INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

F.515

(04/2003)

SERIES F: NON-TELEPHONE TELECOMMUNICATION SERVICES

Directory services

Unified Directory Specification

ITU-T Recommendation F.515

ITU-T F-SERIES RECOMMENDATIONS

NON-TELEPHONE TELECOMMUNICATION SERVICES

TELEGRAPH SERVICE	
Operating methods for the international public telegram service	F.1–F.19
The gentex network	F.20–F.29
Message switching	F.30–F.39
The international telemessage service	F.40–F.58
The international telex service	F.59–F.89
Statistics and publications on international telegraph services	F.90–F.99
Scheduled and leased communication services	F.100–F.104
Phototelegraph service	F.105–F.109
MOBILE SERVICE	
Mobile services and multidestination satellite services	F.110–F.159
TELEMATIC SERVICES	
Public facsimile service	F.160–F.199
Teletex service	F.200–F.299
Videotex service	F.300–F.349
General provisions for telematic services	F.350–F.399
MESSAGE HANDLING SERVICES	F.400–F.499
DIRECTORY SERVICES	**F.500–F.549**
DOCUMENT COMMUNICATION	
Document communication	F.550–F.579
Programming communication interfaces	F.580–F.599
DATA TRANSMISSION SERVICES	F.600–F.699
AUDIOVISUAL SERVICES	F.700–F.799
ISDN SERVICES	F.800–F.849
UNIVERSAL PERSONAL TELECOMMUNICATION	F.850–F.899
HUMAN FACTORS	F.900–F.999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation F.515

Unified Directory Specification

Summary

This Recommendation provides a directory specification to fulfil the service requirements of ITU-T Rec. F.510. The directory specification is named Unified Directory Specification (UDS) to reflect that it allows a great variety of database structures to be supported. The ITU-T Rec. F.510 service is realized by the Unified Directory Access Protocol (UDAP). This protocol is expressed using the eXtensible Markup Language (XML) schema notation.

UDAP has been designed in a way that facilitates an easy mapping to ITU-T Rec. E.115.

The structure of this Recommendation follows the structure of ITU-T Rec. F.510. Certain clauses in ITU-T Rec. F.510 do not require a corresponding specification and are left void.

Source

ITU-T Recommendation F.515 was prepared by ITU-T Study Group 17 (2001-2004) and approved under the ITU-T Recommendation A.8 procedure on 22 April 2003.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Introduction

This Recommendation provides a directory specification to fulfil the service requirements of ITU-T Rec. F.510. The directory specification is named Unified Directory Specification (UDS) to reflect that it allows a great variety of database structures to be supported. The ITU-T Rec. F.510 service is realised by the Unified Directory Access Protocol (UDAP). This protocol is expressed using the eXtensible Markup Language (XML) schema notation.

UDAP has been designed in a way that facilitates an easy mapping to ITU-T Rec. E.115.

The structure of this Recommendation follows the structure of ITU-T Rec. F.510. Certain clauses in ITU-T Rec. F.510 do not require a corresponding specification and are left void.

Annex A, which is an integral part of this Recommendation, list the messages codes as defined in Annex A/F.510, and it lists the conditions under which they are generated and what additional information is associated with each message code.

Annex B, which is an integral part of this Recommendation, gives the formal specification of UDAP in XML schema notation.

Annex C, which is an integral part of this Recommendation, specifies how UDS uses underlying services.

Annex D, which is an integral part of this Recommendation, gives the formal specification of UDAP in ASN.1 notation.

An implementation shall implement the encoding specified either in Annex B, or in Annex D, or in both.

ITU-T Recommendation F.515

Unified Directory Specification

1 Scope

This Recommendation only considers directory information retrieval using the search operation. Requirements on administrative workstations or administrative capabilities are outside the scope of this Recommendation.

This Recommendation supports the Directory Service as specified by ITU-T Rec. F.510. However, it could be used in other environments.

This Recommendation considers only client to directory server communication. It does not consider directory server to directory server communication.

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- ITU-T Recommendation E.115 (1995), *Computerized directory assistance*.
- ITU-T Recommendation F.510 (2003), *Automated directory assistance – White pages service definition*.
- ITU-T Recommendation X.693 (2001) | ISO/IEC 8825-4:2002, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER)*.
- ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.
- ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*.

3 Definitions

3.1 White pages service definitions

The following terms are defined in ITU-T Rec. F.510:

a) (directory) attribute;

b) (directory) entry;

c) directory service;

d) hierarchical group;

e) key attribute;

f) query service entry.

3.2 Unified directory definitions

This Recommendation defines the following terms:

3.2.1 client: An application that issues directory requests and receives directory information in responses.

3.2.2 directory server: An application that can return directory information when receiving request over the Unified Directory Access Protocol.

3.2.3 entry: A collection of directory attributes and families that comprises the information held about an object.

3.2.4 family: A grouping of related directory attributes within an entry.

3.2.5 filter: A construct carried in a search request to be matched against entry information.

3.2.6 filter item: A component of a filter that carries information to be matched against directory attributes of a particular type.

3.2.7 object: An entity, e.g., a person, that is represented by an entry of a directory server.

4 Abbreviations

This Recommendation uses the following abbreviations:

ASN.1	Abstract Syntax Notation One
DMD	Directory Management Domain
GSM	Global System for Mobile communications
IANA	Internet Assigned Numbers Authority
ISDN	Integrated Services Digital Network
LDAP	Lightweight Directory Access Protocol
PSTN	Public Switched Telephone Network
SMS	Short Message Service
SOAP	Simple Object Access Protocol
UDAP	Unified Directory Access Protocol
UDS	Unified Directory Specification
UMTS	Universal Mobile Telecommunications System
XML	eXtensible Markup Language

5 Conventions

5.1 Text conventions

There is terminology collision between that of directories and that of eXtensible Markup Language (XML). The terms attributes and schema are used in both environments. To avoid confusions, those two terms are qualified by "directory" or "XML", as appropriate.

This Recommendation presents XML notation in Courier New, 9-point typeface and ASN.1 in Courier New, 10-point typeface. When XML specifications are referenced in normal text, they are differentiated from other text by presenting them in the bold Courier New, 10-point typeface.

5.2 XML notation conventions

5.2.1 General on XML notation

All complex and simple XML types defined by this Recommendation have a name that starts with an uppercase letter.

All names of non-abstract XML elements defined by this Recommendation start with a lowercase letter.

All names of XML attributes defined by this Recommendation start with a lowercase letter.

All names of abstract elements start with uppercase letter.

5.2.2 Syntaxes with upper bounds

Most directory attribute types with string syntax has a maximum length, termed upper bound. Simple XML types, which specifies upper limits, are defined for that purpose.

The name of such a simple XML type is the letters **`Ub`** plus a number, for example **`Ub64`**, where the number indicates the upper bound in number of characters for the type. Such types are derived from the XML built-in **`string`** type.

5.2.3 Enumerated types

Some elements and XML attributes may take one out of several predefined values. Special XML simple types are defined for such element types.

The XML built-in type **`NCName`** is the base type for an enumerated type.

An XML enumerated simple type is named by taking a *`descriptive name`*, e.g., **`AttributeType`**. The first character of the name of such a simple type is an uppercase letter.

The value of an XML element or an XML attribute of the enumerated type shall be exactly one of the defined enumerated values.

5.2.4 List types

The XML list type facility is used where several values of an enumerated type may be required within a single XML element or XML attribute. An XML simple list type is given a *`descriptive name`* suffixed with an **`s`**, e.g., **`AttributeTypes`**. The first character of the name of such a simple type is an uppercase letter.

A value of this type holds zero or more of the defined enumerated values.

5.2.5 Abstract element schema definitions and substitution groups

Abstract XML element schema definitions are used as "head" elements for XML schema substitution groups. The first character of a name of an abstract XML element is an uppercase letter.

6 Directory service model

Figure 1 illustrates the position of the UDAP with respect to different components defined within clause 6/F.510. The protocol between the access service and the query service agent could be any private or standardized protocol. In particular, it could be UDAP as defined by this Recommendation. The query service agent may, dependent on circumstances, provide some of the ITU-T Rec. F.510 service elements. However, the UDAP supports the full set of ITU-T Rec. F.510 service elements.

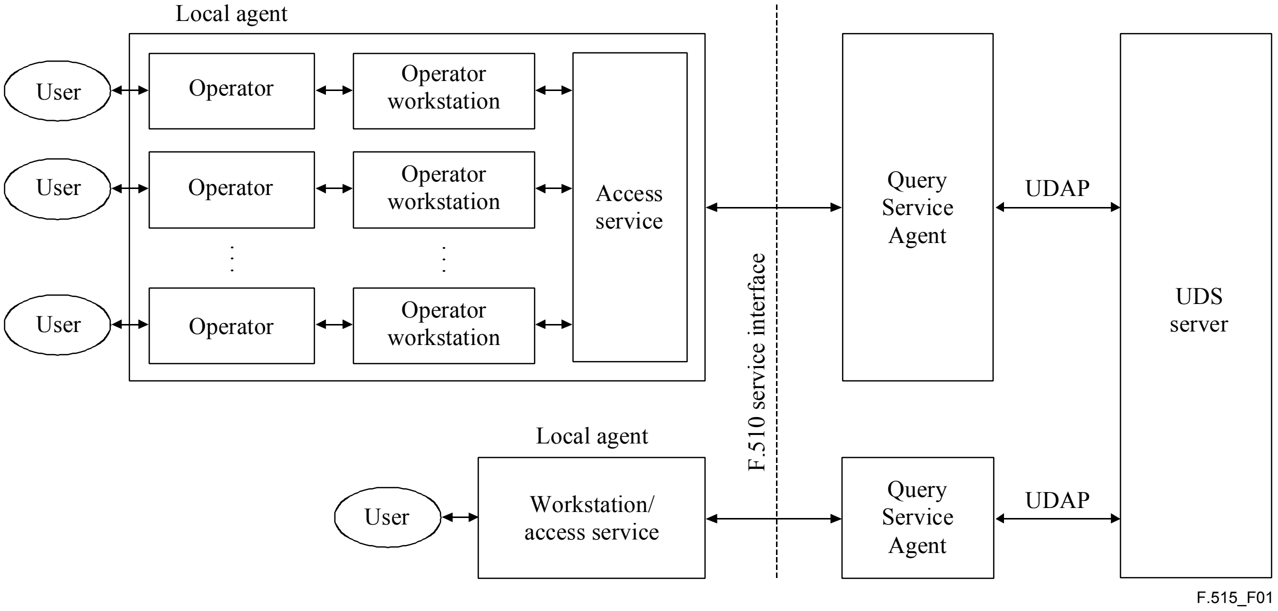

Figure 1/F.515 – Possible service scenarios

As the UDAP may be used in other places than indicated in Figure 1, the term *client* is used for the function that issues a UDAP request, and the term *directory server* is used for the function replying to a UDAP request with a response.

7 Directory information model

7.1 Information structure

7.1.1 Relationship to ITU-T Rec. F.510

ITU-T Rec. F.510 is a service definition for a White Pages service. However, it does not provide details on how this service should be realized. This Recommendation gives a directory specification for the support of ITU-T Rec. F.510. It is also able to cope with some future extensions to that Recommendation.

7.1.2 Entry structure

See Figure 2

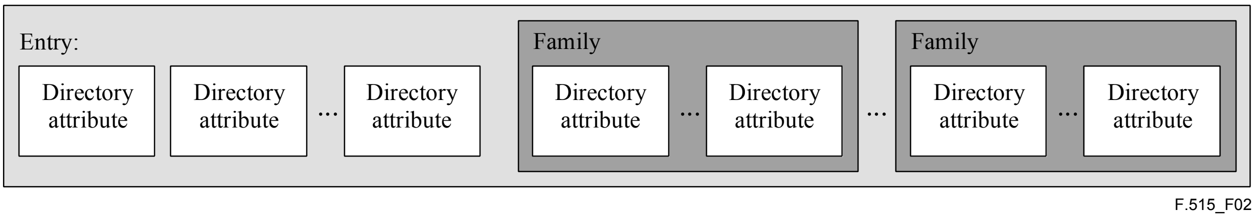

Figure 2/F.515 – Structure of a directory entry

A directory holds information about objects. The following object classes are defined:

a) subscriber, which is either a residential person, an organization or a governmental institution;

b) state-or-province;

c) locality;

d) street; and

e) business category.

The collection of directory information about some object is called an entry. The entry model is only used for defining how directory information is transferred on the protocol. It does not imply anything about the internal structure of a database.

The information within an entry is modelled as a set of directory attributes and families.

Each directory attribute holds a particular piece of information, e.g., a street address. A particular directory attribute is an instance of a directory attribute type. A directory attribute type is a definition of the type of information held by an (instance) directory attribute of that type.

There may be multiple directory attributes of the same type within an entry. As an example, a locality name may be provided in different languages.

The content of directory entries corresponding the above different object classes is outlined in 7.2, 7.3 and 7.5.

A family is a grouping of related directory attributes that describes one particular entity, for example, a communications address (see 7.5.2). A particular family is an instance of a family type.

7.1.3 Directory attributes

7.1.3.1 Directory attribute type definitions

A directory attribute type is defined as an XML element type. The element name is a global unique identification of the directory attribute type. A directory attribute is an instance of this element type where the content of the element is the value of the directory attribute.

The following is a typical directory attribute specification:

```
<streetAddress xml:lang="da">Nakkedraget</streetAddress>
```

The syntax of the value is part of the element definition. Some directory attribute type definitions include the **`xml:lang`** XML attribute, which indicates the language of the directory attribute value. As an example, a street address in Brussels have both a French and a Flemish variant.

All the defined directory attribute types belong to the XML **`Attribute`** substitution group. This means that directory attributes can be inserted into the protocol whenever the **`Attribute`** abstract element is specified within the XML schema (as an example, see 7.1.4).

7.1.3.2 Grouping of directory attributes (families)

It is possible to define different family types. A family type is defined as an XML element type. The element name is a global unique identification of the family type. A family element has as child elements one or more elements of the XML **`Attribute`** substitution group.

A family type definition is member of the **`Family`** substitution group.

The only currently defined member of the **`Family`** substitution group is the **`commsAddress`** family type.

NOTE – Other family types could be defined in the future, e.g., for postal addresses.

7.1.4 Entry representation

Entry information, as it is transmitted in the protocol, is represented by the `entry` XML element (see Figure 3).

