Definition of the inetOrgPerson LDAP Object Class

Status of this Memo

Copyright Notice

Abstract

   While the X.500 standards define many useful attribute types [X520]
   and object classes [X521], they do not define a person object class
   that meets the requirements found in today's Internet and Intranet
   directory service deployments. We define a new object class called
   inetOrgPerson for use in LDAP and X.500 directory services that
   extends the X.521 standard organizationalPerson class to meet these
   needs.

Table of Contents

1. Background and Intended Usage

   The inetOrgPerson object class is a general purpose object class that
   holds attributes about people. The attributes it holds were chosen
   to accommodate information requirements found in typical Internet and
   Intranet directory service deployments. The inetOrgPerson object
   class is designed to be used within directory services based on the
   LDAP [RFC2251] and the X.500 family of protocols, and it should be
   useful in other contexts as well. There is no requirement for
   directory services implementors to use the inetOrgPerson object
   class; it is simply presented as well-documented class that
   implementors can choose to use if they find it useful.

The attribute type and object class definitions in this document are written using the BNF form of AttributeTypeDescription and ObjectClassDescription given in [RFC2252]. In some cases lines have been folded for readability.

Attributes that are referenced but not defined in this document are included in one of the following documents:

The COSINE and Internet X.500 Schema [RFC1274]

Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs) [RFC2079]

A Summary of the X.500(96) User Schema for use with LDAPv3 [RFC2256]

See Appendix A for a summary of the attribute types, associated syntaxes, and matching rules used in this document.

2. New Attribute Types Used in the inetOrgPerson Object Class

2.1. Vehicle license or registration plate.

This multivalued field is used to record the values of the license or registration plate associated with an individual.

```
( 2.16.840.1.113730.3.1.1 NAME 'carLicense'
  DESC 'vehicle license or registration plate'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.2. Department number

Code for department to which a person belongs. This can also be strictly numeric (e.g., 1234) or alphanumeric (e.g., ABC/123).

```
( 2.16.840.1.113730.3.1.2
  NAME 'departmentNumber'
  DESC 'identifies a department within an organization'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.3. Display Name

   When displaying an entry, especially within a one-line summary list,
   it is useful to be able to identify a name to be used. Since other
   attribute types such as 'cn' are multivalued, an additional attribute
   type is needed. Display name is defined for this purpose.

   (2.16.840.1.113730.3.1.241
     NAME 'displayName'
     DESC 'preferred name of a person to be used when displaying entries'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
     SINGLE-VALUE)

2.4. Employee Number

   Numeric or alphanumeric identifier assigned to a person, typically
   based on order of hire or association with an organization. Single
   valued.

     (2.16.840.1.113730.3.1.3
       NAME 'employeeNumber'
       DESC 'numerically identifies an employee within an organization'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
       SINGLE-VALUE)

2.5. Employee Type

   Used to identify the employer to employee relationship. Typical
   values used will be "Contractor", "Employee", "Intern", "Temp",
   "External", and "Unknown" but any value may be used.

     (2.16.840.1.113730.3.1.4
       NAME 'employeeType'
       DESC 'type of employment for a person'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

2.6. JPEG Photograph

   Used to store one or more images of a person using the JPEG File
   Interchange Format [JFIF].

     (0.9.2342.19200300.100.1.60
       NAME 'jpegPhoto'
       DESC 'a JPEG image'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.28)

   Note that the jpegPhoto attribute type was defined for use in the
   Internet X.500 pilots but no referencable definition for it could be
   located.

2.7. Preferred Language

   Used to indicate an individual's preferred written or spoken
   language. This is useful for international correspondence or human-
   computer interaction. Values for this attribute type MUST conform to
   the definition of the Accept-Language header field defined in
   [RFC2068] with one exception: the sequence "Accept-Language" ":"
   should be omitted. This is a single valued attribute type.

     (2.16.840.1.113730.3.1.39
       NAME 'preferredLanguage'
       DESC 'preferred written or spoken language for a person'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
       SINGLE-VALUE)
)

2.8. User S/MIME Certificate

   A PKCS#7 [RFC2315] SignedData, where the content that is signed is
   ignored by consumers of userSMIMECertificate values. It is
   recommended that values have a 'contentType' of data with an absent
   'content' field. Values of this attribute contain a person's entire
   certificate chain and an smimeCapabilities field [RFC2633] that at a
   minimum describes their SMIME algorithm capabilities. Values for
   this attribute are to be stored and requested in binary form, as
   'userSMIMECertificate;binary'. If available, this attribute is
   preferred over the userCertificate attribute for S/MIME applications.

