The drawings contained in this Recommendation have been done in AUTOCAD

Recommendation X.519

THE DIRECTORY - PROTOCOL SPECIFICATIONS [1)]

(Melbourne, 1988)

CONTENTS

0 *Introduction*

1 *Scope*

2 *References*

3 *Definitions*

3.1 OSI Reference Model Definitions

3.2 Basic Directory Definitions

3.3 Distributed Operation Definitions

4 *Abbreviations*

5 *Conventions*

6 *Protocol Overview*

6.1 Directory Protocol Model

6.2 Directory Access Protocol

6.3 Directory System Protocol

6.4 Use of Underlying Services

7 *Directory Protocol Abstract Syntax*

7.1 Abstract Syntaxes

7.2 Directory Application Service Elements

7.3 Directory Application Contexts

7.4 Errors

8 *Mapping onto Used Services*

8.1 Mapping onto ACSE

8.2 Mapping onto ROSE

9 *Conformance*

9.1 Conformance by DUAs

9.2 Conformance by DSAs

Annex A - DAP in ASN.1

Annex B - DSP in ASN.1

Annex C - Reference Definition of Protocol Object Identifiers

---

1) Recomendation X.519 and ISO 9594-5, The Directory - Protocol Specifications, were developed in close collaboration and are technically aligned

0 Introduction

0.1 This document, 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.

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

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

0.3 This Recommendation specifies the application service elements and application contexts for two protocols - the Directory Access Protocol (DAP) and the Directory System Protocol (DSP). The DAP provides for access to the Directory to retrieve or modify Directory information. The DSP provides for the chaining of requests to retrieve or modify Directory information to other parts of the distributed Directory System where the information may be held.

1 Scope

This Recommendation specifies the Directory Access Protocol and the Directory System Protocol, fulfilling the abstract services specified in Recommendations X.511 and X.518.

2 References

Recommendation X.200 - Open Systems Interconnection - Basic Reference Model

Recommendation X.208 - Open Systems Interconnection - Specification of Abstract Syntax Notation (ASN.1)

Recommendation X.209 - Open Systems Interconnection - Specification of Basic Encoding rules for Abstract Syntax Notation One (ASN.1)

Recommendation X.500 - The Directory - Overview of Concepts, Models and Services

Recommendation X.501 - The Directory - Information Framework

Recommendation X.511 - The Directory - Abstract Service Definition

Recommendation X.518 - The Directory - Procedures for Distributed Operation

Recommendation X.520 - The Directory - Selected Attribute Types

Recommendation X.521 - The Directory - Selected Object Classes

Recommendation X.219 - Remote Operations - Model, Notation and Service Definition

Recommendation X.229 - Remote Operations - Protocol Specification

Recommendation X.217 - Open Systems Interconnection - Association Control: Service Definition

Recommendation X.227 - Open Systems Interconnection - Association Control: Protocol Specification

Recommendation X.216 - Open Systems Interconnection - Presentation Layer Service Definition.

3 Definitions

The definitions contained in this paragraph make use of the abbreviations defined in 4.

3.1 *OSI Reference Model definitions*

This Recommendation is based on the concepts developed in Recommendation X.200 and makes use of the following terms defined therein:

a) application-service-element;

b) application-protocol-control-information;

c) application-control-data-unit;

d) application-context;

e) application-entity;

f) abstract-syntax.

3.2 *Basic Directory definitions*

This Recommendation makes use of the following terms defined in Recommendation X.501:

a) the Directory;

b) (Directory) user;

c) Directory System Agent (DSA);

d) Directory User Agent (DUA).

3.3 *Distributed Operation definitions*

This Recommendation makes use of the following terms defined in Recommendation X.518:

a) chaining;

b) referral.

4 Abbreviations

The following abbreviations are used in this Recommendation:

AC Application Context

ACSE Association Control Service Element

AE Application Entity

APCI Application Protocol Control Information

APDU Application Protocol Data Unit

ASE Application Service Element

DAP Directory Access Protocol

DSA Directory System Agent

DSP Directory System Protocol

DUA Directory User Agent

ROSE Remote Operations Service Element.