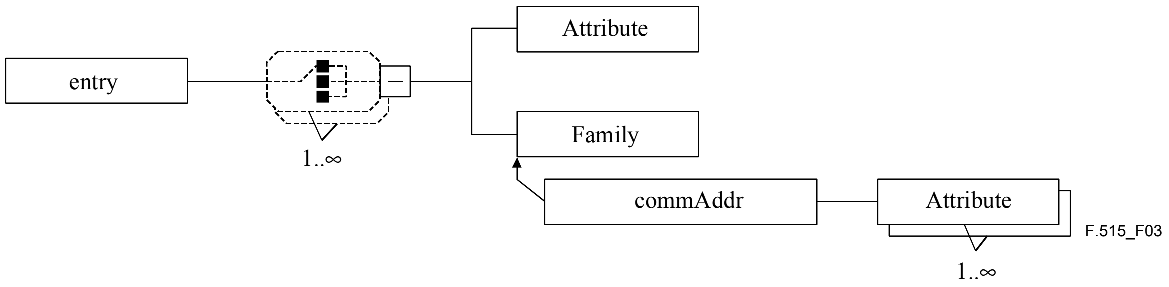

Figure 3/F.515 – Transmitted entry information

A child element of the `entry` element is a choice between an element of the `Attribute` substitution group and an element of the `Family` substitution group. This choice can be taken one or more times as indicated in above figure. This allows such elements to be present in any number and order.

The `entry` element has the following XML attributes:

- The `entryIdentifier` XML attribute allows an entry identifier to be assigned to the entry (see 7.1.5).
- The `hierarchyLevel` XML attribute shall be present for any entry that is a member of a hierarchical group. This XML attribute shall take the value zero for the hierarchical top.
- The `hierarchyBelow` XML attribute indicates whether the entry has any hierarchical children. A value of **true** indicates that hierarchical children exist. If the value is **false** or this XML attribute is absent, no hierarchical children exist.
- The `ReturnedObject` XML attribute indicates the type of entry returned. It can take the value `subscriber` (default), `stateOrProvince`, `locality`, `street` or `businessCategory`.

7.1.5 Entry identifiers

Clause 7.5/F.510 introduces the concept of *entry identifiers*. An entry identifier may be supplied in a UDAP search request (see 10.5.1.2). An entry identifier may also be supplied for some or all of the entries returned in a search result.

ITU-T Rec. F.510 mandates the support of entry identifiers, but support is optional according to this Recommendation. However, if an implementation is able to return identification information in the results, it shall also support identification information in the requests.

Entry identifier can be used for accessing a particular entry directly. It can also be used to establish relationship among members of a hierarchical group.

Some databases have external visible identifiers that can be used as permanent entry identifiers; others have not. Although a database may not provide for such permanent entry identifiers, the directory server may assign temporary entry identifiers to entries returned. The client can then within a locally defined period use an entry identifier to access a particular entry.

The entry identifier is given as an XML attribute with syntax `xsd:base64Binary` allowing any bit string to be represented.

This XML attribute is mandatory for the `base` child element of the `searchRequest` and it is optional for the `entry` element of the `searchResult`.

If the client in a request specified an expired or unknown entry identifier, the directory server shall return an empty result with message code 0.1.

7.1.6 Search filter

7.1.6.1 Filter concept

A **filter** element is part of a search request. It consists of *filter items*. A filter item holds directory attribute information about a particular directory attribute type. The directory server compares this information against corresponding directory attribute(s) within each entry of the directory server. If the filter item matches just one directory attribute of the type, whether that directory attribute is a direct part of the entry or is contained within a family, the matching returns a value of TRUE. Otherwise; it returns a value of FALSE. If the combination of filter element results in a TRUE condition, the entry is a candidate to be returned in the search result. Otherwise, it is not.

NOTE 1 – Although an entry is a candidate to be returned, local policies may prevent that entry from being returned.

NOTE 2 – Currently, only AND'ing of filter element is supported, i.e., that all filter items have to return TRUE for the filter to return TRUE. The filter specification is made in such a way that extension to more complex and powerful filter specifications is possible.

A filter item can hold a complete directory attribute value. This shall be the case for all enumerated directory attribute types. In this case the filter item is represented by an XML element of the **Assertion** substitution group. A filter item can also hold one or more, possible truncated words to be matched against a string-type directory attribute. In this case the filter item is represented by an XML element of the **Substrings** substitution group.

7.1.6.2 Filter structure

A filter is carried in a search request in the **filter** XML element. Its structure is indicated in Figure 4. A child element of the **filter** element is a filter item that is defined as a choice between an element of the **Assertion** substitution group and an element of the **Substrings** substitution group. This choice can be taken one or more times. This allows multiple of such elements to be specified in any order

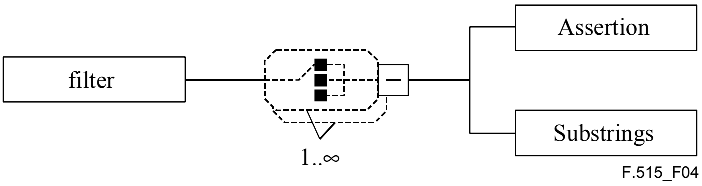

Figure 4/F.515 – Filter structure

7.1.6.3 Directory assertion definitions

A separate version of the directory attribute definition is used for defining elements of the **Assertion** substitution group. The suffix **As** is added to the element name of the corresponding basic directory attribute definition to separate it from that definition.

The following is a typical assertion specification:

```
<subscriberTypeAs weight="...">residential</subscriberTypeAs>
```

In this example, the **subscriberTypeAs** is an enumerated that can take the value **residential**, **organization** or **government**.

An **Assertion** type element has the **weight** XML attribute, which may affect the result of the evaluation (see 9.9).

7.1.6.4 Directory substrings/word matching definitions

A separate version of the directory attribute definition is used for defining elements of the `Substrings` substitution group. The suffix `Sub` is added to the element name of the corresponding basic directory attribute definition used to distinguish it from that definition.

Figure 5 illustrates the XML schema for an XML element of the `Substrings` substitution group. It consists of a global XML element with a name following the rules indicated above and it has as child elements one or more `value` elements.

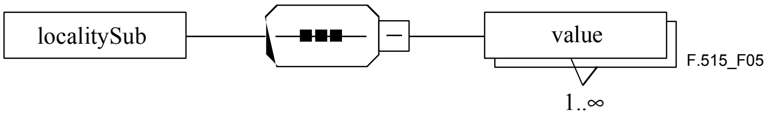

Figure 5/F.515 – Typical substrings definition

The content of each `value` child element is a word, possibly truncated.

The global element has two XML attributes:

– The `weight` XML attribute has the same significance as for an element of the `Assertion` substitution group.

– The `string` attribute indicates the string matching rule as requested by the client (see 9.2.4/F.510).

A `value` element has two XML attributes:

– The `wordMatch` XML attribute indicates the word matching rule as requested by the client (see 9.2.3/F.510).

– The `characterMatch` XML attribute indicates the character matching rule as requested by the client (see 9.2.2/F.510).

Details on matching rule specifications can be found in 9.2.

7.1.7 Defined directory attribute types

This Recommendation defines directory attribute types in 7.2 and 7.5 corresponding to the attribute types defined in ITU-T Rec. F.510.

The directory attributes are listed in tables where the columns have the following meanings:

– The Directory attribute type column gives the name of the directory attribute type, which is also the name of the XML element type representing the directory attribute type.

– The Syntax column gives the syntax used for a directory attribute value of the type.

– The Match type column indicates what type of filter item is used for the directory attribute type. 'As' means that the element of the `Assertion` substitution group has been defined for the directory attribute type. 'Sub' means that the element of the `Substrings` substitution group has been defined for the directory attribute type. '–' means that matching is not relevant for the attribute type.

– The Language attribute column indicates whether the `xml:lang` XML attribute has been included in the definition of the base directory attribute type. 'Y' means it has been included, 'N' means it has not been included and '–' means that a language code is not relevant.

– The Use column gives a little explanation on the use of the directory attribute type.

7.2 Geography

See Table 1.

Table 1/F.515 – Geography Locality directory attributes

Directory attribute type	Syntax (XML type)	Match type	Language attribute	Use
Country	NMTOKEN	As	–	Specifies the ISO 3166-1 code for the country.
Locality	Ub128	Sub	Y	Specifies the name of a locality.
stateOrProvince	Ub128	Sub	Y	Specifies the name of a state or province.
description	Ub1024	–	Y	Specifies free-form text that describes the Returned Object.
localityCode	Ub64	As	–	Specifies a common code to identify a geographical locality.
localityNDC	Num-ub16	As	–	Specifies the value of a field within the ITU-T Rec. E.164.

Table 1/F.510 lists the directory attributes corresponding to the attribute types as they are defined in 7.2/F.510. The ITU-T Rec. F.510 attribute type NoSubscriberInformation is indirectly signalled by the `returnedObject` XML attribute of the `entry` element having a value different from `subscriber`.

7.3 Business classification

No specification beyond ITU-T Rec. F.510 is needed.

7.4 Organizational structure

No specification beyond ITU-T Rec. F.510 is needed.

7.5 Subscriber entry

See Table 2.

Table 2/F.515 – Subscriber directory attributes

Directory attribute type	Syntax (XML type)	Match type	Language attribute	Use
streetAddress	Ub128	Sub	Y	Specifies a site for the local distribution and physical delivery in a postal address, i.e., the street name, place, avenue, etc.
houseId	Ub64	Sub	N	Specifies a linguistic construct used to identify a particular building, for example a house number or house name relative to a street, avenue, town or city, etc.
givenName	Ub64	Sub	N	Specifies the linguistic construct which is normally given to an individual by the individual's parent, or is chosen by the individual, or by which the individual is commonly known.

Table 2/F.515 – Subscriber directory attributes

Directory attribute type	Syntax (XML type)	Match type	Language attribute	Use
title	Ub64	Sub	Y	Specifies the designated position or function of the object within an organization.
description	Ub1024	–	Y	Specifies free-form text that describes the returned subscriber entry.
businessCategory	Ub128	Sub	Y	Specifies information concerning the occupation of some common objects, e.g., people. For example, this directory attribute provides the facility to interrogate the directory about people sharing the same occupation.
postalCode	Ub40	Sub	–	Specifies the postal code of the named object.
postOfficeBox	Ub40	Sub	–	Specifies the Post Office Box by which the object will receive physical postal delivery.
dmdName	Ub64	Sub		Specified a service provider name that has to be unique within a certain scope, e.g., within a country or Europe.
localityCode	Ub64	As	–	Specifies a common code to identify a geographical locality.
nationalDestination Code	Num-ub16	As	–	Specifies the value of a field within ITU-T Rec. E.164.
subscriberName	Ub128	Sub	Y	Specifies the name by which the Subscriber chooses to be known in the directory. The **`subscriberName`** is the surname for a residential subscriber and organizational name for an organizational or governmental subscriber.
subscriberType (Note)	SubscTypes	As	–	Specifies the type of subscriber. It is defined as a multi-valued directory attribute type allowing a subscriber to be simultaneously of multiple types.
secondFamilyName	Ub128	Sub	N	Specifies a residential subscriber when **`subscriberName`** is not sufficiently unique
profession	Ub128	Sub	Y	Specifies the profession or job of an individual, business or department.
language	language	As	–	Specifies the preferred language in which the subscriber entry shall be displayed.
NOTE – The XML syntax is a list of enumerated values. It is therefore possible to add several values to a single XML attribute element.				

7.5.1 Subscriber types

The `subscriberType` directory attribute gives the type of the subscriber. The different subscriber types are described in 7.5.1/F.510. The directory attribute syntax is the `SubscrTypes` simple XML type. It is a list item type derived from the enumerated `SubscrType` XML simple type. The `SubscrType` defines the following enumerated values:

– `residential`;

– `organization`;

– `government`.

It is possible to specify multiple subscriber types, e.g., in the case the same communications address is for both organizational and private use.

7.5.2 Communications address

See Table 3.

Table 3/F.515 – Communication address directory attributes

Directory attribute type	Syntax (XML type)	Match type	Language attribute	Use
telephone	Ub32	Sub	–	Specifies a telephone number associated with an object. A directory attribute value for Telephone Number is a string that complies with the internationally agreed format for showing international telephone numbers, ITU-T Rec. E.123 (e.g., "+ 44 582 10101").
orAddress	ub1024	Sub	–	Specifies a text encoding of an X.400 O/R address, as specified in RFC 987.
mail	ub256	Sub	–	Specifies an electronic mailbox directory attribute following the syntax specified in RFC 2822.
url	ub1024	Sub		Specifies a URL following the syntax specified in RFC 1738.
commService (See item 1 below)	ComServices	As	–	It specifies the type of service(s) associated with a communications address. It describes the class of service that the Communications Address provides access to, for example, telephone (voice), facsimile, electronic mail, SMS (short messaging service), EDI, file transfer, etc.
commNetwork	ComNetwork	As	–	It specifies the type of network for which a communications address is used. It describes the type of network where the Communications Address is allocated. For example, a Public Switch Telephone Network (PSTN), an ISDN network, or a GSM mobile phone network. It could also be an application oriented network, e.g., a banking network.

Table 3/F.515 – Communication address directory attributes

Directory attribute type	Syntax (XML type)	Match type	Language attribute	Use
addrValidFrom	dateTime	–	–	See item 2 below.
addrValidUntil	dateTime	–	–	See item 2 below.
addrValidity	AddrValidity Type	As	–	See item 2 below.
addrCoverage	AddrCoverage Type	As	–	The address coverage directory attribute type specifies area of validity of a communications address.
addrTariff	AddrTariff Type	As	–	The Address Tariff directory attribute type specifies what tariff is associated with a communications address.
addrRestriction	addrRestriction Type	As	–	The Address Restriction directory attribute type provides information about the restriction of access to the Communications Address information.

1) The XML syntax is a list of enumerated. It is therefore possible to add several values to a single XML attribute element.

2) If neither of the directory attribute types is present, the communications address is *valid* and *current*.

If the `addrValidFrom` directory attribute is present, but the `addrValidUntil` directory attribute is absent, and if the current date is equal or greater than the date given in the `addrValidFrom` directory attribute, then the communications address is *current*; otherwise it is *future*.

If the `addrValidUntil` attribute is present, but the `addrValidFrom` directory attribute is absent, and if the current date is less or equal than the date given in the `addrValidUntil` directory attribute, then the communications address is *current*; otherwise it is *old*.