     (2.16.840.1.113730.3.1.40
       NAME 'userSMIMECertificate'
       DESC 'PKCS#7 SignedData used to support S/MIME'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.5)

2.9. User PKCS #12

   PKCS #12 [PKCS12] provides a format for exchange of personal identity
   information. When such information is stored in a directory service,
   the userPKCS12 attribute should be used. This attribute is to be
   stored and requested in binary form, as 'userPKCS12;binary'. The
   attribute values are PFX PDUs stored as binary data.

(2.16.840.1.113730.3.1.216
  NAME 'userPKCS12'
  DESC 'PKCS #12 PFX PDU for exchange of personal identity information'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5)

3. Definition of the inetOrgPerson Object Class

   The inetOrgPerson represents people who are associated with an
   organization in some way. It is a structural class and is derived
   from the organizationalPerson class which is defined in X.521 [X521].

(2.16.840.1.113730.3.2.2
    NAME 'inetOrgPerson'
    SUP organizationalPerson
    STRUCTURAL
    MAY (
        audio $ businessCategory $ carLicense $ departmentNumber $
        displayName $ employeeNumber $ employeeType $ givenName $
        homePhone $ homePostalAddress $ initials $ jpegPhoto $
        labeledURI $ mail $ manager $ mobile $ o $ pager $
        photo $ roomNumber $ secretary $ uid $ userCertificate $
        x500uniqueIdentifier $ preferredLanguage $
        userSMIMECertificate $ userPKCS12
)
)

   For reference, we list the following additional attribute types that
   are part of the inetOrgPerson object class. These attribute types
   are inherited from organizationalPerson (which in turn is derived
   from the person object class):

```
    MUST (
        cn $ objectClass $ sn
    )
    MAY (
        description $ destinationIndicator $ facsimileTelephoneNumber $
        internationaliSDNNumber $ l $ ou $ physicalDeliveryOfficeName $
        postalAddress $ postalCode $ postOfficeBox $
        preferredDeliveryMethod $ registeredAddress $ seeAlso $
        st $ street $ telephoneNumber $ teletexTerminalIdentifier $
        telexNumber $ title $ userPassword $ x121Address
    )
```

4.   Example of an inetOrgPerson Entry

   The following example is expressed using the LDIF notation defined in
   [LDIF].

```
   version: 1
   dn: cn=Barbara Jensen,ou=Product Development,dc=siroe,dc=com
   objectClass: top
   objectClass: person
   objectClass: organizationalPerson
   objectClass: inetOrgPerson
   cn: Barbara Jensen
   cn: Babs Jensen
   displayName: Babs Jensen
   sn: Jensen
   givenName: Barbara
   initials: BJJ
   title: manager, product development
   uid: bjensen
   mail: bjensen@siroe.com
   telephoneNumber: +1 408 555 1862
   facsimileTelephoneNumber: +1 408 555 1992
   mobile: +1 408 555 1941
   roomNumber: 0209
   carLicense: 6ABC246
   o: Siroe
   ou: Product Development
   departmentNumber: 2604
   employeeNumber: 42
   employeeType: full time
   preferredLanguage: fr, en-gb;q=0.8, en;q=0.7
   labeledURI: http://www.siroe.com/users/bjensen My Home Page
```

5. Security Considerations

    Attributes of directory entries are used to provide descriptive
    information about the real-world objects they represent, which can be
    people, organizations or devices. Most countries have privacy laws
    regarding the publication of information about people.

    Transfer of cleartext passwords are strongly discouraged where the
    underlying transport service cannot guarantee confidentiality and may
    result in disclosure of the password to unauthorized parties.

6. Acknowledgments

    The Netscape Directory Server team created the inetOrgPerson object
    class based on experience and customer requirements. Anil Bhavnani
    and John Kristian in particular deserve credit for all of the early
    design work.

    Many members of the Internet community, in particular those in the
    IETF ASID and LDAPEXT groups, also contributed to the design of this
    object class.

7. Bibliography

    [JFIF]   E. Hamilton, "JPEG File Interchange Format (Version 1.02)",
               C-Cube Microsystems, Milpitas, CA, September 1, 1992.

    [LDIF]   G. Good, "The LDAP Data Interchange Format (LDIF) -
               Technical Specification", Work in Progress.

    [PKCS12]   "PKCS #12: Personal Information Exchange Standard", Version
               1.0 Draft, 30 April 1997.