5 Conventions

The Recommendation makes use of the following conventions:

a) the abstract syntax definitions in 7 are defined using the abstract syntax notation defined in Recommendation X.208;

b) the remote operation macros (RO-notation), and the application-service-element and application-context macros are defined in Recommendation X.219;

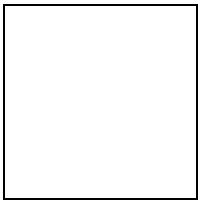
c) the words of defined terms and the names and values of service parameters and protocol fields, unless they are proper names, begin with a lower-case letter and are linked by a hyphen thus: defined-term. Proper names begin with an upper case letter and are not linked by a hyphen thus: Proper Name.

6 Protocol Overview

6.1 *Directory Protocol Model*

Recommendation X.511 defines the abstract service between a DUA and the Directory to support a user accessing Directory services. The Directory is further modelled as being represented by a DSA which supports the particular access point concerned. Recommendation X.518 defines the interactions between a pair of DSAs within the Directory to support user requests which are chained. These concepts are illustrated in Figure 1/X.519.

FIGURE 1/X.519 - T0704650-88



When a DUA is in a different open system from a DSA with which it is interacting, these interactions are supported by the Directory Access Protocol (DAP), which is an OSI application layer protocol. Similarly, when a pair of DSAs which are interacting are in different open systems, the interactions are supported by the Directory System Protocol (DSP), which is also in the application layer.

Both the DAP and the DSP are protocols to provide communication between a pair of application processes. In the OSI environment this is represented as communication between a pair of application- entities (AEs) using the presentation service. The function of an AE is provided by a set of application-service-elements (ASEs). The interaction between AEs is described in terms of their use of the services provided by the ASEs. The two ASEs common to both of the directory protocols are summarized in this paragraph.

The Remote Operations Service Element (ROSE) supports the request/reply paradigm of the abstract operation that occurs at the ports in the abstract model. The Directory ASEs provide the mapping function of the abstract-syntax notation of the directory abstract-service onto the services provided by the ROSE.

The Association Control Service Element (ACSE) supports the establishment and release of an application-association between a pair of AEs. Associations between a DUA and a DSA may be established only by the DUA. Only the initiator of an established association can release it.

6.2 *Directory Access Protocol*

The Directory Access Protocol (DAP) is used to realise the Directory Abstract Service. It comprises three directory specific ASEs in addition to ROSE and ACSE. These are: readASE, searchASE, and modifyASE. They correspond to the readPort, searchPort, and modifyPort of the abstract service. The directoryAccessAC application context identifies the combination of: readASE, searchASE, and modifyASE, aCSE, rOSE.

6.3 *Directory System Protocol*

The Directory System Protocol (DSP) is used to realise the functionality of distributed operation described in Recommendation X.518. It comprises three directory specific ASEs in addition to ROSE and ACSE. These are: **chainedReadASE**, **chainedSearchASE**, and **chainedModifyASE**. They correspond to the chainedReadPort, **chainedSearchPort**, and **chainedModifyPort** of the abstract service. The **directorySystemAC** application context identifies the combination of: **chainedReadASE**, **chainedSearchASE**, and **chainedModifyASE**, **aCSE**, **rOSE**.

6.4 *Use of Underlying Services*

The DAP and DSP protocols make use of underlying services as described below.

6.4.1 *Use of ROSE services*

The Remote Operations Service Element (ROSE) is defined in Recommendation X.219.

The ROSE supports the request/reply paradigm of remote operations.

The Directory ASEs are users of the RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U and RO-REJECT-P services of the ROSE.

The remote operations of the DAP and the DSP are Class 2 (asynchronous) operations. Note that as the DUA is a consumer of the DAP it may choose to operate in a synchronous manner.

DAP uses Association Class 1. This means that the DSA cannot invoke operations on the DUA. DSP uses Association Class 3. This means that the responding DSA can invoke operations on the initiating DSA and vice versa.

6.4.2 *Use of ACSE services*

The Association Control Service Element (ACSE) is defined in Recommendation X.217.

The ACSE provides for the control (establishment, release, abort) of application-associations between AEs.

The Directory Bind and Directory Unbind (or DSA Bind and DSA Unbind) are the sole users of the A-ASSOCIATE and A-RELEASE services of the ACSE in normal mode. The application-process is the user of the A-ABORT and A-P-ABORT services of the ACSE.

