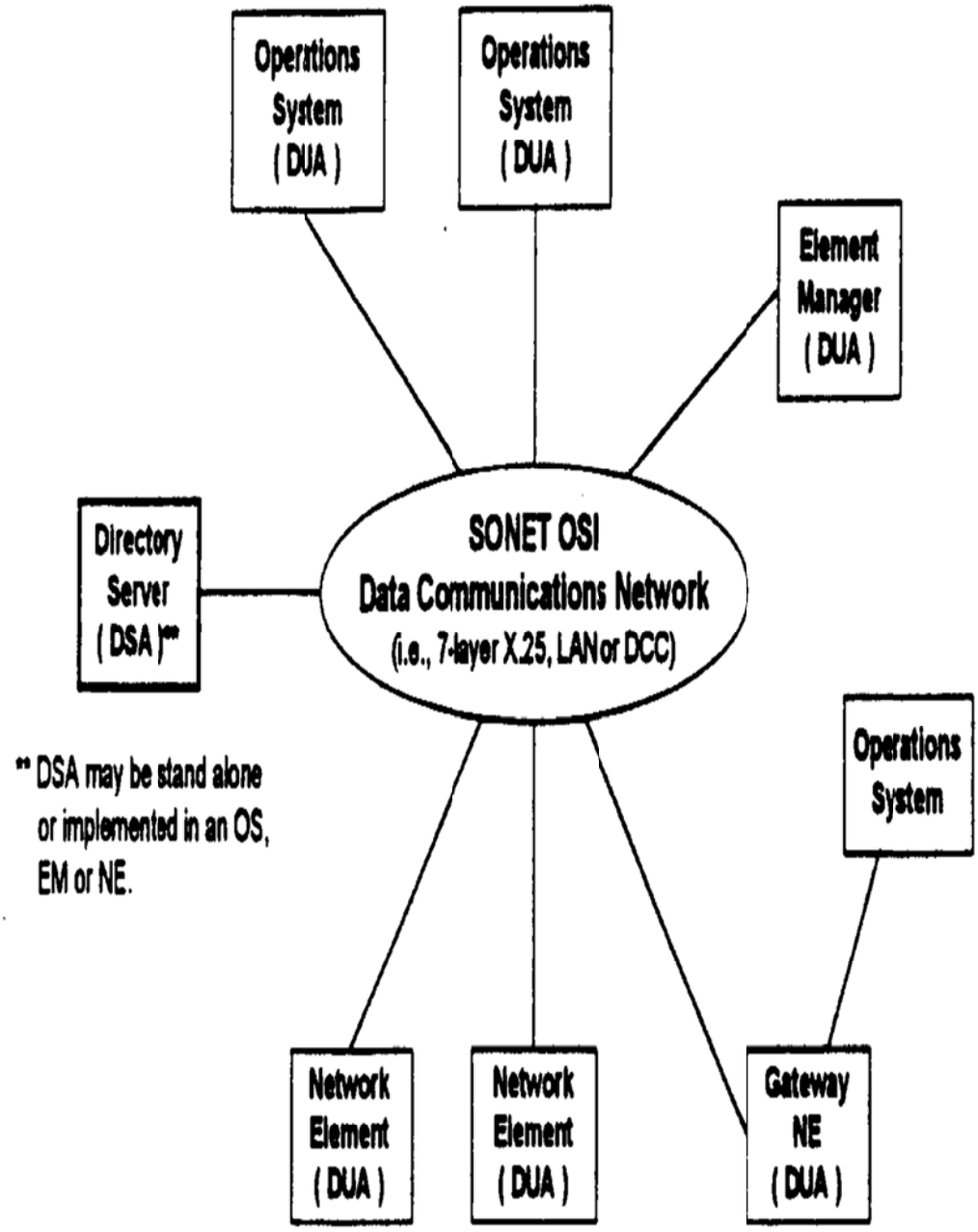


Figure I.4 - Directory Server deployment (abstract view)