    [RFC1274] Barker, P. and S. Kille, "The COSINE and Internet X.500
               Schema", RFC 1274, November 1991.

    [RFC1847] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security
               Multiparts for MIME: Multipart/Signed and
               Multipart/Encrypted", RFC 1847, October 1995.

    [RFC2068] Fielding, R., Gettys, J., Mogul, J., Frystyk, H. and T.
               Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC
               2068, January 1997.

    [RFC2079] Smith, M., "Definition of an X.500 Attribute Type and an
               Object Class to Hold Uniform Resource Identifiers (URIs)",
               RFC 2079, January 1997.

   [RFC2251] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory
             Access Protocol (v3)", RFC 2251, December 1997.

   [RFC2252] Wahl, M., Coulbeck, A., Howes, T., Kille, S., Yeong, W. and
             C. Robbins, "Lightweight Directory Access Protocol (v3):
             Attribute Syntax Definitions", RFC 2252, December 1997.

   [RFC2256] Wahl, M., "A Summary of the X.500(96) User Schema for use
             with LDAPv3", RFC 2256, December 1997.

   [RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version
             1.5", RFC 2315, March 1998.

   [RFC2633] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC
             2633, June 1999.

   [X520]   ITU-T Rec. X.520, "The Directory: Selected Attribute
             Types", 1996.

   [X521]   ITU-T Rec. X.521, "The Directory: Selected Object Classes",
             1996.

8. Author's Address

   Mark Smith
   Netscape Communications Corp.
   501 E. Middlefield Rd., Mailstop MV068
   Mountain View, CA 94043, USA

   Phone: +1 650 937-3477
   EMail: mcs@netscape.com

9. Appendix A - inetOrgPerson Schema Summary

   This appendix provides definitions of all the attribute types
   included in the inetOrgPerson object class along with their
   associated syntaxes and matching rules.

9.1. Attribute Types

9.1.1. New attribute types that are defined in this document

   (2.16.840.1.113730.3.1.1 NAME 'carLicense'
     DESC 'vehicle license or registration plate'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   (2.16.840.1.113730.3.1.2
     NAME 'departmentNumber'
     DESC 'identifies a department within an organization'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   (2.16.840.1.113730.3.1.241
     NAME 'displayName'
     DESC 'preferred name of a person to be used when displaying entries'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
     SINGLE-VALUE)

   (2.16.840.1.113730.3.1.3
     NAME 'employeeNumber'
     DESC 'numerically identifies an employee within an organization'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
     SINGLE-VALUE)

   (2.16.840.1.113730.3.1.4
     NAME 'employeeType'
     DESC 'type of employment for a person'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

```
   ( 0.9.2342.19200300.100.1.60
     NAME 'jpegPhoto'
     DESC 'a JPEG image'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.28 )
  Note: The jpegPhoto attribute type was defined for use in the
     Internet X.500 pilots but no referencable definition for it
     could be located.

   ( 2.16.840.1.113730.3.1.39
     NAME 'preferredLanguage'
     DESC 'preferred written or spoken language for a person'
     EQUALITY caseIgnoreMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
     SINGLE-VALUE )

   ( 2.16.840.1.113730.3.1.40
     NAME 'userSMIMECertificate'
     DESC 'signed message used to support S/MIME'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 )

   ( 2.16.840.1.113730.3.1.216
     NAME 'userPKCS12'
     DESC 'PKCS #12 PFX PDU for exchange of personal identity information'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 )
```

