5. Draft new Recommendation X.525

Information Technology Ñ
Open Systems Interconnection Ñ
The Directory: Replication

Draft Recommendation X.525
ISO/IEC DIS 9594-9

Contents page

Introduction

This Recommendation|International Standard, together with the others of the series, has been produced to facilitate the interconnection of information processing systems to provide directory services. The set of all such systems, together with the directory information which they hold, can be viewed as an integrated whole, called the Directory. The information held by the Directory, collectively known as the Directory Information Base (DIB) is typically used to facilitate communication between, with or about objects such as application-entities, people, terminals and distribution lists.

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

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

This Recommendation|International Standard defines the Replication capabilities provided by DSAs to improve the level of service to Directory users.

Information Technology - Open Systems Interconnection - The Directory - Replication

1 Scope

This Recommendation|International Standard specifies a shadow service which DSAs may use to replicate Directory information. The service allows Directory information to be replicated among DSAs to improve service to Directory users. The shadowed information is updated, using the defined protocol, thereby improving the service provided to users of the Directory.

2 Normative references

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

- Ñ CCITT Recommendation X.500 (1992) | ISO/IEC 9594-1:1992, Information Technology Ñ Open Systems Interconnection Ñ The Directory: Overview of Concepts, Models and Services.
- Ñ CCITT Recommendation X.501 (1992) | ISO/IEC 9594-2:1992, Information Technology Ñ Open Systems Interconnection Ñ The Directory: Models.
- Ñ CCITT Recommendation X.511 (1992) | ISO/IEC 9594-3:1992, Information Technology Ñ Open Systems Interconnection Ñ The Directory: Abstract Service Definition.

- Ñ CCITT Recommendation X.518 (1992) | ISO/IEC 9594-4:1992, Information Technology Ñ Open Systems Interconnection Ñ The Directory: Procedures for Distributed Operation.
- Ñ CCITT Recommendation X.519 (1992) | ISO/IEC 9594-5:1992, Information Technology Ñ Open Systems Interconnection ÑThe Directory: Protocol Specifications.
- Ñ CCITT Recommendation X.520 (1992) | ISO/IEC 9594-6:1992, Information Technology Ñ Open Systems Interconnection ÑThe Directory: Selected Attribute Types.
- Ñ CCITT Recommendation X.521 (1992) | ISO/IEC 9594-7:1992, Information Technology Ñ Open Systems Interconnection ÑThe Directory: Selected Object Classes.
- Ñ CCITT Recommendation X.509 (1992) | ISO/IEC 9594-8:1992, Information Technology Ñ Open Systems Interconnection Ñ The Directory: Authentication Framework.
- Ñ ISO/IEC DIS 8824-1:1992, Information Technology Ñ Open Systems Interconnection Ñ Abstract Syntax Notation One (ASN.1): Specification of Basic Notation.
- Ñ ISO/IEC DIS 8824-2:1992, Information Technology Ñ Open Systems Interconnection Ñ Abstract Syntax Notation One (ASN.1): Information Object Specification.
- Ñ ISO/IEC DIS 8824-3:1992, Information Technology Ñ Open Systems Interconnection Ñ Abstract Syntax Notation One (ASN.1): Constraint Specification.
- Ñ ISO/IEC DIS 8824-4:1992, Information Technology Ñ Open Systems Interconnection Ñ Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 Specifications.
- Ñ CCITT Recommendation X.219 (1988) Remote Operations Ñ Model, Notation and Service Definition.

  ISO/IEC 9072-1:1989, Information Processing Systems Ñ Text Communication Ñ Remote Operations Part 1: Model, Notation and Service Definition.

  ISO/IEC 9072-2:1989, Information Processing Systems Ñ Text Communication Ñ Remote Operations: Protocol Specification.
- Ñ CCITT Recommendation X.218 (1988) Reliable Transfer.

  ISO/IEC 9066-1:1989, Information Processing Systems Ñ Text Communication Ñ Reliable Transfer.

3 Definitions

3.1 The following term is defined in CCITT Rec. X.500 | ISO/IEC 9594-1;

a) (the) Directory.

3.2 The following terms are defined in CCITT Rec. X.501 | ISO/IEC 9594-2;

a) distinguished name;
b) Directory Information Tree*;*
c) DSA Specific Entry;
d) DSA Information Model;
e) DSA Information Tree;
f) Directory System Agent;
g) operational attributes;
h) user attributes.

3.3 The following terms are defined in CCITT Rec. X.518 | ISO/IEC 9594-4;

a) access point;

b) knowledge information*;*

c) name resolution*;*

d) naming context;

e) non-specific subordinate reference*;*

f) subordinate reference*.*

3.4 For purposes of this Recommendation|International Standard, the following definitions apply:

a) *area prefix:* The sequence of RDNs and associated administrative information common to all entries within a replicated area;

b) *cache copy:* A copy of an entry (or part of an entry) whose consistency with its corresponding entry is maintained by means outside the scope of this Directory Specification;

c) *caching:* The process of creating cache copies. This process is outside the scope of this Directory Specification;

d) *consumer reference:* The access point of the shadow consumer;

e) *entry copy:* Shadowed information from an entry;

f) *extended knowledge:* Those subordinate references that would be included as subordinate knowledge if the replicated area were extended to the lower boundary of the naming context;

g) *master DSA:* The DSA which has administrative authority for a naming context. All adds, deletes and modifications to entries in this naming context are done by this DSA. The master DSA may enter into a shadowing agreement with another DSA to provide copies of a subset of this naming context (see unit of replication);

h) *primary shadowing:* Shadowing where the shadow supplier is the master DSA;

i) *replicated area:* A subtree from which entries for replication are selected;

j) *replication:* The process by which copies of entry and operational information are held by DSAs other than the master DSA;

k) *replication base entry:* The distinguished name of the root vertex of the replicated area;

l) *secondary shadowing:* Shadowing where the shadow supplier is not the master DSA;

m) *shadow consumer:* A DSA that receives shadowed information;

n) *shadow service:* The service provided to perform shadowing between two DSAs that have entered into one or more shadowing agreements;

o) *shadow supplier:* A DSA that provides shadowed information. This DSA may or may not be the master DSA;

p) *shadowed DSA specific entry (SDSE):* A unit of shadowed information which is associated with a specific name; it represents the information taken from a DSE which is shadowed;

q) *shadowed information:* The complete set of information associated with a unit of replication. Shadowed information is conceptually held both by the shadow supplier and the shadow consumer for the purposes of the shadow protocol and comprises a tree shaped structure of shadowed DSEs;

r) *shadowing:* Replication between two DSAs whereby shadowed information is copied and maintained using the Directory Information Shadowing Protocol;

s) *shadowing agreement:* The terms conforming to an agreement between two DSA Administrative Authorities (one being the Administrative Authority of the shadow supplier and the other being the Administrative Authority of the shadow consumer);

t) *supplier reference:* The access point of the shadow supplier;

u) *unit of replication:* A specification of the information to be shadowed, including (optionally) subordinate knowledge information.

4 Abbreviations

The following abbreviations are used in this Recommendation|International Standard:

DIB Directory Information Base

DISP Directory Information Shadowing Protocol

DIT Directory Information Tree

DSA Directory System Agent

DSE DSA Specific Entry

RDN Relative Distinguished Name

SDSE Shadowed DSA Specific Entry

5 Conventions

This Directory Specification has been prepared according to the "Presentation of CCITT/ISO/EC common text" guidelines in the Guide for CCITT and ISO/IEC JTC 1 Cooperation, September 1991. Exceptions to these guidelines are described below.

The term "Directory Specification" (as in "this Directory Specification") shall be taken to mean CCITT RecommendationÊX.525|ISO/IECÊ9594-9. The term "Directory Specifications" shall be taken to mean the X.500-Series of Recommendations and all parts of ISO/IEC 9594.

Some Directory Specifications may be organized into sections which signal to the reader the beginning of material significantly different in content to that proceeding. Division into sections merely aids the user and does not affect clause numbering.

If the items in a list are numbered (as opposed to using "Ð" or letters), then the items shall be considered steps in a procedure.

All text in a note is indented (not just the first line). The term "note" is in boldface type and all the note is in nine point. Some notes may contain requirements.

Those Directory Specifications defining Directory operations do so using the Remote Operation notation defined in CCITT RecommendationÊX.219|ISO/IEC 9072-1.

All ASN.1 notation is in a bold sans serif typeface. If an ASN.1 type or value is used in text, it is in a bold sans serif typeface one point smaller than the text of the clause in which it is contained.

In clause 3, the terms being defined appear in italics rather than in boldface (as per the agreed format for common text). Since ASN.1 terms appear throughout the Directory Specification in boldface, the use of italics is intended to clarify that these are defined terms and not ASN.1 types or values.

As these Directory Specifications are being constructed through the ISO/IEC amendment process, additional annexes which form an integral part of this Directory Specification may appear after annexes which are not an integral part of this Directory Specification.

This Directory Specification uses the term "1988 edition systems" to refer to systems conforming to the previous (1988) edition of the Directory Specifications Systems conforming to the current Directory Specifications are referred to as "1992 edition systems".

The last annex of this Directory Specification lists which amendments and technical corrigenda have been incorporated to form this edition of this Directory Specification.

6 Replication in the Directory

Replicated (copied) information can exist in the Directory. Caching and shadowing are two mechanisms for replication. Directory information can also be replicated by means outside this Specification. Any such alternative means of replication will need to ensure that exactly one instance of each replicated entry is identified as the master copy if the Directory and DSA Abstract Services are to be used.

Service controls provide the ability to control whether replicated information may be used in support of directory operations, regardless of the replication mechanism used to acquire the copy.

6.1 Caching

One method of replicating directory information is caching. Caching procedures are considered to be almost entirely governed by local policies, and therefore outside the scope of this Specification.

6.2 Shadowing

Another method of replicating directory information is shadowing. An overview of the Directory information shadow service is found in clause 7. Before shadowing can occur, the Administrative Authorities of the two DSAs involved must come to agreement on the terms under which the shadowing will take place. This negotiation is done prior to any protocol exchange. Components of the agreement are described in clause 9 of this Specification.

Once the terms of the agreement have been established, the DSAs may initiate, modify and subsequently terminate the agreement. This may be done through a shadow operational binding as described in clause 8 of this Specification.