If both the `addrValidFrom` directory attribute and the `addrValidUntil` directory attribute are present, and if the current date is greater or equal to the date given the `addrValidFrom` directory attribute and is less or equal to the date given in the `addrValidUntil` directory attribute, then the communications address is *temporary*. If the current date is less than the date given in the `addrValidFrom` directory attribute, then the communications address is *new*. If the current date is greater than the date given in the `addrValidFrom` directory attribute, then the communications address is *old*.

NOTE 1 – This assumes that the `addrValidFrom` value is less or equal to `addrValidUntil` value. If that is not the case, the specifications are invalid and useless.

The `addrValidity` is used primarily in the case where a database does not support inclusion of validity dates of communications addresses. If `addrValidFrom` and `addrValidUntil` directory attribute types are supported, the `addrValidity` directory attribute, if present, should be derived from those two other directory attribute types.

NOTE 2 – Even when validity dates are supported by the database, the `addrValidity` could be derived to support matching against this directory attribute type.

7.5.2.1 The Communications Service Type directory attribute

The `ComServiceTypes` simple XML type is a list item type derived from the enumerated `ComServiceType` XML simple type. The `ComServiceType` defines the following enumerated values:

– `voice`;

– `fax`:

  - `textPhone`;
  - `videoPhone`;
  - `publicPhoneBox`;
  - `switchBoard`;
  - `pager`;

– `data`;

– `email`;

– `web`.

7.5.2.2 Network Types simple XML type

The `NetworkType` type is an enumerated XML simple type. The following enumerated values are defined:

– `pstn`;

– `isdn`;

– `gsm`;

– `umts`;

– `internet`.

7.5.2.3 Address Validity Types simple XML type

The `AddrValidityType` is an enumerated XML simple type. The following enumerated values are defined:

– `current` indicates that the communications address is currently valid;

– `old` indicates that the communication address has been taken out of service;

– `future` indicates that the communication address has not been put into service; and

– `temporary` indicates that the communications address is currently valid, but its lifetime is limited.

7.5.2.4 Coverage Type simple XML type

The `addrCoverageType` is an enumerated XML simple type. The following enumerated values are defined:

– `international` indicates that the communications address has unlimited coverage.

– `national` indicates that the communications address is valid within the country of the subscriber.

– `stateOrProvince` indicates that the communications address is valid at least within the area indicated by the `stateOrProvince` directory attribute of the subscriber entry.

– `locality` indicates that the communications address is valid at least within the area indicated by the `localityName` directory attribute of the subscriber entry.

7.5.2.5 Tariff simple XML type

The `AddrTariffType` is an enumerated XML simple type. The following enumerated values are defined:

– `normal` indicates that standard tariff applies for the calling the number.

– `premium` indicates that some special tariff is used when calling the number.

– `toll-free` indicates that in general the called part pays for a call.

NOTE – New values may be introduced later. Implementations should be prepared for extensions.

7.5.2.6 Address Restriction simple XML type

The `AddrRestrictionType` is an enumerated XML simple type. The following enumerated values are defined:

– `public` indicates that the communications address can be revealed without restrictions;

– `secret` indicates that the communications address shall not be revealed;

– `call-screen`;

– `no-marketing`;

– `complete-only`.

The `call-screen` value indicates that an operator can call the subscriber to ask if it wants to receive the call, and then allow call completion. The communications address is not revealed.

The `no-marketing` value indicates that the communications address can be revealed, but it shall not be used for unsolicited marketing.

The `complete-only` value indicates that the communications address shall not be revealed, but the operator may put a call through for the inquirer.

NOTE – New values may be introduced later. Implementations should be prepared for extensions.

8 Service specification

No specification beyond ITU-T Rec. F.510 is needed.

9 Common Protocol Specification element

9.1 Character set, encoding and repertoire

The ISO/IEC 10646-1 character set with the UTF-8 encoding shall be used for encoding the UDAP.

The ISO/IEC 10646-1 character repertoire is constrained to the following subset of plane 00:

– Row 00, positions 20-7E (BASIC LATIN);

– Row 00, positions A0-FF (LATIN-1 SUPPLEMENT);

– Row 01, positions 00-13, 16-2B, 2E-4D, 50-7E (part of LATIN EXTENDED-A).

It is expected that the repertoire will be extended as directory services based on this Recommendation spread outside Europe.

To the extent that this repertoire cannot be supported by an implementation, it has to be clearly stated.

NOTE – Service providers may make mutual agreements that further limits the character repertoire.

9.2 Matching rules

Matching as defined in 9.2/F.510 shall be supported in the protocol as XML attributes.

The character mapping rule requirements shall be carried in the `characterMatch` XML attribute with the following enumerated values: `exact`, `caseIgnore` (default) and `mapped`. This XML attribute is associated with the `value` child element of an element of the `Substrings` substitution group and applied to the word provided in the `value` element.

The word matching rule requirements shall be carried in the `wordMatch` XML attribute with the following enumerated values: `exact` (default), `truncated` and `phonetic`. This XML attribute is associated with the `value` child element of an element of the `Substrings` substitution group and applied to the word provided in the `value` element.

The string matching rule requirements shall be carried in the `string` XML attribute with the enumerated values: `exact`, `deletion`, `restrictedDeletion`, `permutation`, `permutationAndDeletion` and `providerDefined`. This XML attribute is associated with an element of the `Substrings` substring group and applies to all the words supplied in the `value` child elements.

Only string type directory attributes types use the matching rules above. Other types of directory attribute types require, e.g., enumerated directory attribute types, use a simple equality match (see 7.1.3.2).

9.3 Language differences

A language specification can be associated with string directory attribute types. Tables 1 to 3 indicate what directory attribute types may include a language specification. The XML language attribute is used for specifying the language. This requires that the http://www.w3.org/2001/xml.xsd has to be imported and a reference to the namespace http://www.w3.org/XML/1998/namespace has to be included (see B.3).

9.4 Ordering of entries within a response

It is for further study.

9.5 Hierarchical groups

9.5.1 Hierarchical group concept

Entries may have hierarchical relationship. *Hierarchical groups* form such relationships by forming a logical tree with a root called the *hierarchical top*.

By referring to hierarchical relationships, it is possible in a search operation to retrieve information from entries matching a search filter, but also from other entries within the same hierarchical group.

To describe navigation within a hierarchical group it is convenient to define terms for the relationships that a given entry has with other entries within the group. An entry that is below some other entry in a hierarchical group is a *hierarchical child* of that other entry. If it is just below that other entry, it is also an *immediate hierarchical child* of that other entry. An entry that is above some other entry is a *hierarchical parent* of that other entry. If it is just above this other entry, it is an *immediate hierarchical parent* of that other entry. An entry that has no hierarchical children is a *hierarchical leaf*. A hierarchical top has no hierarchical parents.

9.5.2 Hierarchy selection

The hierarchy selection is specified in the `hierSelect` XML attribute of the `searchRequest` element. It may take the values `self` (default), `children`, `parent`, `hierarchy`, `top`, `subtree` and `all` with the semantics as described in 9.5/F.510.

The **hierSelect** XML attribute is a list XML attribute type allowing several hierarchy selections to be provided within the same XML attribute as illustrated by the following example:

```
... hierSelect="self top subtree"
```

9.5.3 Return of hierarchical information

Two XML attributes of the **entry** element are related to hierarchical groups:

– The **hierarchyLevel** XML attribute holds an integer that gives the level of the entry within the hierarchical group. The top entry has level 0, all immediate children to the hierarchal top have level 1, etc. This XML attribute shall be included for an entry in the response if the entry is member of a hierarchical group. Otherwise, it shall be absent.

– The **hierarchyBelow** XML attribute indicates whether the entry has one or more hierarchical children. If the entry has hierarchical children, this directory attribute shall be present with the value **true**. If the entry is a hierarchical leaf, it shall either be present with the value **false** or be absent. If the entry is not member of a hierarchical group, this XML attribute shall be absent.

When entries of a hierarchical group are returned, the start and end of the hierarchical group is signalled by the **hierarchy** element as indicated by below example.

```
<hierarchy entries="..." entryNumber="...">
  <entry hierarchyLevel="..." hierarchyBelow="..." entryIdentifier="...">
    <!-- directory attributes -->
  </entry>
  <!-- ..............-->
  <entry hierarchyLevel="..." hierarchyBelow="..." entryIdentifier="...">
    <!-- directory attributes -->
  </entry>
</hierarchy>
```

The **hierarchy** element shall contain one or more **entry** child elements for entries members of the same hierarchical group. It has two XML attributes:

a) **entries**, which signals how many entries that are to be returned for that hierarchical group. It shall be present if paged results are returned; otherwise, it is optional.

b) **entryNumber**, which signals the relative number of the first entry returned within the **hierarchy** element with respect to the first entry to be returned. The first entry is number one, which is also the default. It shall be present if page results are returned and the page starts in the middle of the returned hierararchical group.

9.5.4 Sequential ordering of a hierarchical group

When transmitting a hierarchical group, a sequential ordering rule is required. The sequential order of a hierarchical group comes from following all the strands of the hierarchical group as follows:

a) The top entry is the first entry in the sequence follow by the remaining entries within a complete strand going down from top to a hierarchical leaf. It is a local choice which strand to select as the first one.

b) The next strand to be selected is one that has not previously been selected and which has the maximum number of entries common with the previous selected strand. If several strands are identical in that respect, selection is a local matter. Only those entries not previously included are included in the sequence.

c) The procedure in b) is repeated until all strands have been included.

The rule above also applies when only part of a hierarchical is returned, except it may leave holes in the sequence. If the hierarchical top is returned, it will always be the first one returned. Otherwise, the first one returned is dependent on which strand is selected as the initial one according to the algorithm above.

9.6 Number of entries

The client can specify a limit for the maximum number of entries to be returned. It is specified as a positive integer in the **entryLimit** XML attribute of the **searchRequest**.

9.7 Entry count

The entry count of the response is given in the **entryCount** element of the **searchResult**. The **entryCount** element is an empty element with two XML attributes:

– **count**, which provides the returned entry count; and

– **qualifier**, with the choice of the two values, where the **exact** choice shall be taken if the directory server can supply an exact count; otherwise, the **bestEstimate** choice shall be taken.

9.8 Alternative directory attribute values

This concept is not reflected in the protocol. For further explanation, see 9.8/F.510.

9.9 Weighted directory attribute values

Each **Assertion** type element and each **Substrings** type element has a **weight** XML attribute that can take the values **high** (default) and **low** corresponding to AttributeWeight one and zero as defined in 9.9/F.510.

9.10 Geographical extensions

The search controls for geographical extensions are defined in 9.10/F.510. The Extended Area control is provided as the **extendedArea** XML attribute of the **searchRequest** element. The Include All Areas control is provided as the **includeAllAreas** value of the **searchOptions** XML attribute of the **searchRequest** element.

9.11 Ignore if absent

This concept is not reflected in the protocol. For further explanation, see 9.11/F.510.

9.12 Paged Results request

```
<xsd:complexType name="PagedResults">
  <xsd:attribute name="pageSize" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="pageNumber" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="queryReference" type="xsd:hexBinary" use="optional"/>
</xsd:complexType>
```

The **PagedResults** complex XML type allows the client to ask for piecewise return of the result. In the initial request, the client specifies page size and a possible page number, if the client requires a page different from the first page. To retrieve more pages, the client has to issue a new request for each page to be returned.

A directory server may operate in two different modes:

a) The server may keep status information about the operation progress. To operate in this mode, the server shall return a query reference to be used in subsequent request. In subsequent requests, the requester only needs to send the query reference to retrieve the next page.

b) The server keeps no status information and returns no query reference. The requestor then has to explicitly specify a page number to get anything but the first page returned.

A directory server may also operate in a hybrid mode, supporting both of the above modes.

NOTE – This may occur when the directory server actually consists of several identical servers for load sharing. If a subsequent request gets to a server that has previously served this operation, then the query reference would be usable; otherwise, the page number will be used. For this to work properly, the client has to include both **`pageNumber`** and **`queryReference`**.

The **`pageSize`** XML attribute indicates the number of entries that shall be returned by each paged result. This is a mandatory XML attribute when paged results are requested.

The **`pageNumber`** XML attribute indicates the number of the page to be returned. If this XML attribute is absent, then:

– the first page shall be returned if a **`queryReference`** XML attribute is not supplied or if the directory server does not support the **`queryReference`**;

– the next page shall be returned number if the **`queryReference`** is present and supported.

The **`queryReference`** shall not be present in the initial request. However, the directory server may in the reply to a paged result request return a **`queryReference`** that may be used in subsequent requests.

10 The Unified Directory Access Protocol (UDAP)

10.1 Service model

No specification beyond ITU-T Rec. F.510 is needed.

10.2 Search controls and indications

The SearchType search control from ITU-T Rec. F.510 is provided by the **`searchType`** XML element of the SOAP **`Header`** element (see C.1.1).

The UserClass search control from ITU-T Rec. F.510 is provided by the **`userClass`** XML element of the SOAP **`Header`** element (see C.1.1).

The EntryLimit search control from ITU-T Rec. F.510 is provided by the **`entryLimit`** XML attribute of the **`searchRequest`** (see 9.6).

The ExtendedArea search control from ITU-T Rec. F.510 is provided by the **`extendedArea`** XML attribute of the **`searchRequest`** (see 9.10).

The IncludeAllAreas and PerformExactly search controls from ITU-T Rec. F.510 are provided by the **`searchOption`** XML list attribute allowing the values **`includeAllAreas`** (see 9.10) and **`performExactly`**.

The EntryCount respond indicator from ITU-T Rec. F.510 is provided by the **`entryCount`** child element of the **`searchResult`**.

10.3 Predicates

No specification beyond ITU-T Rec. F.510 is needed.

10.4 Common query conditions

The common query search controls and result indicators as listed in Table 6/F.510 are specified in 10.2.

10.5 Basic service offering

10.5.1 Search request

```
<searchRequest entryLimit="..."
    extendedArea="..." searchOptions="..." hierSelect="..."
  <base entryIdentifier="..."/>
  <filter>
  </filter>
  <infoSelect>
  </infoSelect>
  <pagedResults>
  </pagedResults>
</searchRequest>
```

10.5.1.1 XML attributes of the `searchRequest` element

For the **`hierSelect`** XML attribute, see 9.5.2. For the **`searchOptions`** XML attributes, see 10.2.

10.5.1.2 XML child elements of the `searchRequest` element

The **`base`** is an empty element having a single **`entryIdentifier`** mandatory XML attribute. This element is mandatory for the **`subscriberByEntryIdentifier`** search-type. It is not relevant for other search-types.

