

Groupe de travail Réseau

S. Legg, Ed.

au

Request for Comments : 4517  
 RFC rendues obsolètes: 2252, 2256  
 RFC mise à jour : 3698  
 Catégorie : Norme

eB2Bcom  
 Juin 2006  
 Traduction Claude Brière de L'Isle  
 avril 2007

## Protocole léger d'accès aux répertoires (LDAP) : Syntaxes et règles de correspondance

### Statut de ce mémo

Le présent document spécifie un protocole Internet normalisé pour la communauté de l'Internet, et appelle à discussion et suggestions pour son amélioration. Prière de se référer à l'édition actuelle des "Normes officielles des protocoles Internet" (STD 1) pour l'état de la normalisation et le statut de ce protocole. La distribution du présent mémo n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (2006).

### Résumé

Chaque attribut mémorisé dans un répertoire du protocole léger d'accès aux répertoires (LDAP), dont les valeurs peuvent être transférées dans le protocole LDAP, a une syntaxe définie qui contraint la structure et le format de ses valeurs. La sémantique de comparaison des valeurs d'une syntaxe ne fait pas partie de la définition de la syntaxe mais est plutôt fournie par des règles de correspondance définies séparément. Les règles de correspondance spécifient un argument, une valeur d'assertion, qui a aussi une syntaxe définie. Le présent document définit un ensemble de base des règles de syntaxe et de correspondance à utiliser pour la définition des attributs pour les répertoires LDAP.

### Table des matières

|        |  |    |
|--------|--|----|
| 1      | Introduction.....  | 2  |
| 2      | Conventions .....  | 3  |
| 3      | Syntaxes.....  | 3  |
| 3.1    | Considérations générales.....                              | 3  |
| 3.2    | Définitions communes.....                                  | 4  |
| 3.3    | Définitions de syntaxes .....                              | 4  |
| 3.3.1  | Description de type d'attribut .....                       | 4  |
| 3.3.2  | Chaîne binaire.....  | 5  |
| 3.3.3  | Booléen.....   | 5  |
| 3.3.4  | Chaîne de pays.....  | 5  |
| 3.3.5  | Méthode de livraison .....                                 | 5  |
| 3.3.6  | Chaîne de répertoire.....                                  | 6  |
| 3.3.7  | Description de la règle de contenu de DIT .....            | 7  |
| 3.3.8  | Description de la règle de structure de DIT .....          | 7  |
| 3.3.9  | DN .....   | 7  |
| 3.3.10 | Guide amélioré.....  | 8  |
| 3.3.11 | Numéro téléphonique de télécopie.....                      | 8  |
| 3.3.12 | Fax .....  | 8  |
| 3.3.13 | Generalized Time.....                                      | 9  |
| 3.3.14 | Guide .....  | 10 |
| 3.3.15 | Chaîne IA5.....  | 10 |
| 3.3.16 | Entier .....   | 10 |
| 3.3.17 | JPEG .....   | 10 |
| 3.3.18 | Description de syntaxe LDAP .....                          | 11 |
| 3.3.19 | Description de règle de correspondance .....               | 11 |
| 3.3.20 | Description d'utilisation de règle de corerspondance ..... | 11 |
| 3.3.21 | Name et Optional UID .....                                 | 11 |