This shadowing service for the Directory is based on the models established in CCITT Rec. X.501|ISO/IEC 9594-2, to satisfy the requirements outlined in CCITT Rec. X.500|ISO/IEC 9594-1. The protocol specification for shadowing and conformance requirements are provided in CCITT Rec. X.519|ISO/IEC 9594-5. In addition, this Specification provides the definition of an operational binding for the purpose of initiating and terminating shadowing agreements between DSAs. This operational binding type is defined using the tools specified in CCITT Rec. X.501|ISO/IEC 9594-2.

The directory information shadow service is described in clause 10 of this document. The actual shadowing occurs through the set of operations defined in clause 11. These operations accommodate the transfer of Directory information and updates to the shadowed information.

The use of shadowed information by a DSA to satisfy a Directory request is described in CCITT Rec. X.518|ISO/IEC 9594-4.

6.3 Shadowing functional model

In the standardized form of Directory replication, termed shadowing, a DSA may assume the role of *shadow supplier*, the source of shadowed information, or *shadow consumer*, the recipient of shadowed information. The role played by a DSA when engaging in standardized replication activities (shadow supplier or shadow consumer) is always with respect to another DSA which plays the reciprocal role (shadow consumer or shadow supplier).

A given DSA may assume both roles, either

- Ñ with respect to different DSAs for the same or different units of replication; or
- Ñ with respect to a single DSA (which plays the reciprocal role) for different units of replication.

The Shadowing functional model addresses two approaches to shadowing Directory information:

- Ñ a *primary shadowing* policy requires that each shadow consumer receives its updates directly from the master DSA for the unit of replication;
- Ñ a *secondary shadowing* policy permits a shadow consumer to assume the shadow supplier role with respect to shadow consumers not having a shadowing agreement directly with the master DSA.

The characteristics of these two policies and their approach to satisfying the requirements for levels of performance, availability, reliability and recovery that cannot otherwise be met are described below.

6.3.1 Primary shadowing

Figure 1 depicts primary shadowing. In this case the shadowing policy in effect has the following characteristics:

a) the master DSA is the only shadow supplier;

b) each shadow consumer has a direct shadowing agreement with the master DSA;

c) only read (including compare) and search (including list) operations may be performed at a shadow consumer holding shadowed information. All modification operations are directed to the master DSA.

Because it allows for the placement of copies of often requested information, or knowledge of it, closer to the requestor, this approach may be used to satisfy the performance requirement. Also, because this approach provides for the redundancy of individual entry or knowledge information it also is capable, in a primitive sense, of satisfying the availability, reliability, and recovery requirements.

T0712580-91

FIGURE 1

Primary shadowing

6.3.2 Secondary shadowing

Figure 2 depicts secondary shadowing. In this case the shadowing policy in effect has the following characteristics:

a) the master DSA is not the only shadow supplier. Only some shadow consumers have a direct shadowing agreement with the master DSA as their shadow supplier;

b) other shadow consumers may have a shadowing agreement with a shadow supplier that is not the master for the unit of replication. The shadowing agreements between the master DSA and its direct shadow consumers may, however, have an impact on secondary shadowing agreements;

c) only read (including compare) and search (including list) operations may be performed at a shadow consumer holding shadowed information. All modification operations are directed to the master DSA, whether directly (if a secondary shadow consumer DSA has knowledge of the master DSA) or indirectly via the shadow supplier DSA(s);

Secondary shadowing is very similar to primary shadowing in the way it addresses the performance, availability, reliability and recovery requirements. It differs in that it relieves the single master DSA of the burden of directly contacting all shadow consumers for the shadowed information. This is a desirable combination in environments where a large number of shadow consumers are holding the same shadowed information.

T0712590-91

FIGURE 2

Secondary shadowing

7 Overview of the Directory information shadow service

The directory information shadow service defined here provides the Directory with a standardized mechanism to provide and support replicated information. In outline, the shadow supplier maintains, for each shadowing agreement, information which is to be shadowed (the shadowed information). This information is replicated by protocol exchange between the shadow supplier and the shadow consumer. The information to be shadowed is all or a subset of the information held by the shadow supplier's DSA Information Tree. The shadow consumer's shadowed information becomes part of its DSA Information Tree.

To use the directory information shadow service, the Administrative Authorities of two DSAs must first reach an agreement on the terms under which shadowing will take place. This agreement, and the technical specification related to this agreement (the shadowing agreement), is discussed in 7.1. A description of the manner in which shadowed information is represented for the purposes of shadowing is provided in 7.2 below. The actual transfer of this shadowed information from the shadow supplier to the shadow consumer is accomplished by means of a set of shadow operations, which are introduced in 7.3.

The use of shadowed information to satisfy Directory requests is described in CCITT Rec.ÊX.518 ISO/IEC 9594-4.

7.1 Shadowing agreement

The agreement that is reached between the Administrative Authorities of two DSAs forms the technical and policy basis for the provision of the directory information shadow service. This agreement may include any set of terms acceptable on a bilateral basis to the two Administrative Authorities. For example, the agreement may specify policy information related to security, charging, or special conditions such as the deletion of shadowed information upon termination of the agreement.

Certain aspects of the shadowing agreement must be specified, including an identifier by which the shadowing agreement will be referenced, a specification of the unit of replication, information related to the update mode, and optionally, the access point of the master DSA for the shadowed information. Some technical aspects of the shadowing agreement may be exchanged via protocol and are discussed in detail in clause 9. Access control information is always supplied and need not be explicitly specified in the shadowing agreement. Initially the shadowing agreement is created by an off-line administrative process. It represents essentially a template whose technical parameter values are subsequently validated during the initiating phase of the agreement and possibly modified during modification operations on the agreement. The method of storing this agreement is beyond the scope of this Specification.

It must be noted that although the shadowing agreement will normally provide a true representation of the technical parameters related to the directory information shadow service, there may be exceptional cases in which policy overrides the technical specification resulting in a service inconsistency. For example, there may be certain attributes or attribute values that must

be withheld for security reasons. It may be the case that security policy prevents disclosing the mere existence of these attributes, in which case it would be a violation to represent in the shadowing agreement the fact that they are being withheld. In this type of situation, the behaviour of the shadow supplier DSA will be as if the technical specification were a true representation. Thus, users with access to the sensitive data will receive different views of the affected entries, depending on whether they access the master or a shadow consumer.

7.2 Shadowed information

Shadowed information is the logical set of information which is replicated by the shadow consumer. The three components of shadowed information are:

a) *area information* : Information about DSEs whose names fall within the replicated area;

b) *prefix information* : Information relevant to entries within the replicated area which, with respect to the DSA information model, are positioned between the area prefix and the root DSE. This may contain administrative point and subentry information;

c) *subordinate information* : Information about knowledge references subordinate to the replicated area.

Figure 3 illustrates the derivation of shadowed information.

FIGURE 3

Shadow supplier derivation of shadowed information

As illustrated at the left of figure 3, the replicated area must be fully contained within a single naming context. The unit of replication specifies the replicated area and any requirements pertinent to the shadowing of subordinate knowledge references. The specification of the unit of replication may extend beyond the naming context, however the replicated area is limited to the naming context. From this unit of replication specification, the shadow supplier can derive a representation of the shadowed information, which, as shown at the right of the figure, includes the prefix information, the area information (representing information held by DSEs in the replicated area), and (optionally) subordinate information. This shadowed information is subsequently conveyed by protocol to the shadow consumer which then integrates the information into its own DSA information tree. The shadowed information is built out of shadowed DSEs (SDSEs), which are discussed in 7.2.1. The establishment of shadowed information is discussed in 7.2.2.

Figure 4 illustrates the derivation of shadowed information where extended knowledge is included.

T0212610-91

FIGURE 4

Shadow supplier derivation of shadowed information with extended knowledge

7.2.1 SDSEs

Shadowed DSE (SDSE) : That information within shadowed information that is associated with a specific name. It is the information to be replicated from a DSE. It is distinct from this DSE, and is therefore not part of the DSA Information Model.

The definition and transfer of SDSEs is not intended to impose any obligation to have a distinct representation of SDSEs in any particular implementation.

An SDSE is analogous to a DSE and consists of:

- SDSE type (always);
- user attributes (derived from entry information for DSEs corresponding to entries that are to be shadowed);
- operational attributes (present as required);
- subordinate-completeness flag (for area and subordinate information only);
- attribute-completeness flag (present for area information only).

7.2.1.1 SDSE type

DSE types are defined in CCITT Rec. X.501|ISO/IEC 9594-2. SDSE type, as specified in 11.3.1.1, is analogous to DSE type, but has fewer options; glue, cp, entry, alias, subr, nssr, admPoint, subEntry and as.

7.2.1.2 Subordinate-completeness flag

The subordinate-completeness flag is a boolean that is present for SDSEs within the area information and subordinate information. The flag has the following semantics:

The flag is TRUE if and only if one of the following conditions is met for a particular SDSE:

a) it represents a leaf entry;

b) it contains SDSEs for each subordinate entry or each subordinate or non-specific subordinate reference known to the master DSA.

The flag is FALSE if one of the following conditions is met for a particular SDSE:

a) the subordinates known to the master for that particular SDSE are not all present in the shadow information;

b) it is not determined whether the shadowed information contains SDSEs for all subordinates known to the master DSA;

c) the shadow supplier does not intend to provide information about subordinate completeness, the default value FALSE is used for each SDSE.

7.2.1.3 Attribute-completeness flag

The attribute completeness flag is a boolean and is TRUE if and only if all user attributes of the entry and all relevant collective attributes are present for the SDSE. It is only present for SDSEs containing entry information.

The attribute-completeness flag is not used with respect to directory operational attributes; it is always assumed that they are not all present in the SDSE.

7.2.2 Establishment of shadowed information

The shadowed information represents three basic types of information: prefix information, area information, and subordinate information. Each of these is discussed in the following subclauses.

7.2.2.1 Prefix information

If the replicated area does not start immediately below the root of the DIT, the shadowed information will include SDSEs for each entry that is part of the area prefix of the replicated area (the path down from the root of the DIT to the replication base entry). Each of these SDSEs will be of type glue and will only represent the RDN of the entry, unless it represents an administrative point containing administrative policy information. There is an empty SDSE for the root DSE. There are no subordinate-completeness flags in area prefix SDSEs.

7.2.2.1.1 Administrative point

If the entry is an administrative point that has attributes pertaining to the replicated area, or that has one or more associated subentries whose subtree scope includes some or all of the replicated area, the SDSE is of type admPoint instead of type glue. Any attributes that are relevant for the replicated area are included in the SDSE. The administrativeRole attribute shall be included in all administrative point SDSEs which are relevant to the shadowed information.