The **`filter`** element is described in 7.1.6.

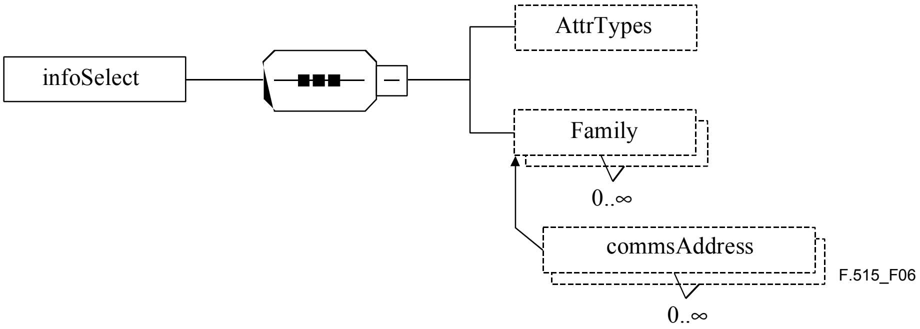

Figure 6/F.515 – Information selection

The **`infoSelect`** element allows a client to select what information (families and directory attribute types) is to be returned from each entry.

Only directory attributes of the types specified in the **`attrTypes`** element may be returned independent on whether they are in the main part of the entry or part of a family. Directory attributes of types not listed should not be returned. Not all directory attributes requested by the client may be returned, as some of the requested directory attributes may not be considered part of the service.

The client can also limit the families to be returned. This is done using one or more elements of the **`Family`** substitution group holding one or more directory attributes. Such an element is compared with all the families of the entry of that type. If the family holds the same attribute types as included in the element of the request and the value matches, then the family shall be returned; otherwise it should not. If an attribute in the element of the request is empty, then the attribute matches independently of the value of the stored directory attribute.

If the directory server does not support the **`infoSelect`** element or part of it, it shall be ignored what is not supported and the request shall be processed accordingly.

The **`pagedResults`** element is of type **`PagedResults`** as described in 9.2.

10.5.2 Search result

10.5.2.1 Search result schema

```
<searchResult requestRef=" ... " msgCode=" ... " queryReference=" ... ">
  <information>
    <hierarchy>
      <entry entryIdentifier= hierarchyLevel=" ... " hierarchyBelow=" ... "
            returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
      </entry>
      <entry hierarchyLevel=" ... " hierarchyBelow=" ... " returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
      </entry>
        <!--               - - -               -->
      <entry hierarchyLevel=" ... " hierarchyBelow=" ... " returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
      </entry>
    </hierarchy>
    <entry hierarchyLevel=" ... " hierarchyBelow=" ... " returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
    </entry>
    <entry hierarchyLevel=" ... " hierarchyBelow=" ... " returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
    </entry>
      <!--               - - -               -->
    <entry hierarchyLevel=" ... " hierarchyBelow=" ... " returnedObject=" ... ">
        <!-- ............ Directory attribute ............ -->
        <!-- ............ Directory attribute ............ -->
        <!--               - - - - - - - -                -->
        <!-- ............ Directory attribute ............ -->
    </entry>
  </information>
  </entryCount count=" ... " qualifier=" ... "
  <notification>
      <!-- ..... Notification attribute ...... -->
      <!-- ..... Notification attribute ...... -->
      <!--            - - -                    -->
      <!-- ..... Notification attribute ...... -->
  </notification>
```

10.5.2.2 XML attributes of the searchResult element

The value of the **msgCode** XML attribute shall be used for a message code value as defined by ITU-T Rec. F.510. It is mandatory in an ITU-T Rec. F.510 environment for situations where a message code is defined. Otherwise, it is optional (see Annex A).

For the **queryReference** XML attribute, see 9.12.

10.5.2.3 Information element

The **information** element shall contain one or more **entry** (see 7.1.2) and/or **hierarchy** (see 9.5.3) child elements. If no information is returned in the **searchResult**, the **information** element shall be absent.

10.5.2.4 Notifications element

This element shall contain one or more notification attributes. The conditions for return of notifications attributes are specified in Annex A.

10.6 Enhanced service offering

The same XML specification applies for both basic and advanced service offering (see 10.5).

11 Operational issues

11.1 Security

This Recommendation does not specify any features. It is assumed that security will be provided by an underlying service.

12 Charging and accounting

This Recommendation does not have any special provision for charging and accounting.

13 Quality of Service

This Recommendation does not have any special provision for Quality of Service.

Annex A

Message codes and notifications

A.1 Notification attribute types

Notification attributes are carried in the `notifications` element of the `searchResult`. They provide information to the user that can help rectify problems in a `searchRequest`.

Notification attribute type	Syntax (XML type)	Use
limitProblem	NCName	The `limitProblem` notification attribute shall be included if the directory server encounters some limit problem. It is defined as an enumerated. It can take the following values: – The `adminLimit` value is used if the directory server has reached some limit set by an administrative authority. – The `permanentRestriction` is used if an operation has caused the directory server to exceed some limit that causes the process to stop and a repeated operation is judged to encounter the same problem. – The `temporaryRestriction` is used if an operation has caused the directory server to exceed some limit that causes the process to stop, but the reason is judged to be a temporary problem, e.g., resources depletion.
serviceProblem	NCName	It describes problems relating to the service provider policy for the service in question.

Notification attribute type	Syntax (XML type)	Use
searchType	SearchType	It gives the search-type of the failing search.
attributeTypeList	AttributeTypes	It gives a list of directory attribute types related to a specific return code.
filterNot	Filter	It gives a filter specification (e.g., see message code 6.3 in A.1).
filterItem	complex	It gives a list of invalid filter items in a search filter.
providerName	ProviderId	
hierarchySelectList	HierarchySelections	It gives a list identifying one ore more hierarchy selection options. When a value is set, it indicates that the corresponding hierarchy selection is invalid. Either a forbidden or unsupported selection has been requested, or the selection has not been requested when it is required.
searchControlOptionsList	SearchOptions	It gives a list identifying one ore more search control options. When a value is set, it indicates that the corresponding search control option selection is invalid. Either a forbidden or unsupported option has been requested, or the option has not been requested when it is required.
attributeCombination	Filter	It gives a list of attribute combinations that were missing in a search filter.
wordRestriction	complex	It specifies word matching features not supported by the service provider and/or it specified minimum string length requirements.
notSupported	Options	It gives a list of optional functions not supported by the service provider. The directory server has made a best effort. This notification attribute shall be returned whenever one or more known facilities are not supported.

A.2 Message codes

F.510 Service Definition Message code	Description	Associated notifications
0 Unified directory specific codes		
0.1 Invalid entry identifier	The search cannot be performed as the entry identifier is not recognized because it is either invalid or expired.	None.
0.2 Invalid query reference	The search cannot be performed as the entry query reference is not recognized because it is either invalid or expired.	None.
1 Unavailable service access		
1.1 System congestion	The search cannot be performed, as the directory server is congested.	Optionally, a `providerName` notification attribute may be returned.

F.510 Service Definition Message code	Description	Associated notifications
1.2 Destination database not accessible	The search cannot be performed, as system that should perform the search cannot be reached.	Optionally, a **providerName** notification attribute may be returned.
1.3 Destination database not accessible	The search cannot be performed as the data on which the search should be performed cannot be reached.	Optionally, a **providerName** notification attribute may be returned.
2 Required service not supported by the service provider		
2.1 Search for service <Search_Type> is not supported	The service provider does not support the requested search-type.	A **searchType** notification attribute having the search-type as the value shall be returned.
2.2 Search for service <Search_Type> supported but not bilaterally agreed	The service provider supports the requested search-type, but the search-type has not been bilaterally agreed.	A **searchType** notification attribute having the search-type as the value shall be returned.
3 Required functionality not supported by the service provider		
3.1 Matching rule <Identifier> not supported by service provider for the whole service	A requested matching rule is or certain aspects of a matching rule are unknown to the service provider, as it is not supported by the directory server. Either **performedExactly** is specified in the request or no alternative matching has been defined by the directory server.	When an implementation does not support certain aspects of a matching, the following is returned: – a **wordRestriction** notification attribute having as value a specification of the matching rules specified in the request that are not supported. An directory attribute type shall not be specified.
3.2 Matching rule <Identifier> not supported by Service Provider for this specific <Attribute_Type>	The requested matching rule and assertion with the specified directory attribute type are recognized and supported by the service provider, but as a policy, the service provider do not allow the requested matching in the particular situation. In addition, the **performExactly** is set or the service provider has not provided for alternative matching.	In case of an unsupported matching rule option for a particular directory attribute type, the following is returned: – one or more **wordRestriction** notification attributes each indicating an directory attribute type and a specification of the matching rule(s) specified in the request that are not supported for the directory attribute type.
3.3 Inappropriate matching rule for <Attribute_Type>	The requested matching rule is recognized and supported by the service provider but is not appropriate for the particular directory attribute type. However, this situation will result in an invalid XML document (see 0.2).	None.

F.510 Service Definition Message code	Description	Associated notifications
3.4 Invalid combination of matching rules for <Attribute_Type>	The specified combination of matching rules (String matching rule, word matching rule and character matching rule) is not supported by the service provider as expressed by applicable search-rule. In addition, the **`performExactly`** is set or the service provider has not provided for alternative matching.	In case of an unsupported combination of matching rule options for a particular directory attribute type, the following is returned: – one or more **`wordRestriction`** notification attributes s each indicating a directory attribute type and the invalid combination.
3.5 Geographical extension not supported	The geographical extension functionality is not supported by the directory server.	None.
3.6 Level of geographical extension not supported	The requested level of geographical extension functionality is not supported by the directory server.	None.
3.7 Service relative to hierarchical group with the specified HierarchySelection value not supported	The requested hierarchy selection value is not supported by the directory server.	UDS recognizes three situations: 1) Hierarchy selection specification (except for **`self`**) is not allowed for the type of service. If then included, the request is rejected with a **`serviceProblem`** notification attribute with the value **`hierSelectForbidden`**. 2) If one or more hierarchy selections are forbidden by the service provider, then the request is rejected with a **`serviceProblem`** notification attribute with the value **`hierSelectNotAvailableForService`** together a **`hierarchySelectList`** notification attribute indicating the invalid settings. 3) If one or more hierarchy selection options are not supported by the implementation and those options are not handled by the cases above, the request is rejected with a **`serviceProblem`** notification attribute with the value **`hierSelectNotSupported`** together a **`hierarchySelectList`** notification attribute indicating the settings in the requests that are unsupported.

F.510 Service Definition Message code	Description	Associated notifications
3.8 Filter on <Attribute_Type> for service <Search_Type> not supported	The search cannot be performed by the directory server, as one or more of the directory attribute types in the filter cannot be used for the given search (e.g., filtering by street address for a country wide search not supported by service provider).	The directory server will return: – an **attributeTypeList** notification attribute having as values the illegal directory attribute types; – a **searchType** notification attribute with the search-type as value.
3.9 Filter on <Attribute_Type, Attribute_Value> for service <Search_Type> not supported	The search cannot be performed by the directory server, as a value of one or more directory attribute types in the filter cannot be used for the given search.	The directory server shall return: – a **filterItem** notification attribute with the failing filter items as value; and – a **searchType** notification attribute with the search-type as value.
3.10 <Attribute_Type> required by service provider to perform the Search	Equivalent to 4.1.	Equivalent to 4.1.
3.11 The value of the <UserClass> parameter is not bilaterally agreed	The search cannot be performed as the directory server does not support the specified **userClass**.	None.
4 Wrong input information		
4.1 Mandatory <Attribute_Types> are missing for requested service <Search_Type>	The search cannot be performed as one or more mandatory directory attributes are missing.	The directory server shall return: – an **attributeTypeList** notification attribute having as values the missing directory attribute types; – a **searchType** notification attribute with the search-type as value.
4.2 Minimum size of truncated word is required for <Attribute_Type>	The search cannot be performed as the minimum number of characters for word truncation to be applied on the specified directory attribute is not respected.	The directory server shall return: – one or more a **wordRestriction** notification attributes indicating the directory attribute type and what the minimum string length are for word matching for that directory attribute type. Two values are given, one for the first word in the sequence and one for the following words; – a **searchType** notification attribute with the search-type as value.

F.510 Service Definition Message code	Description	Associated notifications
4.3 Mandatory combination of <Attribute_Types> are missing for requested service <Search_Type>	The search cannot be performed, as a required combination of directory attribute types is missing. As an example, a service provider may require that either `givenName` or `businessCategory` is specified in a search request, but the user has supplied neither.	The directory server shall return: – an `attributeCombinations` notification attribute identifying the missing combination(s); – a `searchType` notification attribute with the search-type as value.
4.4 Invalid value for <Attribute_Type> because of incorrect syntax (e.g., no figures for a number)	The search cannot be performed as the syntax used for one of the directory attribute is not valid. The directory machinery is not able to process the request.	Covered by 0.2.
4.5 Unrecognized value for <attribute type> or <common parameter> *Unrecognized value for <Attribute_Type> or <Common_Query_Parameter> at the application level (e.g., GroupSelection parameter, DeviceTariff attribute, DeviceService attribute, DeviceAdressType attribute)*	The search cannot be performed as the value provided for a specified parameter does not fit with one value from the set of predefined F.510 values.	Covered by 0.2.
5 Undetermined geographical area		
5.1 The parameter <Attribute_Type, Attribute_Value> does not exist	The search cannot be performed as the value provided for a key directory attributes does not exist.	The directory server will return a `filterItem` notification attribute having the failing filter item as value.
5.2 The parameter <Attribute_Type, Attribute_Value> is not unique	The search cannot be performed as the value provided for a key directory attribute is ambiguous.	None.
5.3 The combination of parameters <Attribute_Type, Attribute_Value>, <Attribute_Type, Attribute_Value>, <Attribute_Type, Attribute_Value>, etc., does not exist	The search cannot be performed as values provided for each geographical locality are in conflict (e.g., the locality does not exist under the specified Province).	The directory server will return a `filterItem` notification attribute having the failing filter items as values.