|             |   |    |
|-------------|---|----|
| 3.3.22      | Description de forme de nom.....                        | 12 |
| 3.3.23      | Chaîne numérique.....                                   | 12 |
| 3.3.24      | Description de classe d'objets.....                     | 12 |
| 3.3.25      | Chaîne d'octets.....                                    | 13 |
| 3.3.26      | OID.....  | 13 |
| 3.3.27      | Other Mailbox.....                                      | 13 |
| 3.3.28      | Adresse postale.....                                    | 14 |
| 3.3.29      | Chaîne imprimable.....                                  | 14 |
| 3.3.30      | Assertion de sous-chaîne.....                           | 15 |
| 3.3.31      | Numéro de téléphone.....                                | 15 |
| 3.3.32      | Identifiant de terminal Télétexte.....                  | 15 |
| 3.3.33      | Numéro Télex.....                                       | 16 |
| 3.3.34      | UTC Time.....   | 16 |
| 4           | Règles de correspondance.....                           | 17 |
| 4.1         | Considérations générales.....                           | 17 |
| 4.2         | Définitions des règles de correspondance.....           | 18 |
| 4.2.1       | bitStringMatch.....                                     | 18 |
| 4.2.2       | booleanMatch.....                                       | 19 |
| 4.2.3       | caseExactIA5Match.....                                  | 19 |
| 4.2.4       | caseExactMatch.....                                     | 19 |
| 4.2.5       | caseExactOrderingMatch.....                             | 19 |
| 4.2.6       | caseExactSubstringsMatch.....                           | 20 |
| 4.2.7       | caseIgnoreIA5Match.....                                 | 20 |
| 4.2.8       | caseIgnoreIA5SubstringsMatch.....                       | 20 |
| 4.2.9       | caseIgnoreListMatch.....                                | 21 |
| 4.2.10      | caseIgnoreListSubstringsMatch.....                      | 21 |
| 4.2.11      | caseIgnoreMatch.....                                    | 22 |
| 4.2.12      | caseIgnoreOrderingMatch.....                            | 22 |
| 4.2.13      | caseIgnoreSubstringsMatch.....                          | 22 |
| 4.2.14      | directoryStringFirstComponentMatch.....                 | 23 |
| 4.2.15      | distinguishedNameMatch.....                             | 23 |
| 4.2.16      | generalizedTimeMatch.....                               | 23 |
| 4.2.17      | generalizedTimeOrderingMatch.....                       | 24 |
| 4.2.18      | integerFirstComponentMatch.....                         | 24 |
| 4.2.19      | integerMatch.....                                       | 24 |
| 4.2.20      | integerOrderingMatch.....                               | 24 |
| 4.2.21      | keywordMatch.....                                       | 25 |
| 4.2.22      | numericStringMatch.....                                 | 25 |
| 4.2.23      | numericStringOrderingMatch.....                         | 25 |
| 4.2.24      | numericStringSubstringsMatch.....                       | 25 |
| 4.2.25      | objectIdentifierFirstComponentMatch.....                | 26 |
| 4.2.26      | objectIdentifierMatch.....                              | 26 |
| 4.2.27      | octetStringMatch.....                                   | 27 |
| 4.2.28      | octetStringOrderingMatch.....                           | 27 |
| 4.2.29      | telephoneNumberMatch.....                               | 27 |
| 4.2.30      | telephoneNumberSubstringsMatch.....                     | 27 |
| 4.2.31      | uniqueMemberMatch.....                                  | 28 |
| 4.2.32      | wordMatch.....  | 28 |
| 5           | Considérations sur la sécurité.....                     | 28 |
| 6           | Remerciements.....                                      | 29 |
| 7           | Considérations relatives à l'IANA.....                  | 29 |
| 8           | Références.....   | 30 |
| 8.1         | Références normatives.....                              | 30 |
| 8.2         | Références informatives.....                            | 31 |
| Appendice A | Récapitulation des identifiants d'objet de syntaxe..... | 31 |
| Appendice B | Changements par rapport à la RFC 2252.....              | 32 |

## 1 Introduction

Chaque attribut mémorisé dans un répertoire du protocole léger d'accès aux répertoires (LDAP) [RFC4510], dont les valeurs peuvent être transférées dans le protocole LDAP [RFC4511], a une syntaxe définie (c'est-à-dire, type de données) qui contraint la structure et le format de ses valeurs. La sémantique de comparaison pour les valeurs d'une syntaxe ne fait pas partie de la définition de syntaxe mais est plutôt fournie à travers des règles de correspondance définies séparément. Les règles de correspondance spécifient un argument, une valeur d'assertion, qui a aussi une syntaxe définie. Le présent document définit un ensemble de base de syntaxes et de règles de correspondance à utiliser pour définir les attributs pour les répertoires LDAP.

Il est conseillé au lecteur de se familiariser avec les Modèles des informations de répertoire [RFC4512] avant de lire le reste du présent document. La section 3 donne les définitions pour l'ensemble de base des syntaxes LDAP. La section 4 donne les définitions pour l'ensemble de base des règles de correspondance pour LDAP.

Le présent document fait partie intégrante de la spécification technique LDAP [RFC4510], qui rend obsolète dans sa totalité la spécification technique LDAP précédemment définie, RFC 3377.

Les sections 4, 5, et 7 de la RFC 2252 sont rendues obsolètes par la [RFC4512]. Le reste de la RFC 2252 est rendu obsolète par le présent document. Les sections 6 et 8 de la RFC 2256 sont rendues obsolètes par le présent document. Le reste de la RFC 2256 est rendu obsolète par la [RFC4519] et la [RFC4512]. Toute la RFC 3698 est rendue obsolète par le présent document, sauf le paragraphe 2.11.

Un certain nombre d'éléments de schéma qui étaient inclus dans la révision précédente de la spécification technique LDAP ne figurent plus dans la présente révision de LDAP. Les éléments de schéma d'infrastructure de clé publique sont maintenant spécifiés dans la [RFC4523]. Sauf réintroduction dans de futures spécifications techniques, le reste est à considérer comme ayant un caractère historique.

Les changements par rapport à la RFC 2252 sont décrits à l'Appendice B du présent document.

## 2 Conventions

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, RFC 2119 [RFC2119].