7.2.2.1.2 Subentries

For subentries below the administrative point for which the subtree scope includes some or all of the replicated area, SDSEs of type subEntry may be included in the shadowed information. If the subtree scope of such a subentry does not include the replicated area or parts of it, no SDSE for this subentry need be included. Collective attributes, schema and access control information selected for the area information are represented in SDSEs of type subEntry.

7.2.2.2 Area information

All entries in the shadow supplier information tree that are included in the replicated area are represented in the shadowed information as SDSEs of type entry (unless removed by filtering). This SDSE contains the attributes of the entry as selected by the attribute selection of the shadowing agreement. Collective attributes held in subentries are selected in the same manner as other attributes and are represented in SDSEs of type subEntry. If any attributes of an entry have been selected for inclusion in the shadow, the objectClass attribute and the relevant entry access control information will be included in the SDSE for that entry. The attribute-completeness flag is set to indicate whether all user attributes in the DSE and all relevant collective attributes are present for the SDSE. The collectiveExclusions operational attribute, if present, is always included in the SDSE. It is always assumed that all directory operational attributes are not present, thus no corresponding flag is provided.

If the DSE is of type admPoint, the corresponding SDSE is of additional type admPoint and SDSEs of type subEntry for all relevant subentries immediately subordinate to the administrative point DSE are included in the shadowed information. The rules for inclusion of subentries are stated in 7.2.2.1.2.

If the DSE is of type cp, the corresponding SDSE is of additional type cp.

If filtering has been applied to the replicated area the resulting shadowed information may no longer be contiguous. There may be entries that have been removed by filtering that will create "holes" in the tree structure of the shadowed information. For each entry that has been removed by filtering the following rules are applied:

a) if there are SDSEs subordinate to that entry within the shadowed information that are not filtered out, an SDSE for the removed entry of type glue is added to the shadowed information. If the shadow supplier DSA supplies subordinate completeness information, then the subordinate completeness flag is set to TRUE if there are no other SDSEs subordinate to that entry that have been filtered out, and to FALSE otherwise. As this SDSE contains no entry information, it has no attribute completeness flag;

b) if there are no other SDSEs subordinate to the entry within the shadowed information, the subordinate-completeness flag of the SDSE for the entry immediately superior to the removed entry is set to FALSE and the SDSE for the removed entry is excluded from the shadowed information.

c) if the DSE is of type admPoint, it is always shadowed and the administrativeRole attribute is included.

Each SDSE in area information has a subordinate-completeness flag. If all of the subordinates of a given entry or a reference to them as known by the master DSA are included in the shadowed information, the subordinate-completeness flag of the SDSE is set to TRUE. Otherwise it is set to FALSE.

7.2.2.3 Subordinate information

The type of subordinate information required (i.e., master access points, shadow access points, or both; and whether extended knowledge is to be included or not) is specified in the shadowing agreement.

If subordinate knowledge is requested, subordinate references directly below the replicated area (master, shadow, or both types of knowledge as appropriate) are included as SDSEs of type subr or nssr as applicable, complete with the appropriate knowledge.

If extended knowledge is specified, subordinate references below (but not immediately subordinate to) the replicated area (master, shadow, or both) are included as SDSEs of type subr or nssr, complete with the appropriate knowledge. Subordinate glue SDSEs must be inserted to maintain connection with SDSEs in the replicated area. This may create glue SDSEs which are either within or below the replicated area. No other glue SDSEs are provided to support subordinate information.

subr and nssr SDSEs carry a subordinate completeness flag. glue SDSEs added for the purpose of extended knowledge carry no subordinate completeness flag and are always assumed to be incomplete (with respect to subordinate knowledge).

More detailed information on the unit of replication and the representation of shadowed information is contained in 9.2.

7.3 Shadow operations

Shadowed information is transmitted from the shadow supplier to the shadow consumer by using directory shadow operations. These operations provide for two fundamentally different models for updating shadowed information: shadow supplier-initiated shadowing (a "push" model), and shadow consumer-initiated shadowing (a "pull" model). These models are described more fully in clause 10. In either model, the information transmitted by protocol takes one of two forms *total* (the complete set of information within the shadowed information is transmitted) or *incremental* (only changes to the shadowed information are transmitted). Each element in total is an SDSE. Each element of incremental is an SDSE change. SDSE changes reflect the net effect of changes that have been made to the corresponding DSEs in the replicated area since the previous update, whether these changes originally occurred as a result of changes to individual DSEs (adds, deletes, etc.) or as a result of changes to multiple DSEs (e.g., resulting from a ModifyDN operation).

Three shadow operations have been defined. The CoordinateShadowUpdate operation is used in the push model to enable the shadow supplier to indicate the shadowing agreement for which it intends to send an update, to indicate the time the last update was sent for that agreement, and to indicate the intended update strategy (e.g., total or incremental). If a positive result is received in response to a CoordinateShadowUpdate operation, the shadow supplier uses the UpdateShadow operation to convey the shadowed information or the changes in the shadowed information, as indicated by the update strategy. For the pull model, the shadow consumer uses a RequestShadowUpdate operation to indicate the shadowing agreement for which it wishes to receive an update, to indicate the time supplied in the last update for that agreement, and to indicate the desired update strategy. If the parameters of the RequestShadowUpdate operation are acceptable to the shadow supplier a positive result is sent to the shadow consumer. The shadow supplier uses the UpdateShadow operation to convey the shadowed information or the changes to the shadowed information, as indicated by the update strategy. These operations are described in detail in clause 11.

7.4 DSA Shadow Bind and DSA Shadow Unbind

The DSAShadowBind and DSAShadowUnbind operation, defined in 7.4.1 and 7.4.2 respectively, are used by a DSA at the beginning and the end of a particular period of providing shadow updates.

7.4.1 DSA Shadow Bind

A DSAShadowBind operation is used at the beginning of a period of prividing shadows.

```
DSAShadowBind ::= BIND
        ARGUMENT    DirectoryBindArgument
        RESULT      DirectoryBindResult
        BIND-ERROR  DirectoryBindError
```

The components of the DSAShadowBind are identical to their counterparts in the DirectoryBind (see CCITT Rec. X.511|ISO/IEC 9594-3) with the following differences:

a) The Credentials of the DirectoryBindArgument allows information identifying the AE-Title of the initiating DSA to be sent to the responding DSA. The AE-Title must be in the form of a Directory Distinguished Name.

b) The Credentials of the DirectoryBindResult allows information identifying the AE-Title of the responding DSA to be sent to the initiating DSA. The AE-Title must be in the form of a Directory Distinguished Name.

7.4.2 DSA Shadow Unbind

A DSAShadowUnbind operation is used at the end of a period of providing shadows.

```
DSAShadowUnbind ::= UNBIND
```

8 Shadow operational binding

This clause defines the operational binding type for shadowing. It uses the elements and mechanisms of the DSA Operational Framework defined in CCITT Rec. X.501|ISO/IEC 9594-2.

The shadow operational binding type may be used to administer a shadowing agreement reached between the administrative authorities of two DSAs. Otherwise, the administration of such an agreement is outside the scope of this Specification. An instance of this operational binding type creates the environment in which shadow operations can be carried out between the two DSAs. Each instance is identified by an OperationalBindingID (AgreementID). The AgreementID is modified in a ModifyOperationalBinding operation.

8.1 Shadow operational binding type characteristics

8.1.1 Symmetry and roles

The shadow operational binding type is an asymmetrical type of operational binding. The two roles in a binding of this type are:

- the role of the shadow supplier (associated with the abstract role "A");
- the role of the shadow consumer (associated with the abstract role "B").

A detailed description of these roles is given in CCITT Rec. X.501|ISO/IEC 9594-2.

8.1.2 Agreement

The agreement that has to be exchanged during the establishment of the shadow operational binding or subsequent modifications is defined by the ASN.1 type ShadowingAgreementInfo defined in 9.1.

8.1.3 Initiator

The establishment, modification and termination of the shadow operational binding can be initiated by either the DSA with role shadow supplier (ROLE-A) or by the DSA with role shadow consumer (ROLE-B);

8.1.4 Establishment parameters

No additional parameters are transferred during the establishment of the binding.

8.1.5 Type identification

The shadow operational binding type is identified by the object identifier assigned by the application of the OPERATIONAL-BINDING-TYPE macro in 8.3.

8.2 DSA procedures for operational binding management

A set of operations has been defined for managing operational bindings. The use of these operations for management of a shadow operational binding is described in 8.2.1 to 8.2.3 below. These procedures apply to DSAs which support the directoryOperationalBindingManagementAC, as defined in CCITT Rec. X.519|ISO/IEC 9594-5. In the event of a protocol loss while initiating or terminating a shadowing agreement, the initiator shall assume the operation did not complete successfully. No specific action is required by the responder. Subsequent procedures are outside the scope of this Specification. Procedures for management of the shadow operational binding for DSAs which do not support the directoryOperationalBindingManagementAC are outside the scope of this Specification.

8.2.1 Establishment procedure

Once the bilateral agreement has been established (using procedures outside the scope of this Specification), it is activated with an EstablishOperationalBinding operation, as defined in CCITT Rec. X.501|ISO/IEC 9594-2. As arguments to this operation, the initiating DSA supplies the AgreementID for the instance of the binding, the role of the initiating DSA for this binding instance (shadow supplier or shadow consumer), and the ShadowingAgreementInfo.

AgreementID ::= OperationalBindingID

AgreementID identifies the shadowing agreement being established. It shall be unique between the pair of DSAs, and is used in subsequent operations to identify this agreement.

The values for the parameters in ShadowingAgreementInfo are simply accepted or rejected; there is no negotiation. The responding DSA does not have the option of returning a modified set of acceptable parameter values. Assuming a successful outcome of the request to establish an agreement, the shadow supplier and shadow consumer have the same information in their shadowing agreement.

If the EstablishOperationalBinding is successful, the shadowing agreement becomes active.

Errors returned in response to an EstablishOperationalBinding operation are interpreted according to the error description in CCITT Rec. X.501 | ISO/IEC 9594-2.

8.2.2 Modification procedure

Modification of the parameters of a shadowing agreement is agreed as part of the bilateral agreement between the two DSA Administrative Authorities. Modification to these parameters results in a new shadowing agreement being established. The parameters of the agreement may be exchanged using a ModifyOperationalBinding operation. The DSA Administrative Authorities should consider the effect of agreement modification on any secondary shadows prior to the modification operation as these secondary agreements may be required to be modified, updated or terminated.