F.510 Service Definition Message code	Description	Associated notifications
6 No entries returned		If relevant, a **`notSupported`** notification attribute may be returned specifying unsupported facilities.
6.1 No subscriber entries are available under geographical area <Attribute_Type, Attribute_Value>	No entries are provided due to the fact that for the geographical area in which the search should be performed, the requested service provider does not have any entries in its databases (e.g., it corresponds for instance to an area in the US where independent companies are operating and no electronic directory is available).	An empty result is returned.
6.2 No subscriber entries have been found for the requested HierarchySelection value	No entries are provided. Entries would be returned if the HierarchySelection parameter value is relaxed.	An empty result is returned.
6.3 No subscriber entries have been found but entries exist without <Attribute_Value> constraint for <Attribute_Type>	No entries are provided. Entries would be returned if the constraint on <Attribute_Value> for the given directory attribute type is relaxed.	A result is returned without the **`information element,`** optionally including a **`filterNot`** notification attribute with a filter specification consisting of the filter supplied in the request with those filter items that that succeeded in matching some entries omitted.
6.4 No entries have been found but entries exist without <Attribute_Type> constraint	No entries are provided. Entries would be returned if the constraint on <Attribute_Type> is relaxed.	As for 6.3.
6.5 Too many entries selected: more selective information is necessary	The search cannot be performed as the scope of it is considered to be too large by the service provider (according to local knowledge). More information is requested from the user to allow the search to be applied.	Same as for 4.1.
6.6 No entry matched the search filter	The search was performed, but no entry matched the filter.	None.
6.7 The requested page is not available	The search was performed. However, the client asked for a page number higher than the number of pages returned by the search.	None.

F.510 Service Definition Message code	Description	Associated notifications
7 List of non-subscriber attributes		If relevant, a **notSupported** notification attribute may be returned specifying unsupported facilities.
7.1 Incomplete list of <returnedObject>, more available	An administrative limit has been reached.	The directory server returns a **limitProblem** notification attribute with the value adminLimit if an administrator-imposed limit has been reached, and with **permanentRestriction** if a permanent implementation restriction has been reached.
7.2 Incomplete list of <returnedObject>, no more available	An administrative limit has been reached.	As for 7.1.
8 Entries returned		If relevant, a **notSupported** notification attribute may be returned specifying unsupported facilities.
8.1 Entries <Returned_Object> found	Either all found entries are returned or the last entries of paged results are returned	None.
8.2 Entries <Returned_object> found, more entries available	A page-full of entries are returned. There are still more entries available.	None.

Annex B

XML schema for UDAP

B.1 Search request

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.itu.int/itu-t/Rec/f515/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:uds="http://www.itu.int/
itu-t/Rec/f515/xsd" elementFormDefault="unqualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
 schemaLocation="xml.xsd"/>
  <xs:include schemaLocation="attributeType-v01.xsd"/>
  <xs:complexType name="PagedResults">
    <xs:attribute name="pageSize" type="xs:positiveInteger" use="optional"/>
    <xs:attribute name="pageNumber" type="xs:positiveInteger" use="optional"/>
    <xs:attribute name="queryReference" type="xs:base64Binary" use="optional"/>
  </xs:complexType>
  <xs:element name="searchRequest">
    <xs:annotation>
      <xs:documentation>Unified Directory Search Request</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="base" minOccurs="0">
          <xs:complexType>
            <xs:attribute name="entryIdentifier" type="xs:base64Binary" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="filter" type="uds:Filter" minOccurs="0"/>
        <xs:element name="infoSelect" minOccurs="0">
```

```
          <xs:complexType>
            <xs:sequence>
              <xs:element name="attrTypes" type="uds:AttributeTypes" minOccurs="0"/>
              <xs:element ref="uds:Family" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="pagedResults" type="uds:PagedResults" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="entryLimit" type="xs:positiveInteger" use="optional"/>
      <xs:attribute name="extendedArea" type="xs:nonNegativeInteger" use="optional"
          default="0"/>
      <xs:attribute name="searchOptions" type="uds:SearchOptions" use="optional"/>
      <xs:attribute name="hierSelect" type="uds:HierarchySelections" use="optional"
          default="self"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="reqHead">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="requestRef" type="xs:integer"/>
        <xs:element name="searchType" type="uds:SearchType"/>
        <xs:element name="userClass" default="operator" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:NCName">
              <xs:enumeration value="operator"/>
              <xs:enumeration value="publicUser"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B.2 Search result

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.itu.int/itu-t/Rec/f515/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:uds="http://www.itu.int/
itu-t/Rec/f515/xsd" elementFormDefault="unqualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd"/>
  <xs:include schemaLocation="attributeType-v01.xsd"/>
  <xs:complexType name="Entry">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="uds:Attribute"/>
        <xs:element ref="uds:Family"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="entryIdentifier" type="xs:base64Binary" use="optional"/>
    <xs:attribute name="returnedObject" use="optional" default="subscriber">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="subscriber"/>
          <xs:enumeration value="stateOrProvince"/>
          <xs:enumeration value="locality"/>
          <xs:enumeration value="street"/>
          <xs:enumeration value="businessCategory"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="hierarchyLevel" type="xs:nonNegativeInteger" use="optional"/>
    <xs:attribute name="hierarchyBelow" type="xs:boolean" use="optional"
            default="false"/>
  </xs:complexType>
  <xs:element name="searchResult">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="performer" type="uds:ProviderId" minOccurs="0"/>
        <xs:element name="information" minOccurs="0">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded">
              <xs:element name="hierarchy">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="entry" type="uds:Entry" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="entries" type="xs:positiveInteger"
use="optional"/>
                  <xs:attribute name="firsteEntry" type="xs:positiveInteger"
                      use="optional" default="1"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="entry">
                <xs:complexType>
                  <xs:complexContent>
                    <xs:extension base="uds:Entry">
                      <xs:attribute name="hierSeq" type="xs:integer" use="optional"/>
                    </xs:extension>
                  </xs:complexContent>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="entryCount" minOccurs="0">
          <xs:complexType>
            <xs:attribute name="count" type="xs:positiveInteger"/>
            <xs:attribute name="qualifier">
              <xs:simpleType>
                <xs:restriction base="xs:NCName">
                  <xs:enumeration value="exact"/>
                  <xs:enumeration value="bestEstimate"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="notifications" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="uds:Notification" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="msgCode" type="xs:decimal" use="optional"/>
      <xs:attribute name="queryReference" type="xs:base64Binary" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="resHead">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="requestRef" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="explanation" type="xs:string"/>
</xs:schema>
```

B.3 Attribute Types

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.itu.int/itu-t/Rec/f515/xsd"
xmlns:uds="http://www.itu.int/itu-t/Rec/f515/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified"
attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd"/>
  <xs:complexType name="ProviderId">
    <xs:sequence>
      <xs:element ref="uds:country"/>
      <xs:element ref="uds:dmdName" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NumericString">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub16NumericString">
    <xs:restriction base="xs:string">
      <xs:maxLength value="16"/>
      <xs:pattern value="[0-9]*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub32">
    <xs:restriction base="xs:string">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub40">
    <xs:restriction base="xs:string">
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub64">
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub128">
    <xs:restriction base="xs:string">
      <xs:maxLength value="128"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub256">
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub512">
    <xs:restriction base="xs:string">
      <xs:maxLength value="512"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Ub1024">
    <xs:restriction base="xs:string">
      <xs:maxLength value="1024"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AttributeType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="country"/>
      <xs:enumeration value="locality"/>
      <xs:enumeration value="stateOrProvince"/>
      <xs:enumeration value="streetAddress"/>
      <xs:enumeration value="houseId"/>
      <xs:enumeration value="givenName"/>
      <xs:enumeration value="title"/>
      <xs:enumeration value="description"/>
      <xs:enumeration value="businessCategory"/>
      <xs:enumeration value="postalCode"/>
      <xs:enumeration value="postOfficeBox"/>
      <xs:enumeration value="dmdName"/>
      <xs:enumeration value="LocalityCode"/>
      <xs:enumeration value="localityNDC"/>
      <xs:enumeration value="subscriberName"/>
      <xs:enumeration value="subscriberType"/>
      <xs:enumeration value="secondFamilyName"/>
      <xs:enumeration value="profession"/>
```

```
      <xs:enumeration value="language"/>
      <xs:enumeration value="telephone"/>
      <xs:enumeration value="orAddress"/>
      <xs:enumeration value="mail"/>
      <xs:enumeration value="url"/>
      <xs:enumeration value="commService"/>
      <xs:enumeration value="commNetwork"/>
      <xs:enumeration value="addrValidFrom"/>
      <xs:enumeration value="addrValidUntil"/>
      <xs:enumeration value="addrValidity"/>
      <xs:enumeration value="addrCoverage"/>
      <xs:enumeration value="addrTariff"/>
      <xs:enumeration value="addrRestriction"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AttributeTypes">
    <xs:list itemType="uds:AttributeType"/>
  </xs:simpleType>
  <xs:simpleType name="SearchType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="searchForStateOrProvince"/>
      <xs:enumeration value="searchForLocality"/>
      <xs:enumeration value="subscriberWithinLocality"/>
      <xs:enumeration value="subscriberByEntryIdentifier"/>
      <xs:enumeration value="subscriberWithinStateOrProvince"/>
      <xs:enumeration value="subscriberWithinCountry"/>
      <xs:enumeration value="searchForStreetAddress"/>
      <xs:enumeration value="subscriberByStreetAddress"/>
      <xs:enumeration value="subscriberByCommunicationsAddress"/>
      <xs:enumeration value="subscriberByBusinessCategory"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="HierarchySelection">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="self"/>
      <xs:enumeration value="children"/>
      <xs:enumeration value="parent"/>
      <xs:enumeration value="hierarchy"/>
      <xs:enumeration value="top"/>
      <xs:enumeration value="subtree"/>
      <xs:enumeration value="all"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="HierarchySelections">
    <xs:list itemType="uds:HierarchySelection"/>
  </xs:simpleType>
  <xs:simpleType name="SearchOption">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="performExactly"/>
      <xs:enumeration value="includeAllAreas"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="SearchOptions">
    <xs:list itemType="uds:SearchOption"/>
  </xs:simpleType>
  <xs:complexType name="Filter">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="uds:Assertion"/>
      <xs:element ref="uds:Substrings"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="Attribute" abstract="true"/>
  <xs:element name="Family" abstract="true"/>
  <xs:element name="Assertion" abstract="true"/>
  <xs:attributeGroup name="assertionAttr">
    <xs:attribute name="weight" use="optional" default="high">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="low"/>
          <xs:enumeration value="high"/>
        </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:element name="Substrings" abstract="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="value" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attributeGroup ref="uds:substringValueAttr"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="uds:substringAttr"/>
  </xs:complexType>
</xs:element>
<xs:attributeGroup name="substringAttr">
  <xs:attribute name="string" use="optional" default="exact">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="exact"/>
        <xs:enumeration value="deletion"/>
        <xs:enumeration value="restrDeletion"/>
        <xs:enumeration value="permutation"/>
        <xs:enumeration value="permutationAndDeletion"/>
        <xs:enumeration value="providerDefined"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="weight" use="optional" default="high">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="low"/>
        <xs:enumeration value="high"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="substringValueAttr">
  <xs:attribute name="wordMatch" use="optional" default="exact">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="exact"/>
        <xs:enumeration value="truncated"/>
        <xs:enumeration value="phonetic"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="characterMatch" use="optional" default="caseIgnore">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="exact"/>
        <xs:enumeration value="caseIgnore"/>
        <xs:enumeration value="mapped"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:element name="commsAddress" substitutionGroup="uds:Family">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="uds:Attribute" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="country" type="xs:NMTOKEN" substitutionGroup="uds:Attribute"/>
<xs:element name="countryAs" substitutionGroup="uds:Assertion">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:NMTOKEN">
        <xs:attributeGroup ref="uds:assertionAttr"/>
      </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="locality" type="uds:Ub128" substitutionGroup="uds:Attribute"/>
  <xs:element name="localitySub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" type="uds:Ub128" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="stateOrProvince" type="uds:Ub128" substitutionGroup="uds:Attribute"/>
  <xs:element name="stateOrProvinceSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub128">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="streetAddress" type="uds:Ub128" substitutionGroup="uds:Attribute"/>
  <xs:element name="streetAddressSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub128">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="houseId" type="uds:Ub64" substitutionGroup="uds:Attribute"/>
  <xs:element name="houseIdSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub64">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="givenName" type="uds:Ub64" substitutionGroup="uds:Attribute"/>
  <xs:element name="givenNameSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub64">
              <xs:attributeGroup ref="uds:substringValueAttr"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="uds:substringAttr"/>
  </xs:complexType>
</xs:element>
<xs:element name="title" substitutionGroup="uds:Attribute">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="uds:Ub64">
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="titleSub" substitutionGroup="uds:Substrings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="value" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="uds:Ub64">
              <xs:attributeGroup ref="uds:substringValueAttr"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="uds:substringAttr"/>
  </xs:complexType>
</xs:element>
<xs:element name="description" substitutionGroup="uds:Attribute">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="uds:Ub1024">
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="businessCategory" substitutionGroup="uds:Attribute">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="uds:Ub128">
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="businessCategorySub" substitutionGroup="uds:Substrings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="value" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="uds:Ub128">
              <xs:attributeGroup ref="uds:substringValueAttr"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="uds:substringAttr"/>
  </xs:complexType>
</xs:element>
<xs:element name="postalCode" type="uds:Ub40" substitutionGroup="uds:Attribute"/>
<xs:element name="postalCodeSub" substitutionGroup="uds:Substrings">
  <xs:complexType>
```

```
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub40">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="postOfficeBox" type="uds:Ub40" substitutionGroup="uds:Attribute"/>
  <xs:element name="postOfficeBoxSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attributeGroup ref="uds:substringValueAttr"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="dmdName" type="uds:Ub64" substitutionGroup="uds:Attribute"/>
  <xs:element name="dmdNameSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub64">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="localityCode" type="uds:Ub64" substitutionGroup="uds:Attribute"/>
  <xs:element name="localityCodeAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:Ub64">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="localityNDC" type="uds:Ub16NumericString"
substitutionGroup="uds:Attribute"/>
  <xs:element name="localityNDCAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:Ub16NumericString">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="subscriberName" substitutionGroup="uds:Attribute">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:Ub64">
          <xs:attribute ref="xml:lang"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="subscriberNameSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub64">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="SubscrType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="residential"/>
      <xs:enumeration value="organization"/>
      <xs:enumeration value="government"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="SubscrTypes">
    <xs:list itemType="uds:SubscrType"/>
  </xs:simpleType>
  <xs:element name="subscriberType" type="uds:SubscrTypes"
substitutionGroup="uds:Attribute"/>
  <xs:element name="subscriberTypeAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:SubscrType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="secondFamilyName" type="uds:Ub128"
substitutionGroup="uds:Attribute"/>
  <xs:element name="secondFamilyNameSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub128">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="profession" substitutionGroup="uds:Attribute">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:Ub128">
          <xs:attribute ref="xml:lang"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="professionSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub128">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="language" type="xs:language" substitutionGroup="uds:Attribute"/>
  <xs:element name="languageAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:language">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="telephone" type="uds:Ub32" substitutionGroup="uds:Attribute"/>
  <xs:element name="telephoneSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub32">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="orAddress" type="uds:Ub1024" substitutionGroup="uds:Attribute"/>
  <xs:element name="orAddressSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub1024">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="mail" type="uds:Ub256" substitutionGroup="uds:Attribute"/>
  <xs:element name="mailSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub256">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="url" type="uds:Ub1024" substitutionGroup="uds:Attribute"/>
  <xs:element name="urlSub" substitutionGroup="uds:Substrings">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="value" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="uds:Ub1024">
                <xs:attributeGroup ref="uds:substringValueAttr"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="uds:substringAttr"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="ComServiceType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="voice"/>
      <xs:enumeration value="fax"/>
      <xs:enumeration value="textPhone"/>
      <xs:enumeration value="videoPhone"/>
      <xs:enumeration value="publicPhoneBox"/>
      <xs:enumeration value="switchBoard"/>
      <xs:enumeration value="pager"/>
      <xs:enumeration value="data"/>
      <xs:enumeration value="e-mail"/>
      <xs:enumeration value="web"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ComServiceTypes">
    <xs:list itemType="uds:ComServiceType"/>
  </xs:simpleType>
  <xs:element name="commService" type="uds:ComServiceTypes"
substitutionGroup="uds:Attribute"/>
  <xs:element name="commServiceAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:ComServiceType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="NetworkType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="pstn"/>
      <xs:enumeration value="isdn"/>
      <xs:enumeration value="gsm"/>
      <xs:enumeration value="umts"/>
      <xs:enumeration value="internet"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="commNetwork" type="uds:NetworkType"
substitutionGroup="uds:Attribute"/>
  <xs:element name="commNetworkAs">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:NetworkType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="addrValidFrom" type="xs:dateTime" substitutionGroup="uds:Attribute"/>
  <xs:element name="addrValidUntil" type="xs:dateTime"
substitutionGroup="uds:Attribute"/>
  <xs:simpleType name="AddrValidityType">
```

```
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="current"/>
      <xs:enumeration value="old"/>
      <xs:enumeration value="future"/>
      <xs:enumeration value="temporary"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="addrValidity" type="uds:AddrValidityType"
substitutionGroup="uds:Attribute"/>
  <xs:element name="addrValidityAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:AddrValidityType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="AddrCoverageType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="international"/>
      <xs:enumeration value="national"/>
      <xs:enumeration value="stateOrProvince"/>
      <xs:enumeration value="locality"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="addrCoverage" type="uds:AddrCoverageType"
substitutionGroup="uds:Attribute"/>
  <xs:element name="addrCoverageAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:AddrCoverageType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="AddrTariffType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="normal"/>
      <xs:enumeration value="premium"/>
      <xs:enumeration value="toll-free"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="addrTariff" type="uds:AddrTariffType"
substitutionGroup="uds:Attribute"/>
  <xs:element name="addrTariffAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:AddrTariffType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="AddrRestrictionType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="public"/>
      <xs:enumeration value="secret"/>
      <xs:enumeration value="call-screen"/>
      <xs:enumeration value="no-marketing"/>
      <xs:enumeration value="complete-only"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="addrRestriction" type="uds:AddrRestrictionType"
substitutionGroup="uds:Attribute"/>
  <xs:element name="addrRestrictionAs" substitutionGroup="uds:Assertion">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="uds:AddrRestrictionType">
          <xs:attributeGroup ref="uds:assertionAttr"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Notification" abstract="true"/>
  <xs:element name="limitProblem" substitutionGroup="uds:Notification">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="adminLimit"/>
        <xs:enumeration value="permanentRestriction"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="serviceProblem" substitutionGroup="uds:Notification">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="hierSelectForbidden"/>
        <xs:enumeration value="hierSelectNotAvailableForService"/>
        <xs:enumeration value="hierSelectNotSupported"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="searchType" type="uds:SearchType"
substitutionGroup="uds:Notification"/>
  <xs:element name="attributeTypeList" type="uds:AttributeTypes"
substitutionGroup="uds:Notification"/>
  <xs:element name="filterNot" type="uds:Filter" substitutionGroup="uds:Notification"/>
  <xs:element name="filterItem" substitutionGroup="uds:Notification">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="uds:Assertion"/>
        <xs:element ref="uds:Substrings"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="providerName" type="uds:ProviderId"
substitutionGroup="uds:Notification"/>
  <xs:element name="hierarchySelectList" type="uds:HierarchySelections"
        substitutionGroup="uds:Notification"/>
  <xs:element name="searchControlOptionsList" type="uds:SearchOptions"
        substitutionGroup="uds:Notification"/>
  <xs:complexType name="AttributeCombinations">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="attributeType" type="uds:AttributeType"/>
      <xs:element name="or" type="uds:Cor"/>
      <xs:element name="not" type="uds:AttributeCombinations"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="Cor">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="attributeType" type="uds:AttributeType"/>
      <xs:element name="and" type="uds:Cand"/>
      <xs:element name="not" type="uds:AttributeCombinations"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="Cand">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="attributeType" type="uds:AttributeType"/>
      <xs:element name="or" type="uds:Cor"/>
      <xs:element name="not" type="uds:AttributeCombinations"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="attributeCombinations" type="uds:AttributeCombinations"
        substitutionGroup="uds:Notification"/>
  <xs:simpleType name="StringMatchType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="exact"/>
      <xs:enumeration value="deletion"/>
      <xs:enumeration value="restrictedDeletion"/>
      <xs:enumeration value="permutation"/>
      <xs:enumeration value="permutationAndDeletion"/>
      <xs:enumeration value="providerDefined"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="StringMatchTypes">
    <xs:list itemType="uds:StringMatchType"/>
  </xs:simpleType>
  <xs:simpleType name="WordMatchType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="exact"/>
      <xs:enumeration value="truncated"/>
      <xs:enumeration value="phonetic"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="WordMatchTypes">
    <xs:list itemType="uds:WordMatchType"/>
  </xs:simpleType>
  <xs:simpleType name="CharacterMatchType">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="exact"/>
      <xs:enumeration value="caseIgnore"/>
      <xs:enumeration value="mapped"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CharacterMatchTypes">
    <xs:list itemType="uds:CharacterMatchType"/>
  </xs:simpleType>
  <xs:element name="wordRestriction" substitutionGroup="uds:Notification">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="stringMatchTypes" type="uds:StringMatchTypes" minOccurs="0"/>
        <xs:element name="wordMatchTypes" type="uds:WordMatchTypes" minOccurs="0"/>
        <xs:element name="characterMatchTypes" type="uds:CharacterMatchTypes"
minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="attributeType" type="uds:AttributeType" use="optional"/>
      <xs:attribute name="initialMinimum" type="xs:positiveInteger" use="optional"/>
      <xs:attribute name="otherMinimum" type="xs:positiveInteger" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="Option">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="paging"/>
      <xs:enumeration value="weighting"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Options">
    <xs:list itemType="uds:Option"/>
  </xs:simpleType>
  <xs:element name="notSupported" type="uds:Options"
substitutionGroup="uds:Notification"/>
</xs:schema>
```