6.4.3 *Use of the Presentation Service*

The presentation-service is defined in Recommendation X.216.

The Presentation Layer coordinates the representation (syntax) of the Application Layer semantics that are to be exchanged.

In normal mode, a different presentation-context is used for each abstract-syntax included in the application-context.

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

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

6.4.4 *Use of Lower Layer Services*

The session-service is defined in Recommendation X.215. The Session Layer structures the dialogue of the flow of information between the end-systems.

The Kernel and Duplex functional units of the session-service are used by the Presentation Layer.

The transport-service is defined in Recommendation X.214. The Transport Layer provides for the end-to-end transparent transfer of data over the underlying network connection.

The choice of the class of transport-service used by the Session Layer depends on the requirements for multiplexing and error recovery. Support for Transport Class 0 (non-multiplexing) is mandatory. Transport Expedited Service is not used.

Support for other classes is optional. A multiplexing class may be used to multiplex the DAP or DSP and other protocols over the same network connection. An error recovery class may be chosen over a network connection with an unacceptable residual error rate.

An underlying network supporting the OSI network-service defined in Recommendation X.213 is assumed.

A network-address is as defined in Recommendation X.121, Recommendations E.163/E.164, or Recommendation X.200 (OSI NSAP-address).

7 Directory Protocol Abstract Syntax

7.1 *Abstract Syntaxes*

The Directory ASEs specified in 7.2.1, 7.2.3 and 7.2.5 share a single abstract syntax, id-**as-directory-AccessAS**. Those specified in 7.2.2, 7.2.4 and 7.2.6 also share a single abstract syntax **id-as-directorySystemAS**. In each case, this defines application-protocol-control-information (APCI) which, when used in conjunction with the ROSE, defines a set of APDUs. The Directory APDUs are defined by the abstract-syntax

of the Directory ASEs and ROSE. These plus the abstract-syntax of ACSE form the complete definition of APDUs used during a Directory association.

The ACSE abstract-syntax **id-as-acse** is needed to establish the associations.

These abstract syntaxes shall (as a minimum) be encoded according to the ASN.1 Basic Encoding Rules.

7.2 *Directory Application Service Elements*

This paragraph specifies the ASEs which are used as "building blocks" in the construction of the various Directory application contexts in 7.3.

Note - These ASEs are used for the construction of the application contexts defined in this Recommendation. They are not intended to allow for claims of conformance to individual, or other combinations of, ASEs.

7.2.1 *Read ASE*

The **readASE** supports the abstract-operations of the **readPort**, namely Read, Compare, and **Abandon**, as defined in Recommendation X.511.

```
readASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
                {read, compare, abandon}
    ::= id-ase-readASE

read        Read      ::= 1
compare     Compare   ::= 2
abandon     Abandon   ::= 3
```

7.2.2 *Chained Read ASE*

The **chainedReadASE** supports the abstract-operation of the ChainedReadPort, i.e. ChainedRead, **ChainedCompare** and **ChainedAbandon**, as defined in Recommendation X.518.

```
chainedReadASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {
                chainedRead,
                chainedCompare
                chainedAbandon}
    ::= id-ase-chainedReadASE

chainedRead         ChainedRead         ::= 1
chainedCompare      ChainedCompare      ::= 2
chainedAbandon      ChainedAbandon      ::= 3
```

7.2.3 *Search ASE*

The searchASE supports the abstract-operations of the SearchPort, namely List and Search, as defined in Recommendation X.511.

```
searchASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES { list, search}
    ::= id-ase-searchASE}

list    List        ::= 4
search  Search      ::= 5
```

7.2.4 *Chained Search ASE*

The chainedSearchASE supports the abstract-operations of the ChainedSearchPort, namely ChainedList and ChainedSearch, as defined in Recommendation X.518.

```
chainedSearchASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {
                chainedList, chainedSearch}
    ::=  id-ase-chainedSearchASE

chainedList          ChainedList          ::= 4

chainedSearch        ChainedSearch        ::= 5
```