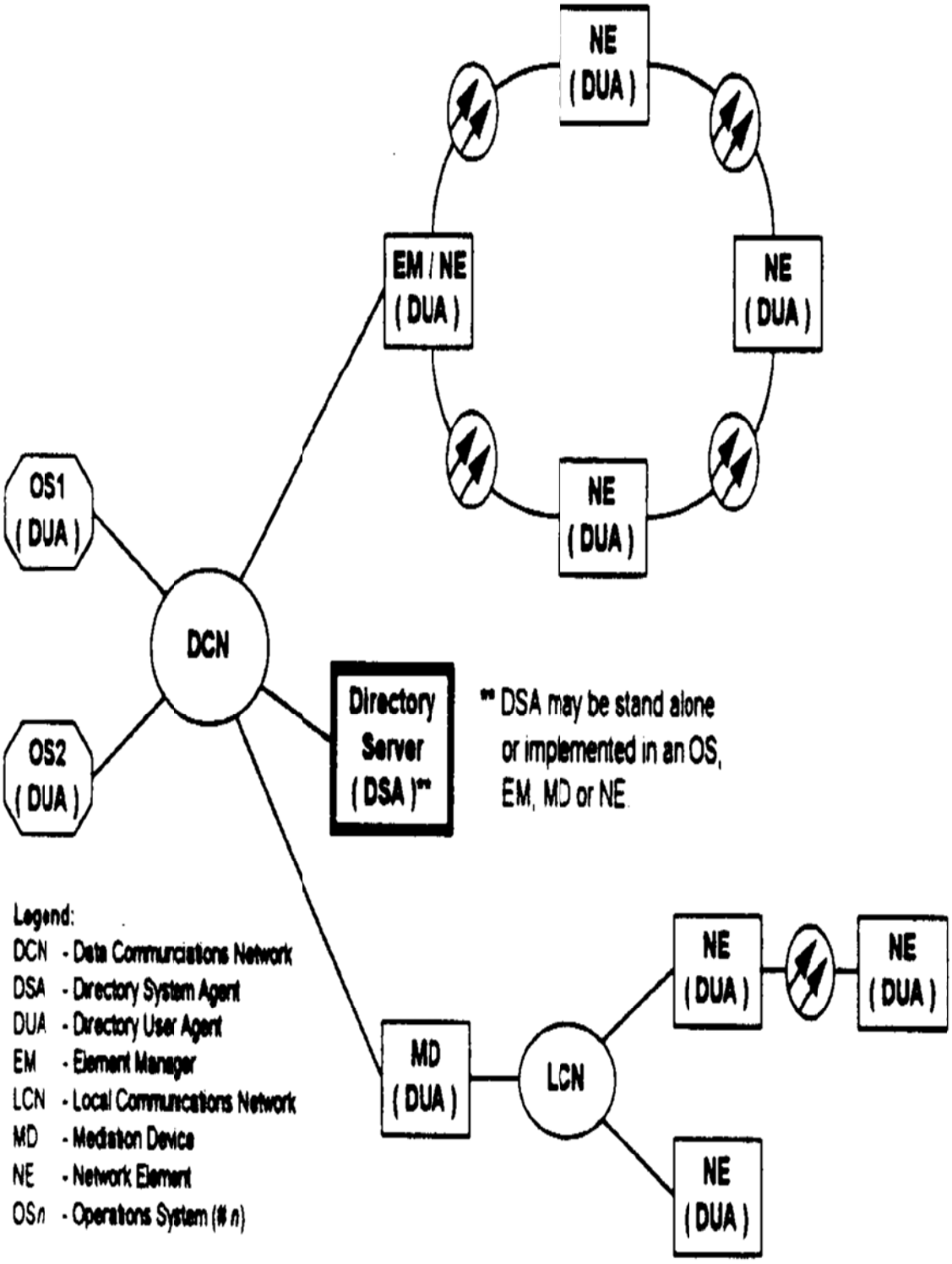


Figure I.5 - Directory Server deployment (example)

I.3 Use of the Directory Service for TMN

This standard describes a minimum set of Directory objects and structure elements for a Directory Service for TMN, in order to assure that the requirements stated in clause 6 are met. This subclause examines each of the requirements in turn and provides detail on how they are satisfied by the defined DIT.

Note that this standard does not preclude additional directory objects and structure being implemented within a compliant Directory for reasons not stated here, nor does it preclude use of the Directory to satisfy additional requirements.

I.3.1 Naming/Addressing Support

> Requirement 6.1(a): [This Directory Service supports the determination of]
> the TMN elements involved in the management subsystem;

The elements of primary interest for identification are the Network Elements that are the managed resources represented by instances of the managedElement managed object class or its subclasses defined in ITU-T Recommendation M.3100. They are represented in the Directory by entries of the class tmnNE (and its subclasses).