Annex C

Use of Web Service facilities

C.1 Use of Simple Object Access Protocol (SOAP)

C.1.1 Use of SOAP Header element

The SOAP **`Header`** element for a search request shall hold the following header elements:

– **`requestRef`** – This element allows a unique identifier to be assigned to a search request. The same value shall be returned in the returned result to be associated it with the corresponding request. An implementation shall ensure that it does not issue a request with a **`requestRef`** with the same value as an outstanding request. It is recommended not to re-use **requestRef** values for a period that is long compared to the maximum expected response time.

– **searchType** – The SearchType search control from ITU-T Rec. F.510 is provided by this element. It has the XML **SearchType** simple type as syntax. This is an enumerated syntax with the following values listed in the same sequence as the corresponding SearcheTypes as defined in Tables 8/F.510 to 11/F.510 and in Tables 13/F.510 to 18/F.510:
  - **searchForStateOrProvince**;
  - **searchForLocality**;
  - **subscriberWithinLocality**;
  - **subscriberByEntryIdentifier**;
  - **subscriberWithinStateOrProvince**;
  - **subscriberWithinCountry**;
  - **searchForStreetAddress**;
  - **subscriberByStreetAddress**;
  - **subscriberByCommunicationsAddress**; and
  - **subscriberByBusinessCategory**.

– **userClass** – The UserClass search control from ITU-T Rec. F.510 is provided by this element. It can take the values:
  - **operator** (default); and
  - **publicUser**.

The SOAP **Header** element for a search result shall hold the following header element:

– **requestRef** – This element shall take same value of as the **requestRef** element of the corresponding search request.

C.1.2 Use of SOAP Fault element

The **faultcode** can only take the value **Client** when reporting UDS problem. Use of **VersionMismatch** is outside the scope of this specification.

The following situations shall be reported specifying the **Client** error code in the **faultcode** SOAP element:

– The document in the request was not well formed, in which case:
  - the **faultstring** SOAP element shall have the content **notWellFormed**; and
  - the **detail** SOAP element shall hold the detailed entry **explanation**, which can contain additional information in free text format, e.g., some diagnosis returned from an XML parser.

– The document in the request was well formed but did not comply with the XML schema specified in the request, in which case:
  - the **faultstring** SOAP element shall have the content **invalidDocument**; and
  - the **detail** SOAP element shall hold the detailed entry **explanation**, which can contain additional information in free text format, e.g., some diagnosis returned from an XML parser.

– The document in the request was well formed, but the XML schema specified in the request was unknown to the server, in which case:
  - the **faultstring** SOAP element shall have as content the string **unknownSchema**; and
  - the **detail** SOAP element shall be absent.

– The **requestRef** of the header of the request had the same value as an outstanding request from the same client, in which case:

- the **faultstring** SOAP element shall have as content the string **duplicateRequestRef**; and
- the **detail** SOAP element shall be absent.

Annex D

ASN.1 version of UDAP

This annex provides a specification of the UDS protocol using ASN.1. The XML documents produced by use of this specification with the ASN.1 XML encoding rules (defined in ITU-T Rec. X.693) are identical to those defined by the XSD specification in the body of this Recommendation. For compact binary encodings of this protocol this annex (together with the ASN.1 Packed Encoding Rules) provides the definitive specification of the protocol.