Les définitions de syntaxe sont écrites conformément à la règle ABNF <SyntaxDescription> de la [RFC4234] spécifiée dans la [RFC4512], et les définitions de règle de correspondance sont écrites conformément à la règle ABNF <MatchingRuleDescription> spécifiée dans la [RFC4512], excepté que les définitions de syntaxe et de règles de correspondance fournies dans le présent document sont avec retour à la ligne pour faciliter la lisibilité. Lorsque de telles définitions sont transférées comme valeurs d'attribut dans le protocole LDAP (par exemple, respectivement comme les valeurs des attributs ldapSyntaxes et matchingRules [RFC4512]), ces valeurs ne devraient pas contenir de coupures de ligne.

## 3 Syntaxes

Les définitions de syntaxe contraignent la structure des valeurs d'attribut mémorisées dans un répertoire LDAP, et déterminent la représentation des valeurs d'attribut et d'assertion transférées dans le protocole LDAP.

Les syntaxes qui sont nécessaires pour le fonctionnement du répertoire, ou celle qui sont d'utilisation courante, sont spécifiées dans la présente section. Les serveurs DEVRAIENT reconnaître toutes les syntaxes dont la liste figure dans le présent document, mais ils ne sont pas autrement obligés de les prendre en charge, et PEUVENT reconnaître ou prendre en charge d'autres syntaxes. Cependant, la définition de syntaxes arbitraires supplémentaires est déconseillée car elle nuirait à l'interopérabilité. Les mises en œuvre de client et de serveur n'ont pas normalement la capacité à reconnaître dynamiquement de nouvelles syntaxes.

### 3.1 Considérations générales

La description de chaque syntaxe spécifie comment les valeurs d'attribut ou d'assertion qui se conforment à la syntaxe sont à représenter lorsqu'elles sont transférées dans le protocole LDAP [RFC4511]. Cette représentation est désignée comme codage spécifique de LDAP pour la distinguer des autres méthodes de codage des valeurs d'attribut (par exemple, le codage des Règles de codage de base (BER, *Basic Encoding Rules*) [BER] utilisé par les répertoires X.500 [X.500]).

Le codage spécifique de LDAP d'une syntaxe d'attribut donné produit des valeurs alignées sur l'octet. Dans la plus grande mesure possible, les règles de codage pour les syntaxes LDAP devraient produire des chaînes de caractères qui peuvent être affichées avec peu de, ou sans traduction, ou par les clients qui mettent en œuvre LDAP. Cependant, les clients NE DOIVENT PAS supposer que le codage spécifique de LDAP d'une valeur d'une syntaxe non reconnue est une chaîne de caractères lisibles par l'homme. Il y a quelques cas (par exemple, la syntaxe JPEG) où il n'est pas raisonnable de produire une représentation lisible par l'homme.

Chaque syntaxe LDAP est identifiée de façon unique par un identifiant d'objet [ASN.1] représenté en format décimal séparé par des points (*dotted-decimal*) (les petits noms descriptifs ne sont pas définis par les syntaxes). Ces identifiants d'objet ne sont pas destinés à être affichés aux utilisateurs. Les identifiants d'objet pour les syntaxes définies dans le présent document sont récapitulés à l'Appendice A.

Une limite supérieure du nombre de caractères d'une valeur d'attribut avec une syntaxe fondée sur une chaîne, ou le nombre des octets dans une valeur pour toutes les autres syntaxes, PEUT être suggérée en ajoutant la limite entre des accolades à la suite de l'OBJECT IDENTIFIER de la syntaxe dans une définition de type d'attribut (voir la règle <noidlen> dans la [RFC4512]). Une telle limite n'est pas considérée comme faisant partie de l'identifiant de syntaxe.

Par exemple, "1.3.6.1.4.1.1466.115.121.1.15{64}" dans une définition d'attribut suggère que le serveur de répertoire va permettre qu'une valeur de l'attribut ait jusqu'à 64 caractères de long, bien qu'il puisse permettre de plus longues chaînes de caractères. Noter qu'un seul caractère de la syntaxe Directory String peut être codé sur plus d'un octet, car UTF-8 [RFC3629] est un codage de longueur variable. Donc, une chaîne de 64 caractères peut avoir plus de 64 octets de long.

### 3.2 Définitions communes

Les règles ABNF suivantes sont utilisées dans un certain nombre de définitions de syntaxes au paragraphe 3.3.

```
PrintableCharacter = ALPHA / DIGIT / QUOTE / LPAREN / RPAREN / PLUS / COMMA / HYPHEN / DOT / EQUALS /
                   SLASH / COLON / QUESTION / SPACE
PrintableString   = 1*PrintableCharacter
IA5String         = *(%x00-7F)
SLASH             = %x2F ; barre oblique ("/")
COLON             = %x3A ; deux points (":")
QUESTION         = %x3F ; point d'interrogation ("?")
```

Les règles <ALPHA>, <DIGIT>, <QUOTE>, <LPAREN>, <RPAREN>, <PLUS>, <COMMA>, <HYPHEN>, <DOT>, <EQUALS>, et <SPACE> sont définies dans la [RFC4512].