7.2.5 *Modify ASE*

The **modifyASE** supports the abstract-operations of the **ModifyPort**, namely **AddEntry**, **RemoveEntry**, **ModifyEntry**, and **ModifyRDN**, as defined in Recommendation X.511.

```
modifyASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
                {addEntry, removeEntry,
                modifyEntry, modifyRDN}
    ::=  id-ase-modifyASE

addEntry        AddEntry        ::= 6

removeEntry     RemoveEntry     ::= 7

modifyEntry     ModifyEntry     ::= 8

modifyRDN       ModifyRDN       ::= 9
```

7.2.6 *Chained Modify ASE*

The **chainedModifyASE** supports the abstract-operations of the **ChainedModifyPort**, namely **ChainedAddEntry**, **ChainedRemoveEntry**, **ChainedModifyEntry** and **ChainedModifyRDN**, as defined in Recommendation X.518.

```
chainedModifyASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
                {chainedAddEntry,
                 chainedRemoveEntry,
                 chainedModifyEntry,
                 chainedModifyRDN}
    ::=  id-ase-chainedModifyASE

chainedAddEntry       ChainedAddEntry       ::= 6
chainedRemoveEntry    ChainedRemoveEntry    ::= 7
chainedModifyEntry    ChainedModifyEntry    ::= 8
chainedModifyRDN      ChainedModifyRDN      ::= 9
```

7.3 *Directory Application Contexts*

7.3.1 *Directory Access Application Context*

The **directoryAccessAC** allows the DUA to access the operations of the following ASEs: **readASE**, **searchASE**, **modifyASE**.

```
directoryAccessAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
                {aCSE}
        BIND DirectoryBind
        UNBIND DirectoryUnbind
        REMOTE OPERATIONS {rOSE}
        INITIATOR CONSUMER OF {
                readASE,
```

```
                searchASE,
                modifyASE}
            ABSTRACT SYNTAXES {
                id-as-acse,
                id-as-directoryAccessAS}
        ::= id-ac-directoryAccessAC
```

7.3.2 *Directory System Application Context*

The **directorySystemAC** allows DSAs to communicate for the purpose of chaining operations.

```
directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
        REMOTE OPERATIONS {rOSE}
        OPERATIONS OF
            {chainedReadASE,
            chainedSearchASE,
            chainedModifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse,
            id-as-directorySystemAS}
::=     id-ac-directorySystemAC
```

7.4 *Errors*

Corresponding to each abstract-error defined in the Abstract Service is an error value which may be conveyed by the protocol. The assignments follow:

```
abandoned           Abandoned       ::= 5
attributeError      AttributeError  ::= 1
nameError           NameError       ::= 2
referral            Referral        ::= 4
securityError       SecurityError   ::= 6
serviceError        ServiceError    ::= 3
updateError         UpdateError     ::= 8
dSAReferral         DSAReferral     ::= 9
abandonFailed       AbandonFailed   ::= 7
```

8 Mapping onto Used Services

This paragraph defines the mapping of the DAP and DSP onto the used services.

8.1 *Mapping onto ACSE*

This paragraph defines the mapping of the abstract-bind (DirectoryBind or DSABind) and abstract-unbind (DirectoryUnbind or DSAUnbind) services onto the services of the ACSE. The ACSE is defined in Recommendation X.217.

8.1.1 *Abstract-bind onto A-ASSOCIATE*

The abstract-bind service is mapped onto the A-ASSOCIATE service of the ACSE. The use of the parameters of the A-ASSOCIATE service is qualified in the following subparagraphs.

8.1.1.1 *Mode*

This parameter shall be supplied by the initiator of the association in the A-ASSOCIATE request primitive, and shall have the value "normal mode".

8.1.1.2 *Application Context Name*

The initiator of the association shall propose either the **directoryAccessAC** or the directorySystemAC application-context.

8.1.1.3 *User information*

The mapping of the bind-operation of the abstract-bind service onto the User Information parameters of the A-ASSOCIATE request primitive is defined in Recommendation X.219.

8.1.1.4 *Presentation Context Definition List*

The initiator of the association shall supply the Presentation Context Definition List in the A-ASSOCIATE request primitive which shall contain the ACSE abstract-syntax (id-as-acse) and either the DAP abstract-syntax (**id-as-directoryAccessAS**) or the DSP abstract-syntax (**id-as- directorySystemAS**).