Use of this specification enables ASN.1 tools to be used for UDS implementation and also enables relaying between XML encodings of UDS messages, and compact binary encodings of those messages with no loss of information.

```
Uds DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
  Decimal, DateTime, Language, NMTOKEN, NCName
    FROM XSD /* The XSD module is defined in the ASN.1 standards. */

  Lang
    FROM Xml;

PagedResults ::= SEQUENCE {
  pageSize        INTEGER(1..MAX) OPTIONAL,
  pageNumber      INTEGER(1..MAX) OPTIONAL,
  queryReference  OCTET STRING OPTIONAL}

-- Unified Directory Search Request
SearchRequest ::= SEQUENCE {
  entryLimit     INTEGER(1..MAX) OPTIONAL,
  extendedArea   INTEGER(0..MAX) DEFAULT 0,
  searchOptions  SearchOptions OPTIONAL,
  hierSelect     HierarchySelections DEFAULT {self},
  base           SEQUENCE {entryIdentifier  OCTET STRING OPTIONAL} OPTIONAL,
  filter         Filter OPTIONAL,
  infoSelect
    SEQUENCE {attrTypes    AttributeTypes OPTIONAL,
              family-list  SEQUENCE OF Family} OPTIONAL,
  pagedResults   PagedResults OPTIONAL}

ReqHead ::= SEQUENCE {
  requestRef  INTEGER,
  searchType  SearchType,
  userClass   ENUMERATED {operator, publicUser} DEFAULT operator}

ProviderId ::= SEQUENCE {country  Country,
                         dmdName  DmdName OPTIONAL}

NumericString-1 ::= IA5String(FROM ("0".."9"))(PATTERN "[0-9]")
```

```
Ub16NumericString ::= IA5String(FROM ("0".."9"))(SIZE (0..16))(PATTERN "[0-9]")

Ub32 ::= UTF8String(SIZE (0..32))

Ub40 ::= UTF8String(SIZE (0..40))

Ub64 ::= UTF8String(SIZE (0..64))

Ub128 ::= UTF8String(SIZE (0..128))

Ub256 ::= UTF8String(SIZE (0..256))

Ub512 ::= UTF8String(SIZE (0..512))

Ub1024 ::= UTF8String(SIZE (0..1024))

AttributeType ::= ENUMERATED {
  country, locality, stateOrProvince, streetAddress, houseId, givenName,
  title, description, businessCategory, postalCode, postOfficeBox, dmdName,
  localityCode, localityNDC, subscriberName, subscriberType, secondFamilyName,
  profession, language, telephone, orAddress, mail, url, commService,
  commNetwork, addrValidFrom, addrValidUntil, addrValidity, addrCoverage,
  addrTariff, addrRestriction}

AttributeTypes ::= SEQUENCE OF AttributeType

SearchType ::= ENUMERATED {
  searchForStateOrProvince, searchForLocality, subscriberWithinLocality,
  subscriberByEntryIdentifier, subscriberWithinStateOrProvince,
  subscriberWithinCountry, searchForStreetAddress, subscriberByStreetAddress,
  subscriberByCommunicationsAddress, subscriberByBusinessCategory}

HierarchySelection ::= ENUMERATED {
  self, children, parent, hierarchy, top, subtree, all}

HierarchySelections ::= SEQUENCE OF HierarchySelection

SearchOption ::= ENUMERATED {performExactly, includeAllAreas}

SearchOptions ::= SEQUENCE OF SearchOption

Filter ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {assertion   Assertion,
            substrings  Substrings}

Attribute ::= CHOICE {
  country           Country,
  locality          Locality,
  stateOrProvince   StateOrProvince,
  streetAddress     StreetAddress,
  houseId           HouseId,
  givenName         GivenName,
  title             Title,
  description       Description,
  businessCategory  BusinessCategory,
  postalCode        PostalCode,
  postOfficeBox     PostOfficeBox,
  dmdName           DmdName,
  localityCode      LocalityCode,
  localityNDC       LocalityNDC,
  subscriberName    SubscriberName,
  subscriberType    SubscriberType,
  secondFamilyName  SecondFamilyName,
  profession        Profession,
  language          Language,
  telephone         Telephone,
  orAddress         OrAddress,
  mail              Mail,
  url               Url,
  commService       CommService,
  commNetwork       CommNetwork,
  addrValidFrom     AddrValidFrom,
  addrValidUntil    AddrValidUntil,
  addrValidity      AddrValidity,
  addrCoverage      AddrCoverage,
  addrTariff        AddrTariff,
  addrRestriction   AddrRestriction}

Family ::= CHOICE {commsAddress  CommsAddress}

Assertion ::= CHOICE {
  countryAs          CountryAs,
  localityCodeAs     LocalityCodeAs,
  localityNDCAs      LocalityNDCAs,
  subscriberTypeAs   SubscriberTypeAs,
  languageAs         LanguageAs,
  commServiceAs      CommServiceAs,
  addrValidityAs     AddrValidityAs,
  addrCoverageAs     AddrCoverageAs,
  addrTariffAs       AddrTariffAs,
  addrRestrictionAs  AddrRestrictionAs}

AssertionAttr ::= SEQUENCE {weight  ENUMERATED {low, high} DEFAULT high}

Substrings ::= CHOICE {
  localitySub          LocalitySub,
  stateOrProvinceSub   StateOrProvinceSub,
  streetAddressSub     StreetAddressSub,
  houseIdSub           HouseIdSub,
  givenNameSub         GivenNameSub,
  titleSub             TitleSub,
  businessCategorySub  BusinessCategorySub,
  postalCodeSub        PostalCodeSub,
  postOfficeBoxSub     PostOfficeBoxSub,
  dmdNameSub           DmdNameSub,
  subscriberNameSub    SubscriberNameSub,
  secondFamilyNameSub  SecondFamilyNameSub,
  professionSub        ProfessionSub,
  telephoneSub         TelephoneSub,
  orAddressSub         OrAddressSub,
  mailSub              MailSub,
  urlSub               UrlSub}

SubstringAttr ::= SEQUENCE {
  string
    ENUMERATED {exact, deletion, restrDeletion, permutation,
                permutationAndDeletion, providerDefined} DEFAULT exact,
  weight  ENUMERATED {low, high} DEFAULT high}

SubstringValueAttr ::= SEQUENCE {
  wordMatch       ENUMERATED {exact, truncated, phonetic} DEFAULT exact,
  characterMatch  ENUMERATED {exact, caseIgnore, mapped} DEFAULT caseIgnore}

CommsAddress ::= SEQUENCE {
  attribute-list  SEQUENCE (SIZE (1..MAX)) OF attribute Attribute}

Country ::= NMTOKEN
```

```
CountryAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                        base  NMTOKEN}

Locality ::= Ub128

LocalitySub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list  SEQUENCE (SIZE (1..MAX)) OF value Ub128}

StateOrProvince ::= Ub128

StateOrProvinceSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

StreetAddress ::= Ub128

StreetAddressSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

HouseId ::= Ub64

HouseIdSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

GivenName ::= Ub64

GivenNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

Title ::= SEQUENCE {lang  Lang OPTIONAL,
                    base  Ub64}

TitleSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

Description ::= SEQUENCE {lang  Lang OPTIONAL,
                          base  Ub1024}

BusinessCategory ::= SEQUENCE {lang  Lang OPTIONAL,
                               base  Ub128}

BusinessCategorySub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

PostalCode ::= Ub40

PostalCodeSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub40}}

PostOfficeBox ::= Ub40

PostOfficeBoxSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr}}

DmdName ::= Ub64

DmdNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

LocalityCode ::= Ub64

LocalityCodeAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                             base  Ub64}

LocalityNDC ::= Ub16NumericString

LocalityNDCAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  Ub16NumericString}

SubscriberName ::= SEQUENCE {lang  Lang OPTIONAL,
                             base  Ub64}

SubscriberNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

SubscrType ::= ENUMERATED {residential, organization, government}

SubscrTypes ::= SEQUENCE OF SubscrType

SubscriberType ::= SubscrTypes

SubscriberTypeAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                               base  SubscrType}

SecondFamilyName ::= Ub128

SecondFamilyNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

Profession ::= SEQUENCE {lang  Lang OPTIONAL,
                         base  Ub128}

ProfessionSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

Language ::= XSD.Language

LanguageAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                         base  Language}

Telephone ::= Ub32

TelephoneSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub32}}

OrAddress ::= Ub1024

OrAddressSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub1024}}

Mail ::= Ub256

MailSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub256}}

Url ::= Ub1024

UrlSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value-list
    SEQUENCE (SIZE (1..MAX)) OF value
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub1024}}

ComServiceType ::= ENUMERATED {
  voice, fax, textPhone, videoPhone, publicPhoneBox, switchBoard, pager,
  data, e-mail, web}

ComServiceTypes ::= SEQUENCE OF ComServiceType

CommService ::= ComServiceTypes
```

```
CommServiceAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                            base  ComServiceType}

NetworkType ::= ENUMERATED {pstn, isdn, gsm, umts, internet}

CommNetwork ::= NetworkType

CommNetworkAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                            base  NetworkType}

AddrValidFrom ::= DateTime

AddrValidUntil ::= DateTime

AddrValidityType ::= ENUMERATED {current, old, future, temporary}

AddrValidity ::= AddrValidityType

AddrValidityAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrValidityType}

AddrCoverageType ::= ENUMERATED {
  international, national, stateOrProvince, locality}

AddrCoverage ::= AddrCoverageType

AddrCoverageAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrCoverageType}

AddrTariffType ::= ENUMERATED {normal, premium, toll-free}

AddrTariff ::= AddrTariffType

AddrTariffAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                           base  AddrTariffType}

AddrRestrictionType ::= ENUMERATED {
  public, secret, call-screen, no-marketing, complete-only}

AddrRestriction ::= AddrRestrictionType

AddrRestrictionAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrRestrictionType}

Notification ::= CHOICE {
  limitProblem              LimitProblem,
  serviceProblem            ServiceProblem,
  searchType                SearchType,
  attributeTypeList         AttributeTypeList,
  filterNot                 FilterNot,
  filterItem                FilterItem,
  providerName              ProviderName,
  hierarchySelectList       HierarchySelectList,
  searchControlOptionsList  SearchControlOptionsList,
  attributeCombinations     AttributeCombinations,
  wordRestriction           WordRestriction,
  notSupported              NotSupported}

LimitProblem ::= ENUMERATED {adminLimit, permanentRestriction}
```

```
ServiceProblem ::= ENUMERATED {
  hierSelectForbidden, hierSelectNotAvailableForService, hierSelectNotSupported
}

AttributeTypeList ::= AttributeTypes

FilterNot ::= Filter

FilterItem ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {assertion   Assertion,
            substrings  Substrings}

ProviderName ::= ProviderId

HierarchySelectList ::= HierarchySelections

SearchControlOptionsList ::= SearchOptions

AttributeCombinations ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            or             Cor,
            not            AttributeCombinations}

Cor ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            and            Cand,
            not            AttributeCombinations}

Cand ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            or             Cor,
            not            AttributeCombinations}

StringMatchType ::= ENUMERATED {
  exact, deletion, restrictedDeletion, permutation, permutationAndDeletion,
  providerDefined}

StringMatchTypes ::= SEQUENCE OF StringMatchType

WordMatchType ::= ENUMERATED {exact, truncated, phonetic}

WordMatchTypes ::= SEQUENCE OF WordMatchType

CharacterMatchType ::= ENUMERATED {exact, caseIgnore, mapped}

CharacterMatchTypes ::= SEQUENCE OF CharacterMatchType

WordRestriction ::= SEQUENCE {
  attributeType        AttributeType OPTIONAL,
  initialMinimum       INTEGER(1..MAX) OPTIONAL,
  otherMinimum         INTEGER(1..MAX) OPTIONAL,
  stringMatchTypes     StringMatchTypes OPTIONAL,
  wordMatchTypes       WordMatchTypes OPTIONAL,
  characterMatchTypes  CharacterMatchTypes OPTIONAL}

Option ::= ENUMERATED {paging, weighting}

Options ::= SEQUENCE OF Option

NotSupported ::= Options
```

```
/*
ENCODING-CONTROL XER
    GLOBAL-DEFAULTS MODIFIED-ENCODINGS
    NAMESPACE "http://www.itu.int/itu-t/Rec/f515/xsd"

    ATTRIBUTE ALL IN PagedResults, entryLimit, hierSelect, searchOptions,
         extendedArea IN SearchRequest, SearchRequest.base.entryIdentifier,
         ALL IN AssertionAttr, ALL IN SubstringAttr, ALL IN
         SubstringValueAttr, WordRestriction.otherMinimum,
         WordRestriction.initialMinimum, WordRestriction.attributeType
    BASE64 SearchRequest.base.entryIdentifier
    LIST AttributeTypes, HierarchySelections, SearchOptions, SubscrTypes,
         ComServiceTypes, StringMatchTypes, WordMatchTypes, CharacterMatchTypes,
         Options
    NAME SearchRequest, ReqHead, CommsAddress, Country, CountryAs,
         Locality, LocalitySub, StateOrProvince, StateOrProvinceSub,
         StreetAddress, StreetAddressSub, HouseId, HouseIdSub, GivenName,
         GivenNameSub, Title, TitleSub, Description, BusinessCategory,
         BusinessCategorySub, PostalCode, PostalCodeSub, PostOfficeBox,
         PostOfficeBoxSub, DmdName, DmdNameSub, LocalityCode,
         LocalityCodeAs, LocalityNDC, LocalityNDCAs, SubscriberName,
         SubscriberNameSub, SubscriberType, SubscriberTypeAs,
         SecondFamilyName, SecondFamilyNameSub, Profession, ProfessionSub,
         Language, LanguageAs, Telephone, TelephoneSub, OrAddress,
         OrAddressSub, Mail, MailSub, Url, UrlSub, CommService,
         CommServiceAs, CommNetwork, CommNetworkAs, AddrValidFrom,
         AddrValidUntil, AddrValidity, AddrValidityAs, AddrCoverage,
         AddrCoverageAs, AddrTariff, AddrTariffAs, AddrRestriction,
         AddrRestrictionAs, LimitProblem, ServiceProblem,
         AttributeTypeList, FilterNot, FilterItem, ProviderName,
         HierarchySelectList, SearchControlOptionsList, WordRestriction,
         NotSupported AS UNCAPITALIZED
    NAME NumericString-1 AS "NumericString"
    NAME Attribute.ALL, Family.ALL, Assertion.ALL, Substrings.ALL,
         notification.ALL, country, dmdName IN ProviderId
         AS REFERENCE UNCAPITALIZED
    UNTAGGED SearchRequest.infoSelect.family-list, CountryAs.base,
         value-list IN ALL, base IN ALL, StateOrProvinceSub.value-
         list.value.base,
         StreetAddressSub.value-list.value.base, HouseIdSub.value-
         list.value.base,
         GivenNameSub.value-list.value.base, TitleSub.value-list.value.base,
         BusinessCategorySub.value-list.value.base,
         PostalCodeSub.value-list.value.base,
         PostOfficeBoxSub.value-list.value.base, DmdNameSub.value-
         list.value.base,
         SubscriberNameSub.value-list.value.base,
         SecondFamilyNameSub.value-list.value.base,
         ProfessionSub.value-list.value.base, TelephoneSub.value-
         list.value.base,
         OrAddressSub.value-list.value.base, MailSub.value-list.value.base,
         UrlSub.value-list.value.base
    TEXT AttributeType:localityCode AS CAPITALIZED
*/
END
```

```
Uds2 DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
  Decimal, DateTime, Language, NMTOKEN, NCName
    FROM XSD -- The XSD module is defined in the ASN.1 standards.

  Lang
    FROM Xml;

Entry ::= SEQUENCE {
  entryIdentifier  OCTET STRING OPTIONAL,
  returnedObject
    ENUMERATED {subscriber, stateOrProvince, locality, street, businessCategory}
      DEFAULT subscriber,
  hierarchyLevel   INTEGER(0..MAX) OPTIONAL,
  hierarchyBelow   BOOLEAN DEFAULT FALSE,
  choice-list      SEQUENCE OF CHOICE {attribute  Attribute,
                                       family     Family}}

SearchResult ::= SEQUENCE {