The modification procedure does not allow modification to the replication base entry.

The arguments to the ModifyOperationalBinding are the AgreementID for this instance of the binding, the AgreementID for the binding after the operation has been applied, the role of the DSA for this binding instance (shadow supplier or shadow consumer), and the new ShadowingAgreementInfo. The values for the parameters of the ShadowingAgreementInfo for the modify operation are accepted or rejected; there is no negotiation. Assuming a successful outcome to the request for a modification of the agreement, the shadow consumer and shadow supplier have the same information in their shadowing agreement.

After the modification operation the data associated with the prior agreement remains in the shadow consumer and becomes the shadowed information for the new agreement. This does not preclude the shadow consumer requesting a total refresh. An update of the shadowed information may be required to remove inconsistencies between prior shadowed data and data required to be shadowed as specified in the UnitOfReplication associated with the new agreement.

Errors returned in response to a ModifyOperationalBinding operation are interpreted according to the error description in CCITT Rec. X.501 | ISO/IEC 9594-2.

8.2.3 Termination procedure

Termination of the operational binding deactivates the shadowing agreement. The termination is accomplished by either the shadow supplier or the shadow consumer initiating the TerminateOperationalBinding operation as specified in CCITT Rec. X.501|ISO/IEC 9594-2. Conditions may have been specified as part of the bilateral agreement regarding subsequent treatment of the data upon termination, such as the removal of the shadowed information from the shadow consumer DSA within a specified time. Such conditions take effect upon termination. In the event that a shadowing agreement is terminated, the shadow consumer shall terminate any secondary shadowing agreements dependent on information in the shadowing agreement in question. The termination of secondary shadowing agreements is decoupled from the original TerminateOperationalBinding operation.

If the TerminateOperationalBinding is successful, the shadowing agreement ceases.

Errors returned in response to a TerminateOperationalBinding operation are interpreted according to the error description in CCITT Rec. X.501 | ISO/IEC 9594-2.

8.2.4 Operations and procedures

The operations that can be executed in the active state of a shadow operational binding are those defined within the shadowConsumerInitiatedAC, shadowSupplierInitiatedAC, reliableShadowConsumerInitiatedAC and reliableShadowSupplierInitiatedAC application contexts defined in CCITT Rec. X.519|ISO/IEC 9594-5 :

- updateShadow operation
- requestShadowUpdate operation
- coordinateShadowUpdate operation

These operations are defined in clause 11. The associated service is defined in clauseÊ10.

8.3 Type definition

This subclause defines the shadow operational binding type using the OPERATIONAL-BINDING-TYPE macro defined in CCITT Rec. X.501|ISO/IEC 9594-2.

```
shadowOperationalBinding  OPERATIONAL-BINDING-TYPE
    AGREEMENT   ShadowingAgreementInfo
    APPLICATION CONTEXTS
        ShadowSupplierInitiatedAC APPLIES TO ALL OPERATIONS
        ShadowConsumerInitiatedAC APPLIES TO ALL OPERATIONS
        ReliableShadowSupplierInitiatedAC APPLIES TO ALL OPERATIONS
        ReliableShadowConsumerInitiatedAC APPLIES TO ALL OPERATIONS
    ASYMMETRIC
        ROLE-A          -- shadow supplier role
            ESTABLISHMENT-INITIATOR
            MODIFICATION-INITIATOR
            TERMINATION-INITIATOR
        ROLE-B          -- shadow consumer role
            ESTABLISHMENT-INITIATOR
            MODIFICATION-INITIATOR
            TERMINATION-INITIATOR
    ::= { operationalBinding 1}
```

The type ShadowingAgreementInfo is defined in 9.1.

9 Shadowing agreement

Before shadowing takes place between two DSAs, a bilateral agreement is established by an offline administrative process. Such an agreement may be explicitly established by the Administrative Authorities of the two DSAs for each individual shadowing agreement. Typically such explicit bilateral agreements may be required where the two DSAs are in different Directory management domains. Alternatively a general agreement may be established which implicitly covers all shadowing agreements among a set of DSAs. Typically such implicit bilateral agreements may be applicable to shadowing agreements between pairs of DSAs within a single Directory management domain.

In addition to the parameters of a shadowing agreement (see below) this bilateral agreement may include policy conditions for the treatment of the data upon termination of the agreement, such as for the removal of shadowed information upon termination (or modification) of the shadowing agreement itself.. Administrative Authorities also need to consider factors affecting interoperability, such as use of RTSE or ROSE ASEs, when establishing agreements.

A shadowing agreement is required before shadowed information may be shared between any pair of DSAs. This establishes the technical parameters of the agreement, specifying update frequency, replicated area and information to be shadowed. This shadowing agreement may be a representation of some of the technical components of the bilateral agreement between the two DSA Administrative Authorities.

The shadowing agreement may be activated by its inclusion in an EstablishOperationalBinding operation (as outlined in 8.2.1) or by means outside the scope of this Specification. In addition a shadowing agreement may be modified through a ModifyOperationalBinding operation (as outlined in 8.2.2). No negotiation of parameters of the agreement is supported by the operational binding management protocol. The parameters are either accepted or rejected.

9.1 Shadowing agreement specification

The shadowing agreement is specified as:

```
ShadowingAgreement ::=          ShadowingAgreementInfo

ShadowingAgreementInfo ::= SEQUENCE {
    shadowSubject       UnitOfReplication,
    updateMode          UpdateMode,
    master              AccessPoint  OPTIONAL,
    completeSubentry            BOOLEAN DEFAULT TRUE }
```

shadowSubject specifies the subtree, entries and attributes to shadow. The components of the UnitOfReplication are defined in 9.2.

updateMode specifies when updates of a shadowed area are scheduled to occur. The components of UpdateMode are defined in 9.3.

master contains the access point of the DSA containing the mastered area. This element is optional and need only be supplied for optimization purposes.

completeSubentry, if set to TRUE, indicates that the shadow supplier shall always supply a complete copy for each subentry which has changed within the shadowed information. If a subentry within the shadowed information requires updating, the shadow supplier provides a total replacement of the changed subentry employing the replace option within ContentChange of SDSEChanges.

9.2 Unit of replication

This subclause describes how portions of the DIT can be replicated, by defining the granularity of the DIT information that can be shadowed. The unit of replication is defined within the Directory information model and a specification mechanism is provided. The shadowing mechanism in the Directory is based on the definition of the subset of the DIT that will be shadowed. This subset is called *unit of replication*.

Because shadowing in the Directory is only defined between pairs of DSAs, there is a constraint that the shadowed information shall be completely within a single DSA. The specification of the unit of replication may extend beyond a naming context, but the replicated area is limited to the naming context.

The unit of replication comprises a three-part specification which defines the scope of the DIT to be replicated, the attributes to be replicated within that scope, and the requirements for subordinate knowledge. The unit of replication also implicitly causes the shadowed information to include policy information in the form of operational attributes held in entries and subentries (e.g., access control information) which is to be used to govern access to the shadowed information. The information to be included begins at an autonomous administrative point and extends to the replication base entry.

The unit of replication is specified as:

```
UnitOfReplication  ::=     SEQUENCE       {
      area                      AreaSpecification,
      attributes                AttributeSelection,
      knowledge                 Knowledge OPTIONAL }

AreaSpecification ::= SEQUENCE {
      contextPrefix             DistinguishedName,
      replicationArea           SubtreeSpecification }

Knowledge  ::=    SEQUENCE {
      knowledgeType             ENUMERATED {
                                          master    (0),
                                          shadow    (1),
                                          both      (2) },
extendedKnowledge               BOOLEAN DEFAULT FALSE }
```

area defines the replicated area. It includes the context prefix of the naming context for the replicated area and the subtree specification relative to that context prefix. SubtreeSpecification is defined in CCITT Rec. X.501|ISO/IEC 9594-2.

attributes defines the set of attributes to be shadowed. It includes specification of user attributes (including collective attributes) and operational attributes, as described in 9.2.2.

knowledge defines the knowledge references to be shadowed. It includes specification of the type of references (master/shadow) to be shadowed as well as whether the knowledge requested is extended knowledge.

master indicates that only references to master naming contexts are to be supplied.

shadow indicates that only references to shadowed naming contexts are to be supplied.

both indicates that references to both master and shadowed naming contexts are to be supplied.

If extendedKnowledge is specified, then all subordinate and non-specific subordinate references of the naming context, which are subordinate to the area prefix are included in the unit of replication. To achieve this glue SDSEs are included in the shadowed area to represent all entries between the lower boundary of the replicated area and the subordinate knowledge references.

The following subclauses define the components of the unit of replication in detail. Support, by a shadow supplier DSA, for various components is optional as specified in 9.3.1 of X.519 9594-5.

9.2.1 Area specification

The replicated area is specified by defining a subtree of the DIT and refining that subtree to exclude those portions not required. The refinements include a filtering of entries and subentries, based on their object or subentry class. These stages are described in 9.2.1.1 and 9.2.1.2.

9.2.1.1 Subtree boundary specification

The first stage is to specify the shape of the subtree that is to be shadowed within a DSA. This is done by drawing the boundary of the subtree based on the tree structure using the subtree specification mechanism as defined in CCITT Rec. X.501|ISO/IEC 9594-2. The component base of SubtreeSpecification is used to provide the replication base entry of the unit of replication relative to an appropriate context prefix. The chop component of SubtreeSpecification is used to define the lower boundary of the subtree that is to be shadowed. The entries that can be referenced by either the exclusions or the maximum component are limited to the lower boundary of the naming context holding the replication base entry. If the chop component is absent, the unit of replication includes the whole subtree starting with the base and proceeding down to the lower boundary of the naming context.

9.2.1.2 Subtree refinement

The next stage of refinement is to apply a filter to the selected subtree. The specification-filter component of SubtreeSpecification is used to specify the filter. Filtering is done on object class or subentry class (only).

Filtering may result in a unit of replication that is no longer a connected subtree in the DSA, from the viewpoint of the Directory Information Model. For such subtrees glue DSEs are required to be supplied for as many entries as are needed to build a connected subtree in the shadow consumer.

9.2.2 Attribute selection

This further stage of refinement of the unit of replication specifies the attributes (user, collective and Directory operational) to be shadowed.