8.1.1.5 *Quality of service*

This parameter shall be supplied by the initiator of the association in the A-ASSOCIATE request primitive, and by the responder of the association in the A-ASSOCIATE response primitive. The parameters "Extended Control" and "Optimized Dialogue Transfer" shall be set to "feature not desired". The remaining parameters shall be such that default values are used.

8.1.1.6 *Session requirements*

This parameter shall be set by the initiator of the association in the A-ASSOCIATE request primitive, and by the responder of the association in the A-ASSOCIATE response primitive. The parameter shall be set to specify the following functional units:

a) Kernel;

b) Duplex.

8.1.1.7 *Application Entity Title and Presentation Address*

These parameters shall be supplied by the initiator and the responder of the association (Application Entity Title is optionally supplied). For a DUA establishing an association for an initial request, these parameters are obtained from locally held information.

For a DUA (or DSA) establishing an association with a DSA to which it has been referred, these parameters are obtained from the AccessPoint value of a ContinuationReference. For a DSA establishing an association, this parameter is obtained from its Knowledge Information, i.e. an external reference.

8.1.2 *Abstract-unbind onto A-RELEASE*

The abstract-unbind service is mapped onto the A-RELEASE service of the ACSE. The use of the parameters of the A-RELEASE service is qualified in the following subparagraph.

8.1.2.1 *Result*

This parameter shall have the value "affirmative".

8.1.3 *Use of A-ABORT and A-P-ABORT services*

The application-process is the user of the A-ABORT and A-P-ABORT services of the ACSE.

8.2 *Mapping onto ROSE*

The Directory ASE services are mapped onto the RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U and RO-REJECT-P services of the ROSE. The mapping of the abstract-syntax notation of the Directory ASEs onto the ROSE services is as defined in Recommendation X.219.

9 Conformance

This paragraph defines the requirements for conformance to this Recommendation.

9.1 *Conformance by DUAs*

A DUA implementation claiming conformance to this Recommendation shall satisfy the requirements specified in 9.1.1 to 9.1.3.

9.1.1 *Statement requirements*

The following shall be stated:

a) the operations of the directoryAccessAC application-context that the DUA is capable of invoking for which conformance is claimed; and

b) the security-level(s) for which conformance is claimed (none, simple, strong).

9.1.2 *Static requirements*

A DUA shall:

a) have the capability of supporting the directoryAccessAC application-context as defined by its abstract syntax in 7.

9.1.3 *Dynamic requirements*

A DUA shall:

a) conform to the mapping onto used services defined in 8.

9.2 *Conformance by DSAs*

A DSA implementation claiming conformance to this Recommendation shall satisfy the requirements specified in 9.2.1 to 9.2.3.

9.2.1 *Statement requirements*

The following shall be stated:

a) the application-contexts for which conformance is claimed: directoryAccessAC, directorySystemAC, or both. If a DSA is such that knowledge of it has been disseminated causing knowledge references to the DSA to be held by other DSA(s) outside of its own DMD, then it shall claim conformance to the directorySystemAC;

Note - An application context shall not be divided, except as stated herein: in particular, conformance may not be claimed to particular ports or operations.

b) whether or not the DSA is capable of acting as a first-level DSA, as defined in Recommendation X.518;

c) if conformance is claimed to the directorySystemAC application-context, whether or not the chained mode of operation is supported, as defined in Recommendation X.518;

d) the security-level(s) for which conformance is claimed (none, simple, strong);

e) the selected attribute types defined in Recommendation X.520 and any other attribute types, for which conformance is claimed; and

f) the selected object classes defined in Recommendation X.521 and any other object classes, for which conformance is claimed.

9.2.2 *Static requirements*

A DSA shall:

a) have the capability of supporting the application-contexts for which conformance is claimed as defined by their abstract syntax in 7;

b) have the capability of supporting the information framework defined by its abstract syntax in Recommendation X.501;

c) conform to the minimal knowledge requirements defined in Recommendation X.518;

d) if conformance is claimed as a first-level DSA, conform to the requirements for support of the root context, as defined in Recommendation X.518;

e) have the capability of supporting the attribute types for which conformance is claimed as defined by their abstract syntaxes; and

f) have the capability of supporting the object classes for which conformance is claimed, as defined by their abstract syntaxes.