  msgCode         Decimal OPTIONAL,
  queryReference  OCTET STRING OPTIONAL,
  performer       ProviderId OPTIONAL,
  information
    SEQUENCE (SIZE (1..MAX)) OF
      CHOICE {hierarchy
                SEQUENCE {entries      INTEGER(1..MAX) OPTIONAL,
                          firsteEntry  INTEGER(1..MAX) DEFAULT 1,
                          entry-list   SEQUENCE (SIZE (1..MAX)) OF entry Entry},
              entry      SEQUENCE {hierSeq  INTEGER OPTIONAL,
                                   base     Entry}} OPTIONAL,
  entryCount
    SEQUENCE {count      INTEGER(1..MAX) OPTIONAL,
              qualifier  ENUMERATED {exact, bestEstimate} OPTIONAL} OPTIONAL,
  notifications
    SEQUENCE {notification-list
                SEQUENCE (SIZE (1..MAX)) OF notification Notification} OPTIONAL}

ResHead ::= SEQUENCE {requestRef  INTEGER}

Explanation ::= UTF8String

ProviderId ::= SEQUENCE {country  Country,
                         dmdName  DmdName OPTIONAL}

NumericString-1 ::= IA5String(FROM ("0".."9"))(PATTERN "[0-9]")

Ub16NumericString ::= IA5String(FROM ("0".."9"))(SIZE (0..16))(PATTERN "[0-9]")

Ub32 ::= UTF8String(SIZE (0..32))

Ub40 ::= UTF8String(SIZE (0..40))

Ub64 ::= UTF8String(SIZE (0..64))

Ub128 ::= UTF8String(SIZE (0..128))

Ub256 ::= UTF8String(SIZE (0..256))

Ub512 ::= UTF8String(SIZE (0..512))
```

```
Ub1024 ::= UTF8String(SIZE (0..1024))

AttributeType ::= ENUMERATED {
  country, locality, stateOrProvince, streetAddress, houseId, givenName,
  title, description, businessCategory, postalCode, postOfficeBox, dmdName,
  localityCode, localityNDC, subscriberName, subscriberType, secondFamilyName,
  profession, language, telephone, orAddress, mail, url, commService,
  commNetwork, addrValidFrom, addrValidUntil, addrValidity, addrCoverage,
  addrTariff, addrRestriction}

AttributeTypes ::= SEQUENCE OF AttributeType

SearchType ::= ENUMERATED {
  searchForStateOrProvince, searchForLocality, subscriberWithinLocality,
  subscriberByEntryIdentifier, subscriberWithinStateOrProvince,
  subscriberWithinCountry, searchForStreetAddress, subscriberByStreetAddress,
  subscriberByCommunicationsAddress, subscriberByBusinessCategory}

HierarchySelection ::= ENUMERATED {
  self, children, parent, hierarchy, top, subtree, all}

HierarchySelections ::= SEQUENCE OF HierarchySelection

SearchOption ::= ENUMERATED {performExactly, includeAllAreas}

SearchOptions ::= SEQUENCE OF SearchOption

Filter ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {assertion   Assertion,
            substrings  Substrings}

Attribute ::= CHOICE {
  country            Country,
  locality           Locality,
  stateOrProvince    StateOrProvince,
  streetAddress      StreetAddress,
  houseId            HouseId,
  givenName          GivenName,
  title              Title,
  description        Description,
  businessCategory   BusinessCategory,
  postalCode         PostalCode,
  postOfficeBox      PostOfficeBox,
  dmdName            DmdName,
  localityCode       LocalityCode,
  localityNDC        LocalityNDC,
  subscriberName     SubscriberName,
  subscriberType     SubscriberType,
  secondFamilyName   SecondFamilyName,
  profession         Profession,
  language           Language,
  telephone          Telephone,
  orAddress          OrAddress,
  mail               Mail,
  url                Url,
  commService        CommService,
  commNetwork        CommNetwork,
  addrValidFrom      AddrValidFrom,
  addrValidUntil     AddrValidUntil,
  addrValidity       AddrValidity,
  addrCoverage       AddrCoverage,
  addrTariff         AddrTariff,
  addrRestriction    AddrRestriction}
```

```
Family ::= CHOICE {commsAddress  CommsAddress}

Assertion ::= CHOICE {
  countryAs           CountryAs,
  localityCodeAs      LocalityCodeAs,
  localityNDCAs       LocalityNDCAs,
  subscriberTypeAs    SubscriberTypeAs,
  languageAs          LanguageAs,
  commServiceAs       CommServiceAs,
  addrValidityAs      AddrValidityAs,
  addrCoverageAs      AddrCoverageAs,
  addrTariffAs        AddrTariffAs,
  addrRestrictionAs   AddrRestrictionAs}

AssertionAttr ::= SEQUENCE {weight  ENUMERATED {low, high} DEFAULT high}

Substrings ::= CHOICE {
  localitySub           LocalitySub,
  stateOrProvinceSub    StateOrProvinceSub,
  streetAddressSub      StreetAddressSub,
  houseIdSub            HouseIdSub,
  givenNameSub          GivenNameSub,
  titleSub              TitleSub,
  businessCategorySub   BusinessCategorySub,
  postalCodeSub         PostalCodeSub,
  postOfficeBoxSub      PostOfficeBoxSub,
  dmdNameSub            DmdNameSub,
  subscriberNameSub     SubscriberNameSub,
  secondFamilyNameSub   SecondFamilyNameSub,
  professionSub         ProfessionSub,
  telephoneSub          TelephoneSub,
  orAddressSub          OrAddressSub,
  mailSub               MailSub,
  urlSub                UrlSub}

SubstringAttr ::= SEQUENCE {
  string
    ENUMERATED {exact, deletion, restrDeletion, permutation,
                permutationAndDeletion, providerDefined} DEFAULT exact,
  weight  ENUMERATED {low, high} DEFAULT high}

SubstringValueAttr ::= SEQUENCE {
  wordMatch       ENUMERATED {exact, truncated, phonetic} DEFAULT exact,
  characterMatch  ENUMERATED {exact, caseIgnore, mapped} DEFAULT caseIgnore}

CommsAddress ::= SEQUENCE {attribute  SEQUENCE (SIZE (1..MAX)) OF Attribute}

Country ::= NMTOKEN

CountryAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                        base  NMTOKEN}

Locality ::= Ub128

LocalitySub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value  SEQUENCE (SIZE (1..MAX)) OF Ub128}

StateOrProvince ::= Ub128

StateOrProvinceSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
```

```
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

StreetAddress ::= Ub128

StreetAddressSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

HouseId ::= Ub64

HouseIdSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

GivenName ::= Ub64

GivenNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

Title ::= SEQUENCE {lang  Lang OPTIONAL,
                    base  Ub64}

TitleSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

Description ::= SEQUENCE {lang  Lang OPTIONAL,
                          base  Ub1024}

BusinessCategory ::= SEQUENCE {lang  Lang OPTIONAL,
                               base  Ub128}

BusinessCategorySub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

PostalCode ::= Ub40

PostalCodeSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub40}}
```

```
PostOfficeBox ::= Ub40

PostOfficeBoxSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF SEQUENCE {COMPONENTS OF SubstringValueAttr}}

DmdName ::= Ub64

DmdNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

LocalityCode ::= Ub64

LocalityCodeAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                             base  Ub64}

LocalityNDC ::= Ub16NumericString

LocalityNDCAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  Ub16NumericString}

SubscriberName ::= SEQUENCE {lang  Lang OPTIONAL,
                             base  Ub64}

SubscriberNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub64}}

SubscrType ::= ENUMERATED {residential, organization, government}

SubscrTypes ::= SEQUENCE OF SubscrType

SubscriberType ::= SubscrTypes

SubscriberTypeAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                               base  SubscrType}

SecondFamilyName ::= Ub128

SecondFamilyNameSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}

Profession ::= SEQUENCE {lang  Lang OPTIONAL,
                         base  Ub128}

ProfessionSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub128}}
```

```
Language ::= XSD.Language

LanguageAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                         base  Language}

Telephone ::= Ub32

TelephoneSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub32}}

OrAddress ::= Ub1024

OrAddressSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub1024}}

Mail ::= Ub256

MailSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub256}}

Url ::= Ub1024

UrlSub ::= SEQUENCE {
  COMPONENTS OF SubstringAttr,
  value
    SEQUENCE (SIZE (1..MAX)) OF
      SEQUENCE {COMPONENTS OF SubstringValueAttr,
                base  Ub1024}}

ComServiceType ::= ENUMERATED {
  voice, fax, textPhone, videoPhone, publicPhoneBox, switchBoard, pager,
  data, e-mail, web}

ComServiceTypes ::= SEQUENCE OF ComServiceType

CommService ::= ComServiceTypes

CommServiceAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                            base  ComServiceType}

NetworkType ::= ENUMERATED {pstn, isdn, gsm, umts, internet}

CommNetwork ::= NetworkType

CommNetworkAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                            base  NetworkType}

AddrValidFrom ::= DateTime

AddrValidUntil ::= DateTime
```

```
AddrValidityType ::= ENUMERATED {current, old, future, temporary}

AddrValidity ::= AddrValidityType

AddrValidityAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrValidityType}

AddrCoverageType ::= ENUMERATED {
  international, national, stateOrProvince, locality}

AddrCoverage ::= AddrCoverageType

AddrCoverageAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrCoverageType}

AddrTariffType ::= ENUMERATED {normal, premium, toll-free}

AddrTariff ::= AddrTariffType

AddrTariffAs ::= SEQUENCE {COMPONENTS OF AssertionAttr,
                           base  AddrTariffType}

AddrRestrictionType ::= ENUMERATED {
  public, secret, call-screen, no-marketing, complete-only}

AddrRestriction ::= AddrRestrictionType

AddrRestrictionAs ::= SEQUENCE {
  COMPONENTS OF AssertionAttr,
  base  AddrRestrictionType}

Notification ::= CHOICE {
  limitProblem              LimitProblem,
  serviceProblem            ServiceProblem,
  searchType                SearchType,
  attributeTypeList         AttributeTypeList,
  filterNot                 FilterNot,
  filterItem                FilterItem,
  providerName              ProviderName,
  hierarchySelectList       HierarchySelectList,
  searchControlOptionsList  SearchControlOptionsList,
  attributeCombinations     AttributeCombinations,
  wordRestriction           WordRestriction,
  notSupported              NotSupported}

LimitProblem ::= ENUMERATED {adminLimit, permanentRestriction}

ServiceProblem ::= ENUMERATED {
  hierSelectForbidden, hierSelectNotAvailableForService, hierSelectNotSupported
}

AttributeTypeList ::= AttributeTypes

FilterNot ::= Filter

FilterItem ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {assertion   Assertion,
            substrings  Substrings}

ProviderName ::= ProviderId
```

```
HierarchySelectList ::= HierarchySelections

SearchControlOptionsList ::= SearchOptions

AttributeCombinations ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            or             Cor,
            not            AttributeCombinations}

Cor ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            and            Cand,
            not            AttributeCombinations}

Cand ::=
  SEQUENCE (SIZE (1..MAX)) OF
    CHOICE {attributeType  AttributeType,
            or             Cor,
            not            AttributeCombinations}

StringMatchType ::= ENUMERATED {
  exact, deletion, restrictedDeletion, permutation, permutationAndDeletion,
  providerDefined}

StringMatchTypes ::= SEQUENCE OF StringMatchType

WordMatchType ::= ENUMERATED {exact, truncated, phonetic}

WordMatchTypes ::= SEQUENCE OF WordMatchType

CharacterMatchType ::= ENUMERATED {exact, caseIgnore, mapped}

CharacterMatchTypes ::= SEQUENCE OF CharacterMatchType

WordRestriction ::= SEQUENCE {
  attributeType        AttributeType OPTIONAL,
  initialMinimum       INTEGER(1..MAX) OPTIONAL,
  otherMinimum         INTEGER(1..MAX) OPTIONAL,
  stringMatchTypes     StringMatchTypes OPTIONAL,
  wordMatchTypes       WordMatchTypes OPTIONAL,
  characterMatchTypes  CharacterMatchTypes OPTIONAL}

Option ::= ENUMERATED {paging, weighting}

Options ::= SEQUENCE OF Option

NotSupported ::= Options

/*
ENCODING-CONTROL XER
    GLOBAL-DEFAULTS MODIFIED-ENCODINGS
    NAMESPACE "http://www.itu.int/itu-t/Rec/f515/xsd"

    ATTRIBUTE entryIdentifier, hierarchyBelow, hierarchyLevel, returnedObject
         IN Entry, entities, firstEntry IN SearchResult.information.hierarchy,
         SearchResult.information.entry.hierSeq, ALL IN SearchResult.entryCount,
         msgCode, queryReference IN SearchResult, ALL IN AssertionAttr,
         ALL IN SubstringAttr, ALL IN SubstringValueAttr,
         otherMinimum, initialMinimum, attributeType IN WordRestriction
    BASE64 Entry.entryIdentifier
    LIST AttributeTypes, HierarchySelections, SearchOptions, SubscrTypes,
         ComServiceTypes, StringMatchTypes, WordMatchTypes, CharacterMatchTypes,
         Options
    NAME SearchResult, ResHead, Explanation, CommsAddress, Country, CountryAs,
         Locality, LocalitySub, StateOrProvince, StateOrProvinceSub,
         StreetAddress, StreetAddressSub, HouseId, HouseIdSub, GivenName,
         GivenNameSub, Title, TitleSub, Description, BusinessCategory,
         BusinessCategorySub, PostalCode, PostalCodeSub, PostOfficeBox,
         PostOfficeBoxSub, DmdName, DmdNameSub, LocalityCode,
         LocalityCodeAs, LocalityNDC, LocalityNDCAs, SubscriberName,
         SubscriberNameSub, SubscriberType, SubscriberTypeAs,
         SecondFamilyName, SecondFamilyNameSub, Profession, ProfessionSub,
         Language, LanguageAs, Telephone, TelephoneSub, OrAddress,
         OrAddressSub, Mail, MailSub, Url, UrlSub, CommService,
         CommServiceAs, CommNetwork, CommNetworkAs, AddrValidFrom,
         AddrValidUntil, AddrValidity, AddrValidityAs, AddrCoverage,
         AddrCoverageAs, AddrTariff, AddrTariffAs, AddrRestriction,
         AddrRestrictionAs, LimitProblem, ServiceProblem,
         AttributeTypeList, FilterNot, FilterItem, ProviderName,
         HierarchySelectList, SearchControlOptionsList, WordRestriction,
         NotSupported AS UNCAPITALIZED
    NAME NumericString-1 AS "NumericString"
    NAME Attribute.ALL, Family.ALL, Assertion.ALL, Substrings.ALL,
         notification.ALL, country, dmdName IN ProviderId
         AS REFERENCE UNCAPITALIZED
    UNTAGGED Entry.choice-list, SearchResult.information.*.hierarchy.entry-list,
         SearchResult.information.*.entry.base,
         SearchResult.notifications.notification-list, CountryAs.base,
         value-list IN ALL, base IN ALL, StateOrProvinceSub.value-
list.value.base,
         StreetAddressSub.value-list.value.base, HouseIdSub.value-
list.value.base,
         GivenNameSub.value-list.value.base, TitleSub.value-list.value.base,
         BusinessCategorySub.value-list.value.base,
         PostalCodeSub.value-list.value.base,
         PostOfficeBoxSub.value-list.value.base, DmdNameSub.value-
list.value.base,
         SubscriberNameSub.value-list.value.base,
         SecondFamilyNameSub.value-list.value.base,
         ProfessionSub.value-list.value.base, TelephoneSub.value-
list.value.base,
         OrAddressSub.value-list.value.base, MailSub.value-list.value.base,
         UrlSub.value-list.value.base
    TEXT AttributeType:localityCode AS CAPITALIZED
*/
END
```

```
Xml DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
  AnyURI, NCName, Language
     FROM XSD; -- The XSD module is defined in the ASN.1 standards.

Lang ::= Language

Space ::= ENUMERATED {default, preserve}

Base ::= AnyURI
```

```
SpecialAttrs ::= SEQUENCE {
  base   Base OPTIONAL,
  lang   Lang OPTIONAL,
  space  Space OPTIONAL}

/*

ENCODING-CONTROL XER
    ATTRIBUTE ALL IN SpecialAttrs, Space, Lang, Base
    TEXT Space

*/
END
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	**Non-telephone telecommunication services**
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2003