In addition to those specified here, access control operational attributes and modification timestamps are always included in a unit of replication. Also, if knowledge is specified (as defined in 9.2.3) the knowledge operational attributes will be included in the shadowed information and need not be enumerated as part of this attribute selection.

The attribute selection shall be specified to reflect, if at all possible, any restrictions on shadow consumer access to the information. However it is possible that some security policies may cause very limited exceptions to this norm where particular information is withheld from the shadowed information.

```
AttributeSelection  ::=  SET OF ClassAttributeSelection

ClassAttributeSelection  ::=  SEQUENCE {
     class                      OBJECT IDENTIFIER OPTIONAL,
     classAttributes       ClassAttributes }

ClassAttributes  ::= CHOICE {
     allAttributes          NULL,
     include    [0]         AttributeTypes,
     exclude    [1]         AttributeTypes
                            }          DEFAULT allAttributes  NULL

AttributeTypes ::=          SET OF AttributeType
```

Each element of AttributeSelection is a ClassAttributeSelection, and specifies the attributes for objects of the indicated class. The class may be an object class (for entries) or a subentry class. Specification of attributes for an object superclass also applies to any subclasses of the named class. If the class is omitted, the selection applies to all classes.

The default allAttributes specifies that all user attributes (including collective attributes) are to be included. If there are relevant collective attributes associated with the class, the appropriate collectiveAttributeArea subentries are implicitly included. If any directory operational attributes (other than access control, timestamps and knowledge) are to be included, they must be identified in the include element of the specification.

Attributes are implicitly included in the case where allAttributes is specified. In addition, when using the exclude specification, any attributes contained in an entry which are not explicitly excluded are implicitly included. The specification of an attribute supertype implicitly includes any subtypes of that attribute.

Explicit include or exclude of a collective attribute for a particular class results in the corresponding inclusion or exclusion of the corresponding collectiveAttributeArea subentries.

Where entries belong to more than one of the specified classes, the specifications are cumulative. In the case of conflicting specifications include has priority over explicitly excluded attributes and exclude has priority over implicitly included attributes.

9.2.3 Subordinate knowledge

The next stage in defining the unit of replication is the inclusion of subordinate knowledge. This knowledge may include subordinate knowledge of either master or shadowed naming contexts and may include specific or non-specific references. Additionally, such subordinate knowledge references may be included in the unit of replication, even if they are not immediately subordinate to entries in the replicated area, in which case they are referred to as extendedKnowledge references.

9.2.4 Subentries

Subentries are included in the unit of replication for access control, schema and collective entries as described below.

9.2.4.1 Access control information

It is the responsibility of the shadow supplier to provide properly transformed access control information for each item in the unit of replication. The nature of the transformation is specified as part of the shadowing agreement and may be as simple as the identity transformation.

NoteÑFor example, the transformation may reflect a local policy that states it is not necessary to shadow permission related to controlling modification of the shadowed items. Such a policy is consistent with the read-only nature of shadowed information.

The following access control information shall always be shadowed:

a) prescriptive access controls relevant to the reading of the replicated information and found in access-control-specific or inner points within the replicated area up to and including the first access-control-specific or autonomous administrative point encountered proceeding from the area prefix towards the root;

b) entry access controls relevant to the reading of each entry shadowed.

The shadow consumer shall enforce access control using the shadowed access control information.

NoteÑIt is desirable that changes in access control policy, as expressed by prescriptive ACI, should be propagated to shadowing DSAs (and other DSAs) as soon as possible. Such changes may cause (for example) the initiation of a (normal) incremental refresh exchange to affected DSAs, without regard for any particular periodic strategy. The refresh would include (for consistency) any other updates pending to the unit of replication. A similar consideration may apply to group-of-unique-name updates when these updates affect access control.

9.2.4.2 Schema information

The schema information required by a shadow consumer to accommodate the shadowed information in its DSA Information Tree and to satisfy Directory query operations on that shadowed information, needs to be shadowed as part of the unit of replication.

The relevant operational attributes of the subschema subentry are specified for that class in the AttributeSelection component of the UnitOfReplication specification.

9.2.4.3 Entry collection information

Collective attributes are included in/excluded from the unit of replication as user attributes. If allAttributes is specified, all corresponding collectiveAttributeArea subentries are implicitly included in the unit of replication. If user attributes explicitly included in the unit of replication are collective attributes the corresponding attributes of the collectiveAttributeArea are included in the unit of replication.

9.2.5 Overlapping replicated areas

A shadow consumer may optionally be involved in two or more shadowing agreements specifying overlapping replicated areas. The procedures to be followed by DSAs that do not support overlapping replicated areas are defined in 9.2.5.1. The procedures to be followed by DSAs that support overlapping replicated areas are defined in 9.2.5.2.

To support overlapping replicated areas and overlapping subentry information in the shadow consumer the createTimestamp and modifyTimestamp shall be provided by the shadow supplier in the shadow information (entries and subentries). The createTimestamp shall be conveyed in the SDSEContent during a total refresh or if a new shadow DSE is added. The modifyTimestamp shall always be conveyed in the SDSEContent if present in the shadow supplierÕs DSE for that entry or subentry.

9.2.5.1 Procedures for DSAs not supporting overlapping replicated areas

This clause defines the procedures to be followed by shadow consumers that do not support overlapping replicated areas.

A shadow consumer shall not engage in two or more shadowing agreements whose UnitOfReplication specifies overlapping replicated areas. However, the shadow consumer may encounter cases where non-overlapping replicated areas share the same area prefix, resulting in overlapping area prefix SDSEs. Thus, any subentry SDSEs within an area prefix may be subject to separate (uncoordinated) updates from different shadowing agreements. Changes in subentries (such as prescriptive access control information) need to be associated with particular data and updates reflecting such changes will only be sent for relevant shadowing agreements. Subentries for shadowing agreements which share an area prefix need to be logically maintained separately and associated with the appropriate replicated area.

9.2.5.2 Procedures for DSAs supporting overlapping replicated areas

This clause defines the procedures to be followed by shadow consumers that support overlapping replicated areas.

Each replicated area (associated with a shadowing agreement) shall be represented in the shadow consumer by a separate Òinformation plane.Ó When the shadowed information associated with a shadowing agreement is updated, only the Òinformation planeÓ that represents that shadowed information shall be affected.

When performing a Directory interrogation operation on a given replicated area, a shadow consumer shall do one of the following:

a) select an "information plane" that is capable of satisfying the specified Directory operation. The procedure used to select the appropriate Òinformation planeÓ is outside the scope of this Specification. Once an appropriate "information plane" is found, only the shadow DSEs contained in that "plane" are considered during the execution of the Directory operation, i.e., information contained in other "information planes" is ignored;

b) consider the aggregate of shadowed information the shadow consumer holds for the relevant replicated area by merging the shadow DSEs from different "information planes"into one single set of shadow DSEs, one for each replicated entry. If the resulting shadowed information is capable of satisfying the Directory operation, execute the latter on the resulting set of shadow DSEs.

NoteÑA shadow DSE resulting from the union of all shadow DSEs representing a given replicated entry should contain the most current shadowed information from the set of all applicable "information planes".

9.3 Update mode

The updateMode argument in the shadowing agreement specifies when updates are expected to occur for shadowed information.

```
UpdateMode ::=CHOICE    {
        supplierInitiated   [0]   SupplierUpdateMode,
        consumerInitiated       [1]       ConsumerUpdateMode    }

SupplierUpdateMode ::=      CHOICE       {
        onChange            BOOLEAN DEFAULT TRUE,
        scheduled           SchedulingParameters }

ConsumerUpdateMode    ::= SchedulingParameters
```

The components of updateMode are defined in 9.3.1 through 9.3.3.

For each shadowing agreement, a choice has to be made between the shadow supplier or shadow consumer initiating the update. This is specified by selecting supplierInitiated or consumerInitiated. This choice does not preclude either party in a shadowing agreement from initiating (or attempting to initiate) an update at times outside those specified by the UpdateMode.

9.3.1 Supplier update mode

In SupplierUpdateMode, onChange indicates that the shadow supplier is expected to provide updates when changes occur within the shadowed information as specified by the unit of replication. Should the shadow consumer be unavailable, the shadow supplier shall resend the update within an appropriate, locally-defined, time period. If, due to the unavailability of the shadow consumer, a number of changes are outstanding, the shadow supplier may transmit them within one single UpdateShadow operation.

scheduled allows updates from the shadow supplier to be scheduled as specified by SchedulingParameters.

9.3.2 Consumer update mode

In ConsumerUpdateMode the scheduling of the update requests is as specified by the SchedulingParameters.

9.3.3 Scheduling parameters

The SchedulingParameters provide the information required to schedule the requests for updates.

```
SchedulingParameters ::= SEQUENCE   {
    periodic   PeriodicStrategy     OPTIONAL,
        -- must be present if othertimes is set to FALSE --
    othertimes  BOOLEAN DEFAULT FALSE        }
```

Scheduling can be based on a periodic basis (periodic), an exception basis (othertimes) or a combination of both.

If present, periodic indicates that update windows are expected to occur on a regular basis. PeriodicStrategy is used to specify the windows by providing a start time of the first window, the size of each window and the amount of time between windows. These parameters provide guidance as to when updates are expected to occur, however updates may also be attempted, for a number of reasons, outside the windows specified.

```
PeriodicStrategy ::= SEQUENCE {
    beginTime                   Time OPTIONAL,
    windowSize                  INTEGER,
    frequencyOfUpdate           INTEGER          }

Time      ::=  GeneralizedTime  -- per 30.3 b) of Recommendation X.208|ISO 8824--
```

beginTime specifies the start time of the first window.

windowSize is the length of the update window in minutes.

frequencyOfUpdate is the period between update windows in minutes.

If beginTime is not specified, the update strategy starts at the time the shadowing agreement is activated.

othertimes indicates that updates can be scheduled according to local requirements. When this is set as part of the shadowing agreement, then the shadow supplier may include the updateWindow parameter during shadow update operations to signal the window for the next expected update.

If periodic is present and othertimes is TRUE the window selected via othertimes has precedence over those specified in PeriodicStrategy (e.g. if othertimes calls for a later time than the next periodic update according to the periodic strategy, the periodic update strategy time is ignored).

10 Directory information shadow service

The directory information shadow service defined here provides the Directory with a mechanism to provide and support replicated information. The use of shadowed information to satisfy Directory requests is described in CCITT Rec. X.518|ISO/IEC 9594-4.