9.1.2. Attribute types from RFC 2256

   Note that the original definitions of these types can be found in
   X.520.

```
     ( 2.5.4.15
       NAME 'businessCategory'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128} )

     ( 2.5.4.3
       NAME 'cn'
       SUP name )

     ( 2.5.4.13
       NAME 'description'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024} )
```

```
( 2.5.4.27
  NAME 'destinationIndicator'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.44{128} )

( 2.5.4.23
  NAME 'facsimileTelephoneNumber'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.22 )

( 2.5.4.42
  NAME 'givenName'
  SUP name )

( 2.5.4.43
  NAME 'initials'
  SUP name )

( 2.5.4.25
  NAME 'internationaliSDNNumber'
  EQUALITY numericStringMatch
  SUBSTR numericStringSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36{16} )

( 2.5.4.7
  NAME 'l'
  SUP name )

( 2.5.4.0
  NAME 'objectClass'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

( 2.5.4.10
  NAME 'o'
  SUP name )

( 2.5.4.11
  NAME 'ou'
  SUP name )

( 2.5.4.19
  NAME 'physicalDeliveryOfficeName'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128} )
```

```
    ( 2.5.4.18
      NAME 'postOfficeBox'
      EQUALITY caseIgnoreMatch
      SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{40} )

    ( 2.5.4.16
      NAME 'postalAddress'
      EQUALITY caseIgnoreListMatch
      SUBSTR caseIgnoreListSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )

    ( 2.5.4.17
      NAME 'postalCode'
      EQUALITY caseIgnoreMatch
      SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{40} )

    ( 2.5.4.28
      NAME 'preferredDeliveryMethod'
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.14
      SINGLE-VALUE )

    ( 2.5.4.26
      NAME 'registeredAddress'
      SUP postalAddress
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )

    ( 2.5.4.34
      NAME 'seeAlso'
      SUP distinguishedName )

    ( 2.5.4.4
      NAME 'sn'
      SUP name )

    ( 2.5.4.8
      NAME 'st'
      SUP name )

    ( 2.5.4.9
      NAME 'street'
      EQUALITY caseIgnoreMatch
      SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128} )
```

```
( 2.5.4.20
  NAME 'telephoneNumber'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50{32} )

( 2.5.4.22
  NAME 'teletexTerminalIdentifier'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.51 )

( 2.5.4.21
  NAME 'telexNumber'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.52 )

( 2.5.4.12
  NAME 'title'
  SUP name )

( 2.5.4.36
  NAME 'userCertificate'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )

( 2.5.4.35
  NAME 'userPassword'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128} )

( 2.5.4.24
  NAME 'x121Address'
  EQUALITY numericStringMatch
  SUBSTR numericStringSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36{15} )

( 2.5.4.45
  NAME 'x500UniqueIdentifier'
  EQUALITY bitStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.6 )
```

Some attribute types included in inetOrgPerson are derived from the 'name' and 'distinguishedName' attribute supertypes:

```
( 2.5.4.41
  NAME 'name'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

```
( 2.5.4.49
  NAME 'distinguishedName'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

9.1.3. Attribute types from RFC 1274

```
( 0.9.2342.19200300.100.1.55
  NAME 'audio'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{250000} )
```
Note: The syntax used here for the audio attribute type is Octet
   String. RFC 1274 uses a syntax called audio which is not defined
   in RFC 1274.

```
( 0.9.2342.19200300.100.1.20
  NAME 'homePhone'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
```
Note: RFC 1274 uses the longer name 'homeTelephoneNumber'.

```
( 0.9.2342.19200300.100.1.39
  NAME 'homePostalAddress'
  EQUALITY caseIgnoreListMatch
  SUBSTR caseIgnoreListSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )
```

```
( 0.9.2342.19200300.100.1.3
  NAME 'mail'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )
```
Note: RFC 1274 uses the longer name 'rfc822Mailbox' and syntax OID
   of 0.9.2342.19200300.100.3.5. All recent LDAP documents and most
   deployed LDAP implementations refer to this attribute as 'mail'
   and define the IA5 String syntax using using the OID
   1.3.6.1.4.1.1466.115.121.1.26, as is done here.

```
( 0.9.2342.19200300.100.1.10
  NAME 'manager'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

```
      ( 0.9.2342.19200300.100.1.41
        NAME 'mobile'
        EQUALITY telephoneNumberMatch
        SUBSTR telephoneNumberSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
    Note: RFC 1274 uses the longer name 'mobileTelephoneNumber'.

      ( 0.9.2342.19200300.100.1.42
        NAME 'pager'
        EQUALITY telephoneNumberMatch
        SUBSTR telephoneNumberSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
    Note: RFC 1274 uses the longer name 'pagerTelephoneNumber'.

      ( 0.9.2342.19200300.100.1.7
        NAME 'photo' )
    Note: Photo attribute values are encoded in G3 fax format with an
        ASN.1 wrapper. Please refer to RFC 1274 section 9.3.7 for
        detailed syntax information for this attribute.