ITU-T Recommendation M.3010 notes that all TMN NEs have Q interfaces. There may also be NE-like equipment without TMN-standardized interfaces that is accessible through a Q-Adapter Function (on a Q-Adapter or Mediation Device). Although these may be considered outside the TMN, by virtue of the *Q-Adapter Function* (QAF), they may be manageable from within the TMN as if they were TMN NEs. Directory entries of the class tmnNE could also be used for such NE-like equipment to support this extended reach. (The value of the attribute "entityAddress" for such an entry, if any, would have only local significance.)

NEs are identified through a structure of country, locality, organization, and organizationalUnit entries that may reflect ownership, jurisdiction, domains, or network structure. The requirements for the DIT structure to reflect any particular useful organizational information are not developed or standardized.

Entries of type Country are typically held in a superior DSA (country-owned), but might be mirrored locally where the DSA is not externally connected. Entries of type Locality are also typically held in a superior DSA "owned" by the locality (such as a state), but they may also be held by the country's DSA. These entries pertain to a geographical or logical area.[15]

Entries of types organization and organizationalUnit refer to the root and recursive subdivisions of some organized hierarchy of control, ownership, or jurisdiction. This standard does not specify how these are to be used. They may or may not reflect TMN or organizational hierarchy, but they most likely do reflect some aspects of carrier, service, or network boundaries (i.e., domains).

The organizational hierarchy in the TMN Directory Service is there solely to provide a logical organization of entries, and is defined to facilitate identifying the elements involved in the management subsystem. It provides an organization that is logical from the point of view of the user (human or application). Much depth is not recommended, since a deep hierarchy is more prone to changes and lower reaches become obscure and less helpful to use as differentiation points in searches.

---

[15] Although ITU-T Recommendation X.521, Annex B, suggests that Locality can be subordinate to Organization, this is deprecated since a locality exists independent of any particular organization. (If a company has a New York division, that is considered an organizationalUnit named by a localityName rather than a case of New York being subordinate to the company.)

One purpose for the Directory Service is to enable identification and addressing of specific applications, which are the managers and agents relating to the NEs. These are covered by the use of applicationProcess, applicationEntity, and apaeAlias entries.

Containment of such an entry in the DIB subordinate to an entry for an NE implies that that application supports its application context for that NE. Thus, an applicationEntity entry indicating the system management context, under a tmnNE entry, is interpreted as the agent for that NE. This cannot be rigorously assumed, since the AE could also be a manager application resident on the NE (if the NE has an OS Function). Note that the AE need not be resident on the NE (the applicationEntity entry has its own address attribute); thus, the use of proxy agents is accommodated. An entry of the type apaeAlias is used to point to the entry for the application if it exists elsewhere in the DIB. This is expected to be the norm in the case of proxy agents, where the actual entry exists under the organization or under the NE on which the application resides.

ApplicationEntity and applicationProcess entries may exist independent of NE entries, and be subordinate to organization or organizationalUnit entries. Interpretation in this case is not standardized. Typically, this is used to provide transparency of the actual device where the application resides, and to provide an indication of the domain to which the application pertains; for example, an applicationProcess entry for a manager application, placed under an organizationalUnit entry for a local network, may represent the Operations System Function for that domain.

In order to facilitate identification of AEs, the standard includes a content rule to specifically require that the supportedApplicationContexts attribute be present in all applicationEntity entries (ITU-T Recommendation X.521 includes it as an optional attribute).

There is currently no support for identifying the TMN elements involved in the management subsystem on the basis of role (OS, MD, QA, etc.), nor to identify application processes on the basis of function (QAF, NEF, OSF, etc.) as described in the TMN model. The Logical Layered Architecture for the TMN is not reflected in the Directory in a standardized way (it is not possible to separate Element Management entities from Service Management entities, for example). Careful design of the organizational structure and the use of applicationEntity entries within such a structure can accommodate certain aspects of domain, and the use of the supportedApplicationContexts attribute in applicationEntity entries can accommodate certain aspects of role. Requirements for Directory support for more of the TMN model have not been determined.

Requirement 6.1(b): [This Directory Service supports the determination of] the identity of a Network Element based on its Network Address;

The tmnNE directory object class includes an attribute "entityAddress". If the Network Address (e.g., NSAP) is known, the tmnNE entry may be found by initiating a search, with a base object defined to the nearest organizationalUnit entry that is known, specifying a filter of objectClass=tmnNE and entityAddress=<known value>.

If the organizationalUnit is known to the level just above the tmnNE entry, a List operation could also be performed to retrieve all the tmnNE entries within that organizationalUnit; these could then be checked by the application for the presence of the desired Network Address.