Once a shadowing agreement has been activated, shadowing may take place in the form of updates by using abstract operations of the Directory Information Shadowing Protocol (DISP). Three distinct operations are available: CoordinateShadowUpdate, UpdateShadow, and RequestShadowUpdate. Descriptions of how these operations are used for shadow supplier initiated update and for shadow consumer initiated update are provided in 10.1 and 10.2 below. In both cases the updates for a particular agreement are sent in a single operation. The operations themselves are defined in clause 11 and the associated errors in clause 12.

10.1 Shadow supplier initiated service

This subclause describes shadow supplier initiated update using the CoordinateShadowUpdate and UpdateShadow operations. The CoordinateShadowUpdate operation, invoked by the shadow supplier, identifies the shadowing agreement for which the shadow supplier intends to send an update.

Upon receipt of a positive acknowledgement, the shadow supplier sends the update for the shadowing agreement by using the UpdateShadow operation.

Otherwise the shadow supplier responds with a ShadowError. The circumstances under which particular errors will be returned are defined in clause 12.

Although the CoordinateShadowUpdate operation applies to only a single shadowing agreement, several shadowing agreements can be updated within a single application association. For any one shadowing agreement, the CoordinateShadowUpdate operation (request and result) must precede the UpdateShadow operation. Only one instance of the UpdateShadow operation can be invoked per CoordinateShadowUpdate instance. For any one shadowing agreement, there can only be a single CoordinateShadowUpdate operation for which the response and UpdateShadow operation are outstanding at any one time.

Under certain circumstances, a failure of underlying services may be detected by the shadow supplier and/or shadow consumer (e.g., as a result of a RO-REJECT-P or provider abort). If such an indication is received at any point prior to receipt of a positive response to the UpdateShadow operation, the shadow supplier shall assume that the combination of CoordinateShadowUpdate and UpdateShadow failed . If the shadow consumer receives such an indication at any point prior to responding to the UpdateShadow operation, the shadow consumer shall also assume that the entire combination failed. Assuming such a failure the shadow consumer, upon receipt of another CoordinateShadowUpdate operation for this shadowing agreement shall disregard any previously outstanding CoordinateShadowUpdate rather than return an error. Procedures for recovery are outside the scope of this Specification.

10.2 Shadow consumer initiated service

This clause describes shadow consumer initiated update using the RequestShadowUpdate and UpdateShadow operations. The RequestShadowUpdate operation, invoked by the shadow consumer, identifies the shadowing agreement for which the shadow consumer wishes to receive an update.

If the parameters in the RequestShadowUpdateArgument are acceptable to the shadow supplier, a result will be returned although no information will be conveyed with it. The shadow supplier sends the update for the shadowing agreement using the UpdateShadow operation.

Otherwise the shadow supplier responds with a ShadowError. The circumstances under which particular errors will be returned are defined in clause 12.

Although the RequestShadowUpdate operation applies to only a single shadowing agreement, several shadowing agreements can be updated within a single application association. For any one shadowing agreement the RequestShadowUpdate operation (request and result) must precede the UpdateShadow operation. Only one instance of the UpdateShadow operation can be invoked per RequestShadowUpdate instance. For any one shadowing agreement there can only be a single RequestShadowUpdate operation for which the response and UpdateShadow operation are outstanding at any one time.

Under certain circumstances, a failure of underlying services may be detected by the shadow supplier and/or shadow consumer (e.g., as a result of a RO-REJECT-P or provider abort). If such an indication is received at any point prior to receipt of a positive response to theUpdateShadow operation, the shadow supplier shall assume that the combination of RequestShadowUpdate and UpdateShadow failed . If the shadow consumer receives such an indication at any point porior to responding to the UpdateShadow operation, the shadow consumer shall also assume that the entire combination failed. Assuming such a failure the shadow supplier, upon receipt of another RequestShadowUpdate operation for this shadowing agreement shall disregard any previously outstanding RequestShadowUpdate rather than return an error. Procedures for recovery are outside the scope of this Specification.

11 Shadow operations

The operations of the Directory Information Shadowing Protocol (DISP), used by shadow suppliers and shadow consumers to realize the directory information shadow service described in clause 10, are defined in 11.1 through 11.3. The associated errors are defined in clause 12.

11.1 Coordinate Shadow Update operation

The CoordinateShadowUpdate operation is used by the shadow supplier to indicate the shadowing agreement for which it intends to send updates.

```
CoordinateShadowUpdate ::= OPERATION
    ARGUMENT        CoordinateShadowUpdateArgument
    RESULT          CoordinateShadowUpdateResult
    ERRORS          ShadowError

CoordinateShadowUpdateArgument ::=  SEQUENCE {
    agreementID     AgreementID,
    lastUpdate      Time,
    updateStrategy  CHOICE {
                        standard ENUMERATED {
                                        noChanges       (0),
                                        incremental     (1),
                                        total           (2)  },
                        other   EXTERNAL  }  }

CoordinateShadowUpdateResult ::= NULL
```

11.1.1 Coordinate Shadow Update Parameters

The various parameters have the meanings defined below:

The agreementID argument identifies the shadowing agreement as defined in 9.1.

The lastUpdate argument indicates the shadow supplier's understanding of the time at which the last update for this agreement was sent and is the time as provided by the shadow supplier DSA.

The updateStrategy argument identifies the update strategy the shadow supplier intends to use for this update. Within the choice of standard, the shadow supplier may select noChanges (indicating no modifications to the shadowed information), incremental (indicating incremental changes), or total (indicating a complete replacement of the unit of replication).

The option noChanges should only be used when the shadow supplier wishes to inform the shadow consumer that no modifications have occurred to the replicated area since the last update (e.g. in the case where a regularly scheduled update is expected). This shall be followed by an UpdateShadow operation with RefreshInformation set to noRefresh.

11.1.2 Coordinate Shadow Update success

Should the request succeed, a result will be returned, although no information will be conveyed with it.

11.1.3 Coordinate Shadow Update failure

Should the request fail, a ShadowError shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An invalidAgreementID shadow problem is returned if the shadow consumer DSA does not recognize the AgreementID specified within the set of AgreementIDs with this shadow supplier DSA.

An inactiveAgreement shadow problem is returned if the shadow consumer DSA recognizes the AgreementID as a valid AgreementID for this shadow supplier DSA, but the shadow consumer DSA understands that the AgreementID is inactive.

An unsupportedStrategy shadow problem is returned if the shadow consumer DSA does not support the refresh strategy selected by the shadow supplier DSA for this shadowing agreement.

A missedPrevious shadow problem is returned if the shadow consumer DSAÕs understanding of the time of last update is earlier than the time indicated by the value received in lastUpdate.

A fullUpdateRequired shadow problem is returned by the shadow consumer DSA to inform the shadow supplier that a total refresh is required to bring the shadow consumer DSA into a state of consistency with the shadow supplier. This could be returned, for instance, if the shadow consumer DSA is recovering from a major failure and does not currently understand its state of consistency with respect to the shadow supplier.

An unwillingToPerform shadow problem is returned by the shadow consumer DSA to indicate that it is unwilling to perform the update operation associated with this coordinate operation. Interpretation of this shadow problem is outside the scope of this Specification.

An unsuitableTiming shadow problem is returned if the shadow consumer DSA is unwilling to perform the update associated with this operation at this time.

An updateAlreadyReceived shadow problem is returned if the shadow consumer DSAÕs understanding of the time of last update is later than the time indicated by the value received in lastUpdate.

The invalidInformationReceived shadow problem is not returned in response to this operation.

An invalidSequencing shadow problem is returned to signal the receopt of multiple consecutive CoordinateShadowUpdate requests for a single shadowing agreement without completing an intervening UpdateShadow operation or receiving an underlying service failure indication.

11.2 Request Shadow Update operation

A RequestShadowUpdate operation is used by the shadow consumer to request updates from the shadow supplier.

```
RequestShadowUpdate ::= OPERATION
    ARGUMENT        RequestShadowUpdateArgument
    RESULT          RequestShadowUpdateResult
    ERRORS          ShadowError

RequestShadowUpdateArgument ::= SEQUENCE {
    agreementID             AgreementID,
    lastUpdate              Time,
    requestedStrategy       CHOICE {
                              standard          ENUMERATED  {
                                                  incremental   (1),
                                                  total         (2) },
                              other             EXTERNAL } }

RequestShadowUpdateResult ::=  NULL
```

11.2.1 Request Shadow Update parameters

The various parameters have the meanings as defined below:

The agreementID argument identifies the shadowing agreement as defined in 9.1.

The lastUpdate argument is the time provided by the shadow supplier in the most recent successful update.

The requestedStrategy argument identifies the type of update being requested by the shadow consumer.

The shadow consumer may request either an incremental or a total update from the shadow supplier. However, if the shadow consumer requests an incremental update and the shadow supplier determines that it needs to send a total update, it will return a ShadowError with problem set to fullUpdateRequired.

11.2.2 Request Shadow Update success

Should the request succeed, a result will be returned although no information will be conveyed with it.

11.2.3 Request Shadow Update failure

Should the request fail, a ShadowError shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An invalidAgreementID shadow problem is returned if the shadow supplier DSA does not recognize the AgreementID specified within the set of AgreementIDs with this shadow consumer DSA.

An inactiveAgreement shadow problem is returned if the shadow supplier DSA recognizes the AgreementID as a valid AgreementID for this shadow consumer DSA, but the shadow supplier DSA understands that the AgreementID is inactive.

An unsupportedStrategy shadow problem is returned if the shadow supplier DSA does not support the refresh strategy selected by the shadow consumer DSA for this shadowing agreement.

A fullUpdateRequired shadow problem is returned by the shadow supplier DSA to inform the shadow consumer that a total refresh is required to bring the shadow consumer DSA into a state of consistency with the shadow supplier. This could be returned, for instance, if the shadow supplier DSA is unable to construct a meaningful incremental update with respect to the value received in lastUpdate.

An unwillingToPerform shadow problem is returned by the shadow supplier DSA to indicate that it is unwilling to perform the update operation associated with this request operation. Interpretation of this shadow problem is outside the scope of this Specification.

An unsuitableTiming shadow problem is returned if the shadow supplier DSA is unwilling to perform the update associated with this request operation at this time.

The invalidInformationReceived, missedPrevious, and updateAlreadyReceived shadow problems are not returned in response to this operation.

An invalidSequencing shadow problem is returned to signal the receipt of multiple consecutive RequestShadowUpdate requests for a single shadowing agreement without completing an intervening UpdateShadow operation or receiving an underlying service failure indication.