      ( 0.9.2342.19200300.100.1.6
        NAME 'roomNumber'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )

      ( 0.9.2342.19200300.100.1.21
        NAME 'secretary'
        EQUALITY distinguishedNameMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

      ( 0.9.2342.19200300.100.1.1
        NAME 'uid'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
    Note: RFC 1274 uses the longer name 'userid'.
```

9.1.4. Attribute type from RFC 2079

```
      ( 1.3.6.1.4.1.250.1.57
        NAME 'labeledURI'
        EQUALITY caseExactMatch
        SUBSTR caseExactSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

9.2. Syntaxes

9.2.1. Syntaxes from RFC 2252

     (1.3.6.1.4.1.1466.115.121.1.5 DESC 'Binary')

     (1.3.6.1.4.1.1466.115.121.1.6 DESC 'Bit String')

     (1.3.6.1.4.1.1466.115.121.1.8 DESC 'Certificate')

     (1.3.6.1.4.1.1466.115.121.1.12 DESC 'DN')

     (1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String')

     (1.3.6.1.4.1.1466.115.121.1.22 DESC 'Facsimile Telephone Number')

     (1.3.6.1.4.1.1466.115.121.1.26 DESC 'IA5 String')

     (1.3.6.1.4.1.1466.115.121.1.28 DESC 'JPEG')

     (1.3.6.1.4.1.1466.115.121.1.36 DESC 'Numeric String')

     (1.3.6.1.4.1.1466.115.121.1.38 DESC 'OID')

     (1.3.6.1.4.1.1466.115.121.1.41 DESC 'Postal Address')

     (1.3.6.1.4.1.1466.115.121.1.44 DESC 'Printable String')

     (1.3.6.1.4.1.1466.115.121.1.50 DESC 'Telephone Number')

9.2.2. Syntaxes from RFC 2256

     (1.3.6.1.4.1.1466.115.121.1.14 DESC 'Delivery Method')

     (1.3.6.1.4.1.1466.115.121.1.40 DESC 'Octet String')

     (1.3.6.1.4.1.1466.115.121.1.51 DESC 'Teletex Terminal Identifier')

     (1.3.6.1.4.1.1466.115.121.1.52 DESC 'Telex Number')

9.3. Matching Rules

9.3.1. Matching rules from RFC 2252

   Note that the original definition of many of these matching rules can
   be found in X.520.

```
   ( 2.5.13.16 NAME 'bitStringMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.6 )

   ( 1.3.6.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

   ( 2.5.13.11 NAME 'caseIgnoreListMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )

   ( 2.5.13.2 NAME 'caseIgnoreMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

   ( 2.5.13.1 NAME 'distinguishedNameMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

   ( 2.5.13.8 NAME 'numericStringMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )

   ( 2.5.13.0 NAME 'objectIdentifierMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

   ( 2.5.13.20 NAME 'telephoneNumberMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
```

9.3.2. Matching rule from RFC 2256

   Note that the original definition of this matching rule can be found
   in X.520.

```
   ( 2.5.13.17 NAME 'octetStringMatch'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

9.3.3. Additional matching rules from X.520

   caseExactMatch

```
      ( 2.5.13.5 NAME 'caseExactMatch'
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

   This rule determines whether a presented string exactly matches an
   attribute value of syntax DirectoryString. It is identical to
   caseIgnoreMatch except that case is not ignored. Multiple adjoining
   whitespace characters are treated the same as an individual space,
   and leading and trailing whitespace is ignored.

caseExactSubstringsMatch

     (2.5.13.7 NAME 'caseExactSubstringsMatch'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.58)

  This rules determines whether the initial, any and final substring
  elements in a presented value are present in an attribute value of
  syntax DirectoryString. It is identical to caseIgnoreSubstringsMatch
  except that case is not ignored.

caseIgnoreListSubstringsMatch

     (2.5.13.12 NAME 'caseIgnoreListSubstringsMatch'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.58)

  This rule compares a presented substring with an attribute value
  which is a sequence of DirectoryStrings, but where the case of
  letters is not significant for comparison purposes. A presented
  value matches a stored value if and only if the presented value
  matches the string formed by concatenating the strings of the stored
  value. Matching is done according to the caseIgnoreSubstringsMatch
  rule except that none of the initial, final, or any values of the
  presented value match a substring of the concatenated string which
  spans more than one of the strings of the stored value.

9.3.4. Matching rules not defined in any referenced document

  caseIgnoreIA5SubstringsMatch

     (1.3.6.1.4.1.1466.109.114.3 NAME 'caseIgnoreIA5SubstringsMatch'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.58)

  This rules determines whether the initial, any and final substring
  elements in a presented value are present in an attribute value of
  syntax IA5 String without regard to the case of the letters in the
  strings. It is expected that this matching rule will be added to an
  update of RFC 2252.

10. Full Copyright Statement

Acknowledgement