Note that the entityAddress attribute may have multiple values (network redundancy, for example, or different network connections for different purposes). This will not affect the ability to find an NE with a given Network Address. It does mean, however, that it is more difficult to provide the reverse look-up (find the Network Address for a given NE) since it would not be possible to attach any significance to a particular value (primary/backup, etc.).

Requirement 6.1(c): [This Directory Service supports the determination of] the identity of an NE based on technology-, implementation-, network- or operation-dependent naming (such as M.3100:userLabel) where available;

The tmnNE directory object class includes an attribute "commonName". This is used to provide a unique name within the superior organizationalUnit. It is, however, a multi-valued attribute. Only the distinguished value that is used to provide the entry's Directory Name is strictly governed within the DIT structure. Other values may reflect other names that are unique within some other useful domain. Subclause 9.2 discusses particular values that are synchronized with the MIB to provide alternate names for searching. Use of other values is not defined by this standard and is left up to the applications using the Directory.

If such a name is known, the tmnNE entry may be found by initiating a search, with a base object defined to the nearest organizationalUnit entry that is known, specifying a filter of objectClass=tmnNE and commonName=<known value>. Note that uniqueness requirements are not enforced by the Directory but are assumed to be inherent in the nature of the source for these names. Multiple entries may be returned if the source of these alternate names does not enforce uniqueness, or if the sources for different alternate values draw from the same name space. (This could occur, for example, if both systemTitle and userLabel were placed into commonName and the systemTitle of one NE was the same as a userLabel for another.)

Requirement 6.1(d): [This Directory Service supports the determination of] the identity of NEs based on vendor or locality information, where known;

The tmnNE directory object class includes the attributes "vendorName" and "localityName". Subclause 9.2 discusses synchronization of these with the MIB values for the corresponding managedElement managed object. If such a name is known, the tmnNE entry may be found by initiating a search, with a base object defined to the nearest organizationalUnit entry that is known, specifying a filter of objectClass=tmnNE and vendorName (or localityName)=<known value>. Note that these names are unlikely to be unique, and so the search could return multiple tmnNE entries. The use of standardized values for naming a particular vendor or locality is not covered by this standard.

Requirement 6.1(e): [This Directory Service supports the determination of] the identity of an NE from its MIB naming attribute (M.3100:managedElementId);

The managedElementId attribute is an attribute of the tmnNE entry. A search specified within a suitable domain (base object) with a filter of managedElementId = <known value> will return the corresponding NE entry. Note that there is a possibility of such a search returning more than one entry if managedElementIds are not unique within the domain of the search (in the MIB, they only need be unique within the containing network; a Search in the Directory spanning multiple networks could result in more than one hit).

Requirement 6.1(f): [This Directory Service supports the determination of] the value of the M.3100:managedElementId of an NE for use in management protocol exchanges;[16]

The managedElementId is an attribute of the tmnNE entry. Thus, if any identifying information is known about the NE (such as any of those described above), its entry may be found as described above by a Search operation with the appropriate filter set to the known information. The Search operation can request the return of the value of the managedElementId attribute.

Requirement 6.1(g): [This Directory Service supports the determination of] the identity of an NE based on its function and role.

The neType attribute is an attribute of the tmnNE entry. A search specified within a suitable domain (base object) with a filter of neType = <known value> will return the corresponding NE entry. Note that there is a likelihood of such a search returning more than one entry.

Requirement 6.1(h): [This Directory Service supports the determination of] the identity of a Bidirectional Line Switched Ring (BLSR) NE based on its ring node ID information.

The nodeIdInfo attribute is an attribute of the sdhNEEntry auxiliary object class intended for use with the tmnNE entries of SONET/SDH NEs belonging to one or more BLSRs. A search specified within a suitable domain (base object) with a filter of nodeIdInfo = <known value> will return the corresponding NE entry. The nodeIdInfo attribute is made up of two parts: nodeId and ringId. The nodeIdInfo matching rule allows users of the directory to search the DIB based on either part. To search based on nodeId only, the ringID parameter shall be absent from the nodeIdInfo filter value, and vice versa. Note that searches performed on a single parameter are likely to return more than one entry.