11.3 Update Shadow operation

An UpdateShadow operation is invoked by the shadow supplier to send updates to the shadow consumer for a unit of replication. Prior to this operation being initiated, a CoordinateShadowUpdate or a RequestShadowUpdate operation must have been successfully completed.

```
UpdateShadow ::= OPERATION
    ARGUMENT            UpdateShadowArgument
    RESULT              UpdateShadowResult
    ERRORS              ShadowError

UpdateShadowArgument ::= SEQUENCE {
    agreementID         AgreementID,
    updateTime              Time,
    updateWindow        UpdateWindow    OPTIONAL,
    updatedInfo             RefreshInformation }

UpdateShadowResult ::= NULL
```

11.3.1 Update Shadow Parameters

The various parameters have the meanings as defined below:

The agreementID identifies the shadowing agreement that has been established.

The updateTime argument is supplied by the shadow supplier. This time is used during the next CoordinateShadowUpdate or RequestShadowUpdate to ensure that the shadow supplier and shadow consumer have a common view of the shadowed information.

The updateWindow argument, when present, indicates the next window during which the shadow supplier expects to send an update. This parameter is only allowed if the SchedulingParameter of the UpdateMode of the shadowing agreement has the othertimes parameter set to TRUE.

```
UpdateWindow ::=  SEQUENCE {
    start           Time,
    stop            Time }
```

The updatedInfo argument provides the information required by the shadow consumer to update its shadowed information. This may be a total copy of the shadowed information or only incremental updates for a set of SDSEs. Although this need not provide a Òmirror imageÓ in the shadow consumer of the shadow supplierÕs information at any particular instant in time, the updates sent must be internally consistent for the replicated area.

The semantics of the information conveyed in this parameter shall result in the shadow consumer reflecting the changes supplied. Furthermore each update shall be applied independently and without regard to previously transmitted updates. If for instance, a particular add or delete was sent twice (in two separate updates with different update times), the shadow consumer would not signal an error, as the effect of adding the same shadow DSE twice in immediate succession is the same as adding it once. Similarly, deletin twice in immediate succession is the same as deleting once. However, neither would the shadow consumer disregard the second update on the basis of having received an earlier identical update, since intervening changes to the DSE (within the update window) could make the second update significant.

```
RefreshInformation ::= CHOICE {
    noRefresh                   NULL,
    total           [0]         TotalRefresh,
    incremental                 [1]         IncrementalRefresh,
    otherStrategy               EXTERNAL  }
```

noRefresh indicates that there have been no changes to the shadowed information from the previous instance to the present. This may be used where an UpdateShadow operation must be supplied at a certain intervals defined in the shadowing agreement (updateMode), but no modification has actually occurred.

total provides a new instance of the shadowed information.

incremental provides, instead of a complete replacement of the shadowed information, only the changes which have occurred to that shadowed information between lastUpdate in the most recent CoordinateShadowUpdate (or RequestShadowUpdate request), and updateTime in the current UpdateShadow request (or RequestShadowUpdate response).

otherStrategy provides the ability to send updates by mechanisms outside the scope of this Directory Specification.

NoteÑRefresh of a subtree of the replicated area can be accomplished through shadowing agreements with overlapping areas of replication.

11.3.1.1 Total refresh

The complete shadowed information is included starting at the root and including all SDSEs within the shadowed information.

```
TotalRefresh ::= SEQUENCE {
    sDSE                SDSEContent OPTIONAL,
    subtree     SET OF Subtree OPTIONAL }

SDSEContent ::= SEQUENCE {
    sDSEType            SDSEType,
    subComplete         BOOLEAN DEFAULT FALSE,
    attComplete                 BOOLEAN OPTIONAL,
                        SET OF Attribute }
```

```
SDSEType ::= BIT STRING {
                    glue        (1),    -- represents knowledge of a name --
                    cp          (2)     -- context prefix --
                    entry       (3),    -- object entry --
                    alias       (4),    -- alias entry --
                    subr        (5),    -- subordinate reference --
                    nssr        (6),    -- non-specific subordinate reference --
                    admPoint    (9),    -- administrative point --
                    subEntry    (10),   -- subentry --
                    immSupr     (13),   -- immediate superior reference
                    as          (15) }  -- subordinate alias --

    Subtree ::= SEQUENCE {
        RelativeDistinguishedName,
        COMPONENTS OF TotalRefresh }
```

Absence of objects (SDSEs) formerly contained in the shadowed information indicates their deletion.

Subtree is omitted for SDSEs which have no subordinate SDSEs.

subComplete is a boolean that, if present, indicates whether or not subordinate knowledge is complete. If TRUE, subordinate knowledge is complete. If FALSE, subordinate knowledge is incomplete or unknown.

attComplete is a boolean that, if present, indicates whether or not all user attributes are included. If TRUE, all user attributes are included. If FALSE, some user attributes have been omitted. If absent, it is undefined whether or not all user attributes are present.

11.3.1.2 Incremental refresh

Only the changes to the shadowed information are included in the IncrementalRefresh.

```
IncrementalRefresh ::= SEQUENCE  OF  {
    IncrementalStepRefresh      IncrementalStepRefresh }

IncrementalStepRefresh   ::= SEQUENCE {
    sDSEChanges     CHOICE {
        add         [0]     SDSEContent,
        remove              NULL,
        modify      [1]     ContentChange } OPTIONAL
    subordinateUpdates      SET OF SubordinateChanges OPTIONAL }

ContentChange ::= SEQUENCE {
    rename          RelativeDistinguishedName OPTIONAL,
    CHOICE {
        replace     [0]     SET OF Attribute,
        changes             SEQUENCE OF EntryModification
                                    } OPTIONAL,
    sDSEType                SDSEType,
     subComplete    [1]     BOOLEAN DEFAULT FALSE,
     attComplete    [2]     BOOLEAN OPTIONAL }

SubordinateChanges ::= SEQUENCE {
    RelativeDistinguishedName,
    COMPONENTS OF IncrementalRefresh }
```

The sequence of incrementalStepRefresh within the IncrementalRefresh shall be applied to the replicated area in the order supplied. This is required to support incremental updates in the case of reuse of a Distinguished Name.

incrementalStepRefresh specifies a group of changes to be applied to the replicated area.

SDSEChanges indicates changes which need to be reflected in the shadowed information.

add provides a copy of a complete SDSE. The shadow DSE in the shadow consumer has no subordinates. If a shadow DSE with this name already exists in the shadow consumer, any subordinates are deleted and the shadow DSE replaced.

remove indicates that this SDSE, and any subordinates to it, should not be represented by shadow DSEs in the shadow consumer.

modify includes those changes that need to be reflected in a particular SDSE.

rename is used to indicate changes to the distinguished value of one or more attributes which needs to be reflected in the SDSE. This does not affect the attribute types used for naming..

If the changes to the SDSE are extensive, a complete replacement of content is achieved using replace. Otherwise changes is used to indicate changes which need to be reflected in the SDSE.

If attComplete is absent this indicates that its value is undefined and it should not be included in the SDSE.

SubordinateChanges is used to create or modify subordinates. It is absent if there are no differences in the subordinate SDSEs.

11.3.2 Update Shadow success

Should the request succeed, a result will be returned, although no information will be conveyed with it.

11.3.3 Update Shadow failure

Should the request fail, a ShadowError shall be reported. Circumstances under which the particular shadow problems will be returned are defined below.

An invalidAgreementID shadow problem is returned if the shadow consumer DSA does not recognize the AgreementID specified within the list of AgreementIDs with this shadow supplier DSA.

An inactiveAgreement shadow problem is returned if the shadow consumer DSA recognizes the AgreementID as a valid AgreementID for this shadow supplier DSA, and if the shadow consumer DSA understands that the AgreementID is inactive.

An invalidInformationReceived shadow problem is returned if the shadow consumer DSA determines that, as a result of an error in the received data, it may not be able to use the received data to provide directory services to the directory users. As a general rule, extraneous data (e.g., entries that should have been filtered as a result of object class selection, attributes that should have been filtered out, etc.) is not considered sufficiently serious to require the return of this shadow problem as it can be ignored by the shadow consumer. Interpretation of this shadow problem is outside the scope of this Specification.

An unwillingToPerform shadow problem is returned by the shadow consumer DSA to indicate that the shadow consumer DSA is unwilling to perform this update operation. It may be returned, for example, to indicate that the APDU size exceeds local limits. Interpretation of this shadow problem is outside the scope of this Specification.

The unsupportedStrategy, missedPrevious, fullUpdateRequired, unsuitableTiming, and updateAlreadyReceived shadow problems are not returned in response to this operation.

An invalidSequencing shadow problem is returned to signal the receipt of an UpdateShadow operation for which there had been no prior CoordinateShadowUpdate or RequestShadowUpdate operation.

12 Shadow error

For any of the operations defined in clause 11 a ShadowError may be returned, the nature of the problem, and optionally the lastUpdate with a more suitable updateWindow.

```
ShadowError ::= ERROR
    PARAMETER SEQUENCE{
        problem             ShadowProblem,
        lastUpdate          Time OPTIONAL,
        updateWindow        UpdateWindow OPTIONAL}

ShadowProblem ::= INTEGER {
    invalidAgreementID          (1),
    inactiveAgreement           (2),
    invalidInformationReceived  (3),
    unsupportedStrategy         (4),
    missedPrevious              (5),
    fullUpdateRequired          (6),
    unwillingToPerform          (7) ,
    unsuitableTiming            (8),
    updateAlreadyReceived       (9),
    invalidSequencing                   (10) }
```

12.1 Shadow error problems

One of the following problems encountered is specified in problem:

a) invalidAgreementID: This DSA does not recognize the AgreementID specified within the list of AgreementIDs with that DSA;

b) inactiveAgreement: This error is returned when the agreement with this DSA exists but has not yet become active, or has become inactive but still exists;

c) invalidInformationReceived: This error indicates a serious problem with the shadow consumer DSA's understanding of the data received (i.e., the shadow consumer DSA is unable to use the data to provide directory services to directory users);

d) unsupportedStrategy: Indicates that the refresh strategy selected is not in the shadowing agreement or is not supported by this DSA;

e) missedPrevious: Indicates that the value received in lastUpdate is not consistent with the time the shadow consumer DSA understands was the time of the last update;

f) fullUpdateRequired: Indicates that the only acceptable strategy at this time (e.g., in the event of an otherwise unrecoverable mismatch of time stamps) is a full update;

g) unwillingToPerform: Indicates that the responder is unwilling to perform the requested operation. Interpretation of and action following receipt of this error is outside the scope of this Specification.

h) unsuitableTiming: Indicates that the responder is unwilling to handle the update or the generation of the update at this time.

i) updateAlreadyReceived: Indicates that the shadow consumer has already received the update associated with lastUpdate.

j) invalidSequencing: Indicates the receipt of shadow operations that are out of sequence.