9.2.3 *Dynamic requirements*

A DSA shall:

a) conform to the mapping onto used services defined in 8 of this Recommendation;

b) conform to the procedures for distributed operation of the Directory related to referrals, as defined in Recommendation X.518;

c) if conformance is claimed to the directoryAccessAC application-context, conform to the procedures of Recommendation X.518 as they relate to the referral mode of the DAP;

d) if conformance is claimed to the directorySystemAC application-context, conform to the referral mode of interaction, as defined in Recommendation X.518;

e) if conformance is claimed to the chained mode of interaction, conform to the chained mode of interaction, as defined in Recommendation X.518.

Note - Only in this case is it necessary for a DSA to be capable of invoking operations using the directorySystemAC.

ANNEX A

(to Recommendation X.519)

DAP in ASN.1

This Annex is part of the Recommendation.

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

```
DirectoryAccessProtocol {joint-iso-ccitt ds(5) modules(1) dap(11)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directoryAccessAC, readASE, searchASE, modifyASE;

IMPORTS
    abstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    id-ac-directoryAccessAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-as-directoryAccessAS, id-as-acse
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    DirectoryBind, DirectoryUnbind, Read, Compare, Abandon, List,
    Search, AddEntry, RemoveEntry, ModifyEntry, ModifyRDN, Abandoned, AbandonFailed,
    AttributeError, NameError, Referral, SecurityError, ServiceError,
    UpdateError
    FROM    DirectoryAbstractService
            directoryAbstractService;
```

```
-- Application Contexts --
directoryAccessAC
      APPLICATION-CONTEXT
            APPLICATION SERVICE ELEMENTS {aCSE}
            BIND DirectoryBind
      UNBIND DirectoryUnbind
            REMOTE OPERATIONS {rOSE}
                  INITIATOR CONSUMER OF {readASE, searchASE, modifyASE}
            ABSTRACT SYNTAXES {
                  id-as-acse, id-as-directoryAccessAS}
       ::= id-ac-directoryAccessAC
-- Read ASE --
readASE
      APPLICATION-SERVICE-ELEMENT
            CONSUMER INVOKES {read, compare, abandon}
      ::= id-ase-readASE
-- Search ASE --
searchASE
      APPLICATION-SERVICE-ELEMENT
            CONSUMER INVOKES {list, search}
      ::= id-ase-searchASE
-- Modify ASE --
modifyASE
      APPLICATION-SERVICE-ELEMENT
            CONSUMER INVOKES
                  {addEntry, removeEntry,
                   modifyEntry, modifyRDN}
            ::= id-ase-modifyASE
-- Remote Operations --
read                  Read                  ::= 1
compare               Compare               ::= 2
abandon               Abandon               ::= 3
list                  List                  ::= 4
search                Search                ::= 5
addEntry              AddEntry              ::= 6
removeEntry           RemoveEntry           ::= 7
modifyEntry           ModifyEntry           ::= 8
modifyRDN             ModifyRDN             ::= 9
-- Remote Errors --
attributeError        AttributeError        ::= 1
nameError             NameError             ::= 2
serviceError          ServiceError          ::= 3
referral              Referral              ::= 4
abandoned             Abandoned             ::= 5
securityError         SecurityError         ::= 6
abandonFailed         AbandonFailed         ::= 7
updateError           UpdateError           ::= 8
END
```

ANNEX B

(to Recommendation X.519)

DSP in ASN.1

This Annex is part of the Recommendation.