I.3.2 Association Resolution

Requirement 6.2(a): [This Directory Service supports the determination of] the AE titles of the entities with which management associations may be established;

Within OSI systems management, AE-Titles are used as pointers to peer entities; for example, event forwarding is based on eventForwardingDiscriminators, which store the target destinations as AE-Titles. AE-Titles are also a logical way for a manager AE to keep track of the agents it is overseeing (as part of configuration data or locally stored dynamic information). Obtaining an AE-Title is typically an intermediate step in using the Directory to look-up more useful information about that entity.

An applicationEntity entry contains a "commonName" attribute which is used to form its Distinguished Name, which is the Directory form of the AE-Title.

An AE-Title is a way of naming an Application Entity that is part of an Application Process. An Application Process may contain a number of Application Entities. An Application Process has an AP-Title that shall

[16] One such usage will be the Distinguished Name form of the baseManagedObject parameter in a CMISE GET operation. Since the managedElement MO is the top of the containment tree on an NE, it is quite likely to be specified as a base object in management operations.

be unique within the Open System Environment (i.e., globally). An AE-Title is made from the AP-Title plus an AE-qualifier. The AE-qualifier need only be unique within the AP to ensure that the AE-Title is also globally unique.

In accordance with ITU-T Recommendation X.227 (ACSE), AP-Title and AE-Title take two possible forms. One form is a registered object identifier, which is effective but obviates the need for a Directory.[17] The other form is the Name form, which is equivalent to a Directory Distinguished Name. This is a sequence of Relative Distinguished Names that define a path down the Directory Information Tree to the entry corresponding to the AP or AE.

The Directory object classes applicationProcess and applicationEntity *do not* have attributes called "title" with a specific syntax appropriate to AP- and AE-Titles. Instead, for both classes the naming attribute specified in the schema structure rules is "commonName", which has a generic syntax. This is because the naming attributes are *not* themselves the AP- and AE-Titles, but are a portion of them.

The commonName attribute for an applicationProcess is only the final RDN in the chain that constitutes the AP-Title. To get the AP-Title, one shall construct the DN of the applicationProcess by concatenating, in sequence, the RDNs of all nodes superior to the applicationProcess entry, plus the commonName attribute of the entry itself.

Similarly, the commonName attribute for an applicationEntity is only the final RDN in the chain that constitutes the AE-Title. When the applicationEntity's entry in the DIT is subordinate to an entry for an applicationProcess, then its commonName attribute is just the AE-qualifier. The AE-Title is then constructed by adding this to the AP-Title constructed above. The Directory schema also allows an applicationEntity entry to be placed in the DIT *without* a superior applicationProcess entry (since sometimes it is not worth distinguishing between them). In that case, the commonName of the applicationEntity entry would be a combination of the AE-qualifier/RDN and the RDN that would have been the commonName for the applicationProcess entry if there had been one, so that the full AE-Title would still be defined by the naming path from the top of the DIT to the applicationEntity entry.[18]

Thus, the AE-Title may be obtained by locating the entry for the AE in the Directory. The identification of Application Processes and Application Entities with which peer entities may wish to communicate is described in connection with Requirement 6.1(a). This may be done by a Search operation if some other attribute of the AE is known. For example, the Search may start with a tmnNE as a base object and look for subordinate applicationEntity entries with supportedApplicationContext=<system management>, to find the agent for that NE based on the assumptions described for Requirement 6.1(a) above. The search result includes the Directory name of the applicationEntity entry, which is the AE-Title.

Requirement 6.2(b): [This Directory Service supports the determination of] the presentation addresses of those entities;

The applicationEntity entry includes a "presentationAddress" attribute, which includes at least one value. Once the applicationEntity entry is located, a Read operation can be performed to obtain the value of the presentationAddress attribute. Note that this attribute has a single value, but the network address component of a presentationAddress value may itself have multiple values (i.e., a set of values), if network redundancy has been provided.

———

[17] The standards and the OIW Stable Implementation Agreements clearly indicate that the Directory is not used for OID look-up.

[18] Note that ISO 7498-3 on Naming and Addressing would suggest that part of an AP-Title might be the system title for the system on which the process resides, yet the proposed schema in ITU-T Recommendation X.521 does not include a system entry as a potential superior to applicationProcess. The system title may be buried within the commonName, or perhaps the authors of ITU-T Recommendation X.521 were taking a broader view of an application process and felt that the system (device entry) and processes are unrelated.

Requirement 6.2(c): [This Directory Service supports the determination of]
given a managed object and optionally the name of a desired management capability, the identity of one or more management agents capable of providing that management function.