12.2 Last update

If a missedPrevious error is reported by the shadow consumer the lastUpdate argument may be provided.. This allows the shadow supplier to determine whether it should send a total or incremental update. The means by which the shadow supplier reaches this decision is outside the scope of this Specification.

12.3 Update window

The updateWindow argument is (optionally) provided only if the shadow supplier is reporting an unsuitableTiming error. This is used by the shadow supplier to indicate the preferred window for the next attempt to refresh the shadow.

Annex A

ASN.1 for Directory shadow operations

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

This Annex includes all of the ASN.1 type, and value and definitions contained in this Specification in the form of the ASN.1 module DirectoryShadowAbstractService.

```
DirectoryShadowAbstractService {joint-iso-ccitt ds(5) modules(1) directoryShadowAbstractService(15) version (2) }
DEFINITIONS  ::=
BEGIN    IMPLICIT-TAGS

--    EXPORTS ALL The types and values defined in this module are being exported for use in
--    the other ASN.1 modules contained within the Directory Specifications, and for the use of
--    other applications which will use them to access Directory services.  Other applications may
--    use them for their own purposes but those uses will not constrain extensions and
--    modifications needed to maintain or improve the Directory service.--

IMPORTS
      BIND, UNBIND, OPERATION, ERROR
FROM Remote-Operation-Notation {joint-iso-ccitt remoteOperations(4) notation(0)}

      id-pt-shadow-update
FROM DirectoryShadowObjectIndentifiers directoryShadowObjectIdentifiers

      Attribute, ATTRIBUTE, ATTRIBUTE-SYNTAX, attributeSyntax,  AttributeType, Name,
RelativeDistinguishedName, DistinguishedName, SubtreeSpecification
FROM InformationFramework informationFramework

      DirectoryBind, Entry Modification
FROM DirectoryAbstractService directoryAbstractService

      AccessPoint
FROM DistributedOperations distributedOperations

      OPERATIONAL-BINDING-TYPE, OperationalBindingID
FROM DirectoryOperationalBindingManagementAbstractService
directoryOperationalBindingManagementAbstractService;
```

-- bind and unbind --

```
DSAShadowBind ::= BIND
      ARGUMENT          DirectoryBindArgument
      RESULT            DirectoryBindResult
      BIND-ERROR        DirectoryBindError

DSAShadowUnbind ::= UNBIND
```

```
-- shadow operational binding --

shadowOperationalBinding  OPERATIONAL-BINDING-TYPE
    AGREEMENT   ShadowingAgreementInfo
    APPLICATION CONTEXTS
        ShadowSupplierInitiatedAC APPLIES TO ALL OPERATIONS
        ShadowConsumerInitiatedAC APPLIES TO ALL OPERATIONS
        ReliableShadowSupplierInitiatedAC APPLIES TO ALL OPERATIONS
        ReliableShadowConsumerInitiatedAC APPLIES TO ALL OPERATIONS
    ASYMMETRIC
        ROLE-A          -- shadow supplier role
            ESTABLISHMENT-INITIATOR
            MODIFICATION-INITIATOR
            TERMINATION-INITIATOR
        ROLE-B          -- shadow consumer role
            ESTABLISHMENT-INITIATOR
            MODIFICATION-INITIATOR
            TERMINATION-INITIATOR
    ::= { operationalBinding 1}

AgreementID ::=     OperationalBindingID

ShadowingAgreement ::= ShadowingAgreementInfo

ShadowingAgreementInfo ::= SEQUENCE {
    shadowSubject           UnitOfReplication,
    updateMode              UpdateMode,
    master                  AccessPoint  OPTIONAL,
    completeSubentry            BOOLEAN DEFAULT TRUE }

UnitOfReplication           ::=SEQUENCE   {
    area                    AreaSpecification,
    attributes              AttributeSelection,
    knowledge               Knowledge OPTIONAL }

AreaSpecification ::= SEQUENCE {
    contextPrefix           DistinguishedName,
    replicationArea         SubtreeSpecification }

Knowledge  ::=   SEQUENCE {
    knowledgeType       ENUMERATED {
                            master   (0),
                            shadow  (1),
                            both      (2) },
    extendedKnowledge           BOOLEAN DEFAULT FALSE }

AttributeSelection  ::=  SET OF ClassAttributeSelection

ClassAttributeSelection  ::=  SEQUENCE {
    class                       OBJECT IDENTIFIER OPTIONAL,
    classAttributes         ClassAttributes }

ClassAttributes  ::= CHOICE {
    allAttributes                   NULL,
    include     [0]         AttributeTypes,
    exclude     [1]         AttributeTypes
                            }       DEFAULT allAttributes  NULL

AttributeTypes::=           SET OF AttributeType
```

```
UpdateMode  ::=CHOICE           {
    supplierInitiated     [0]       SupplierUpdateMode,
    consumerInitiated               [1]       ConsumerUpdateMode    }

SupplierUpdateMode ::=          CHOICE          {
    onChange            BOOLEAN DEFAULT TRUE,
    scheduled           SchedulingParameters    }

ConsumerUpdateMode    ::=       SchedulingParameters

SchedulingParameters ::= SEQUENCE {
    periodic   PeriodicStrategy     OPTIONAL,
        -- must be present if othertimes is set to FALSE --
    othertimes  BOOLEAN DEFAULT FALSE       }

PeriodicStrategy ::= SEQUENCE  {
    beginTime                   Time OPTIONAL,
    windowSize                          INTEGER,
    frequencyOfUpdate           INTEGER         }

Time      ::=  UTCTime

UpdateWindow ::= SEQUENCE {
    start         Time,
    stop          Time    }
```

-- shadow operations, arguments, and results --

```
CoordinateShadowUpdate ::= OPERATION
    ARGUMENT                    CoordinateShadowUpdateArgument
    RESULT              CoordinateShadowUpdateResult
    ERRORS              ShadowError

CoordinateShadowUpdateArgument ::=  SEQUENCE {
    agreementID         AgreementID,
    lastUpdate          Time,
    updateStrategy      CHOICE {
                            standardENUMERATED {
                                        noChanges       (0),
                                        incremental     (1),
                                        total           (2)  },
                            other     EXTERNAL  }  }

CoordinateShadowUpdateResult ::= NULL

UpdateShadow ::= OPERATION
    ARGUMENT                    UpdateShadowArgument
    RESULT              UpdateShadowResult
    ERRORS              ShadowError

UpdateShadowArgument ::= SEQUENCE {
    agreementID         AgreementID,
    updateTime                  Time,
    updateWindow        UpdateWindow    OPTIONAL,
    updatedInfo                 RefreshInformation }

UpdateShadowResult ::= NULL
```

```
RefreshInformation ::= CHOICE {
    noRefresh           NULL,
    total       [0]     TotalRefresh,
    incremental         [1]     IncrementalRefresh,
    otherStrategy       EXTERNAL  }

TotalRefresh ::= SEQUENCE {
    sDSE                SDSEContent OPTIONAL,
    subtree     SET OF Subtree OPTIONAL }

SDSEContent ::= SEQUENCE {
    sDSEType            SDSEType,
    subComplete         BOOLEAN DEFAULT FALSE,
    attComplete                 BOOLEAN OPTIONAL,
                        SET OF Attribute }

SDSEType  ::=  BIT STRING  {
                                glue            (1),    -- represents knowledge of a name --
                                cp              (2)     -- context prefix --
                                entry           (3),    -- object entry --
                                alias           (4(     -- alias entry --
                                subr            (5),    -- subordinate reference --
                                nssr            (6),    -- non-specific subordinate reference --
                                admPoint        (9),    -- administrative point --
                                subEntry        (10),   -- subentry --
                    i           immSupr         (13),   -- immediate superior reference
                                        as              (15) }      -- subordinate alias --

Subtree ::= SEQUENCE {
    RelativeDistinguishedName,
    COMPONENTS OF TotalRefresh }

IncrementalRefresh ::= SEQUENCE  OF{
    incrementalStepRefresh      IncrementalStepRefresh }

IncrementalStepRefresh  ::=     SEQUENCE {
    sDSEChanges         CHOICE {
        add             [0]     SDSEContent,
        remove                  NULL,
        modify          [1]     ContentChange } OPTIONAL
    subordinateUpdates          SET OF          SubordinateChanges OPTIONAL }

ContentChange ::= SEQUENCE {
    rename              RelativeDistinguishedName OPTIONAL,
    CHOICE {
        replace         [0]     SET OF Attribute,
        changes                 SEQUENCE OF EntryModification
                                        } OPTIONAL,
    sDSEType                    SDSEType,
     subComplete        [1]     BOOLEAN DEFAULT FALSE,
     attComplete        [2]     BOOLEAN OPTIONAL }

SubordinateChanges ::= SEQUENCE {
    RelativeDistinguishedName,
    COMPONENTS OF IncrementalRefresh }

RequestShadowUpdate ::= OPERATION
    ARGUMENT                    RequestShadowUpdateArgument
    RESULT          RequestShadowUpdateResult
    ERRORS          ShadowError
```

```
RequestShadowUpdateArgument ::= SEQUENCE {
    agreementID          AgreementID,
    lastUpdate           Time,
    requestedStrategy          CHOICE {
                               standard        ENUMERATED   {
                                                   incremental      (1),
                                                   total            (2) },
                               other           EXTERNAL } }

RequestShadowUpdateResult ::=  NULL
```

- *errors* --

```
ShadowError ::= ERROR
    PARAMETER SEQUENCE {
        problem                ShadowProblem,
        lastUpdate             Time OPTIONAL,
        updateWindow           UpdateWindow OPTIONAL}

ShadowProblem ::= INTEGER {
    invalidAgreementID         (1),
    inactiveAgreement          (2),
    invalidInformationReceived (3),
    unsupportedStrategy        (4),
    missedPrevious             (5),
    fullUpdateRequired         (6),
    noInformation              (7),
    unwillingToPerform         (8) ,
    unsuitableTiming           (9),
    updateAlreadyReceived      (10)
    invalidSequencing                  (11) }

END
```