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

```
DirectorySystemProtocol {joint-iso-ccitt ds(5) modules(1) dsp(12)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directorySystemAC, chainedReadASE, chainedSearchASE, chainedModifyASE;

IMPORTS
    distributedOperations, directoryAbstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}
    id-ac-directorySystemAC, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE,
    id-as-directorySystemAS, id-as-acse;
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    Abandoned, AttributeError, AbandonFailed,
    NameError, DSAReferral, SecurityError, ServiceError, UpdateError
    FROM    DirectoryAbstractService directoryAbstractService

    DSABind, DSAUnbind,
    ChainedRead, ChainedCompare, ChainedAbandon,
    ChainedList, ChainedSearch,
    ChainedAddEntry, ChainedRemoveEntry, ChainedModifyEntry,
    ChainedModifyRDN, DSAReferral,
    FROM    DistributedOperations
            distributedOperations;

    -- Application Contexts --

directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
        REMOTE OPERATIONS {rOSE}
            OPERATIONS OF {
                    chainedReadASE, chainedSearchASE, chainedModifyASE}
    ABSTRACT SYNTAXES {
                id-as-acse, id-as-directorySystemAS}
    ::=  {id-ac-directorySystemAC}

    -- Chained Read ASE --

chainedReadASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS          {chainedRead, chainedCompare, chainedAbandon}
::=  id-ase-chainedReadASE

-- Chained Search ASE --

chainedSearchASE
```

```
        APPLICATION-SERVICE-ELEMENT
            OPERATIONS {chainedList, chainedSearch}
        ::=  id-ase-chainedSearchASE

    -- Chained Modify ASE --

    chainedModifyASE
        APPLICATION-SERVICE-ELEMENT
            OPERATIONS
                    {chainedAddEntry, chainedRemoveEntry,
                    chainedModifyEntry, chainedModifyRDN}
        ::=  id-ase-chainedModifyASE

    -- Remote Operations --

    chainedRead                 ChainedRead                 ::= 1
    chainedCompare              ChainedCompare              ::= 2
    chainedAbandon              ChainedAbandon              ::= 3
    chainedlist                 ChainedList                 ::= 4
    chainedSearch               ChainedSearch               ::= 5
    chainedAddEntry             ChainedAddEntry             ::= 6
    chainedRemoveEntry          ChainedRemoveEntry          ::= 7
    chainedModifyEntry          ChainedModifyEntry          ::= 8
    chainedModifyRDN            ChainedModifyRDN            ::= 9

    -- Remote Errors --

    attributeError              AttributeError              ::= 1
    nameError                   NameError                   ::= 2
    serviceError                ServiceError                ::= 3
    abandoned                   Abandoned                   ::= 5
    securityError               SecurityError               ::= 6
    abandonFailed               AbandonFailed               ::= 7
    updateError                 UpdateError                 ::= 8
    dsaReferral                 DSAReferral                 ::= 9

    END
```

ANNEX C

(to Recommendation X.519)

Reference definition of protocol object identifiers

This Annex is part of the Recommendation.

This Annex includes all of the ASN.1 Object Identifiers assigned in this Recommendation in the form of ASN.1 module, **ProtocolObjectIdentifiers**.

```
ProtocolObjectIdentifiers {joint-iso-ccitt ds(5) modules(1)  protocolObjectIdentifiers(4)}

DEFINITIONS ::=
BEGIN
EXPORTS
    id-ac-directoryAccessAC, id-ac-directorySystemAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE, id-as-acse,
    id-as-directoryAccessAS, id-as-directorySystemsAS;

IMPORTS
    id-ac, id-ase, id-as
```

```
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- Application Contexts --
id-ac-directoryAccessAC         OBJECT IDENTIFIER ::= {id-ac 1}
id-ac-directorySystemAC         OBJECT IDENTIFIER ::= {id-ac 2}
-- ASEs --
id-ase-readASE                  OBJECT IDENTIFIER ::= {id-ase 1}
id-ase-searchASE                OBJECT IDENTIFIER ::= {id-ase 2}
id-ase-modifyASE                OBJECT IDENTIFIER ::= {id-ase 3}
id-ase-chainedReadASE           OBJECT IDENTIFIER ::= {id-ase 4}
id-ase-chainedSearchASE         OBJECT IDENTIFIER ::= {id-ase 5}
id-ase-chainedModifyASE         OBJECT IDENTIFIER ::= {id-ase 6}
-- ASs --
id-as-directoryAccessAS         OBJECT IDENTIFIER ::= {id-as 1}
id-as-directorySystemAS         OBJECT IDENTIFIER ::= {id-as 2}
id-as-acse                      OBJECT IDENTIFIER ::=
                                {joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0)
                                version1(1)}
END
```