The Directory does not currently contain the necessary management knowledge to provide this type of service. The current structure allows a Search operation that will identify an AE supporting the system management application context for a given NE (managedElement), but does not contain sufficient detail to refine this to contained managed objects or specific management capabilities.

I.3.3 Management Knowledge

Requirement 6.3: This Directory Service supports the determination of:

a. the supported application contexts of a management application entity;

b. the supported functional units of a management application entity;

c. the supported management profiles of a management application entity;

d. the list of managed objects or managed object classes in a managed application entity; and

e. the grouping of managed systems and management systems into management domains.

The Directory shall not be used for dynamic state information which is obtainable from the MIB.

The above requirements are currently not covered in this standard. Work is underway within ISO to define Directory object classes to contain this management knowledge. The classes defined in ISO Draft International Standard (DIS) 10164-16 are auxiliary classes expected to be used with applicationEntity entries to add system management knowledge to the DIB for AEs which are management AEs. An implementation could use these to satisfy the above requirements, but inclusion in this standard is for further study.

Note that the supportedApplicationContexts attribute of the applicationEntity object allows 6.3(a) to be satisfied in a generic sense. Use of this attribute is mandatory for applicationEntity entries within the Directory Service for TMN.

The statement that “the Directory shall not be used for dynamic state information” reflects an agreement that it is inappropriate to use the Directory for information that will change frequently or for attributes of managed objects that are more appropriately accessed by manager applications through management protocol exchanges with agents. The Directory is intended to provide support in the form of locating and identifying TMN elements involved in the management subsystem, with sufficient relatively static information to support identification and the management protocol exchanges that will take place.

I.3.4 Security Support

These requirements have not yet been addressed by the standard but will be covered in future expansions of the standard.

Requirement 6.4(a): [This Directory Service supports the determination of]
passwords for management entities to support simple authentication during management association establishment;

Passwords are currently not supported in the standard. The userPassword attribute, which ITU-T Recommendation X.521 uses with entries corresponding to persons, is not available with the applicationEntity object class.

Requirement 6.4(b): [This Directory Service supports the determination of] certified public keys for management entities to support strong authentication during management association establishment;

Some capability is available in the profile defined in the DAP PICS, but its use within the Directory Service for TMN is not described by this standard. This is a subject for further study.

Requirement 6.4 (c): [This Directory Service supports the determination of] access control information to support access to the Directory.

Simple access control is available in the profile defined in the DAP PICS, but its use within the Directory Service for TMN is not described by this standard. This is a subject for further study.

This TMN Directory standard does not address the full authentication framework for TMN services. Future expansions of the TMN Directory standard will address such a framework. Such a framework may be based, for example, on public key encryption as described in ITU-T Recommendation X.509.

I.3.5 Administrative Usage

Requirement 6.5: This Directory Service allows: a. a human administrator to search, modify, add and delete entries and sub-tree structures (within access rights) for the purposes of management of this Directory; and b. an application to search, modify, add and delete entries and sub-tree structures (within access rights) for the purposes of administering the Directory.

This capability is inherent in the operations defined for the Directory and required by the DAP profile specified in Annex B. A human administrator will require an administrative application incorporating a DUA and providing a user interface; an application shall incorporate a DUA. Certain implementations may also provide an administrator interface to the Directory data that is outside the scope of the ITU-T Recommendation X.500 protocols. This is also outside the scope of this standard.

I.3.6 Messaging & Use by Persons

Requirement 6.6: In order to support interpersonal messaging, this Directory Service supports the determination of: a. the ITU-T Recommendation X.400 addresses of roles and persons involved in TMN management; and b. other contact information (telephone numbers, facsimile, postal etc.) of organizations, roles and persons involved in TMN management.

These requirements are currently not satisfied by mandatory elements of this standard. However, the information profile in annex A does allow optional support of person and related object classes and attributes defined in ITU-T Recommendations X.520/521, and directory object classes and attributes defined by ITU-T Recommendation X.402. If these are included in an implementation, then these requirements can be satisfied. Inclusion of these elements as mandatory for a Directory Service within TMN is for further study.

Annex J
(normative)

J Back-up Registration Manager

J.1 Back-up Registration Manager

Even with the use of shadowing, a potential point of failure for the DS remains. If a Registration Manager (RM) failed, automatic registration would be halted until the RM came back on-line. Use of a back-up RM (B-RM) would prevent this.

A B-RM would detect the failure of the Primary RM (P-RM), and subsequently make itself the new P-RM. The determination of primary and back-up roles and switching between them is not covered within the scope of this document.