### 3.3 Définitions de syntaxes

#### 3.3.1 Description de type d'attribut

Une valeur de syntaxe de description de type d'attribut est la définition d'un type d'attribut. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <AttributeTypeDescription> dans la [RFC4512].

Par exemple, la définition suivante du type d'attribut createTimestamp tiré de la [RFC4512] est aussi une valeur de la syntaxe de description de type d'attribut. (Note : Les coupures de ligne ont été ajoutées pour la lisibilité ; elles ne font pas partie de la valeur lors de son transfert dans le protocole.)

```
( 2.5.18.1 NAME 'createTimestamp' EQUALITY generalizedTimeMatch ORDERING generalizedTimeOrderingMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 SINGLE-VALUE NO-USER-MODIFICATION USAGE directoryOperation )
```

La définition LDAP de la syntaxe de description de type d'attribut est :

```
( 1.3.6.1.4.1.1466.115.121.1.3 DESC 'Attribute Type Description' )
```

Cette syntaxe correspond au type ASN.1 de AttributeTypeDescription tiré de [X.501].

### 3.3.2 Chaîne binaire

Une valeur de la syntaxe de Bit String est une séquence de chiffres binaires. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
BitString = SQUOTE *binary-digit SQUOTE "B"
binary-digit = "0" / "1"
```

La règle <SQUOTE> est définie dans la [RFC4512].

Exemple :

```
'0101111101'B
```

La définition LDAP de la syntaxe de Bit String est :

```
( 1.3.6.1.4.1.1466.115.121.1.6 DESC 'Bit String' )
```

Cette syntaxe correspond au type ASN.1 BIT STRING tiré de [ASN.1].

### 3.3.3 Booléen

Une valeur de la syntaxe Boolean est une des valeurs booléennes, vrai ou faux. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
Boolean = "TRUE" / "FALSE"
```

La définition LDAP pour la syntaxe Boolean est :

```
( 1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean' )
```

Cette syntaxe correspond au type ASN.1 BOOLEAN tiré de [ASN.1].

### 3.3.4 Chaîne de pays

Une valeur de la syntaxe de Country String est un des codes à deux caractères tirés de ISO 3166 [ISO3166] pour représenter un pays. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
CountryString = 2(PrintableCharacter)
```

La règle <PrintableCharacter> est définie au paragraphe 3.2.

Exemples :

```
US
AU
```

La définition LDAP pour la syntaxe de Country String est :

```
( 1.3.6.1.4.1.1466.115.121.1.11 DESC 'Country String' )
```

Cette syntaxe correspond au type ASN.1 suivant tiré de [X.520] :

```
PrintableString (SIZE (2)) – codes ISO 3166 seulement
```

### 3.3.5 Méthode de livraison

Une valeur de la syntaxe Delivery Method est une séquence d'éléments qui indiquent, dans l'ordre de préférence, le ou les services par lesquels une entité est désireuse et/ou capable de recevoir des messages. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
DeliveryMethod = pdm *( WSP DOLLAR WSP pdm )
```

pdm = "any" / "mhs" / "physical" / "telex" / "teletex" / "g3fax" / "g4fax" / "ia5" / "videotex" / "telephone"

Les règles <WSP> et <DOLLAR> sont définies dans la [RFC4512].

Exemple :

telephone \$ videotex

La définition LDAP pour la syntaxe de Delivery Method est :

( 1.3.6.1.4.1.1466.115.121.1.14 DESC 'Delivery Method' )

Cette syntaxe correspond au type ASN.1 suivant tiré de [X.520] :

```
SEQUENCE OF INTEGER {
    any-delivery-method      (0),
    mhs-delivery             (1),
    physical-delivery        (2),
    telex-delivery           (3),
    teletex-delivery         (4),
    g3-facsimile-delivery    (5),
    g4-facsimile-delivery    (6),
    ia5-terminal-delivery    (7),
    videotex-delivery        (8),
    telephone-delivery      (9) }
```

### 3.3.6 Chaîne de répertoire

Une valeur de la syntaxe de Directory String est une chaîne d'un ou plusieurs caractères arbitraires tirés de l'ensemble de caractères universel (UCS) [UCS]. Une chaîne de caractères de longueur zéro n'est pas permise. Le codage spécifique de LDAP d'une valeur de cette syntaxe est le codage UTF-8 [RFC3629] de la chaîne de caractères. De tels codages sont conformes à l'ABNF suivant :

DirectoryString = 1\*UTF8

La règle <UTF8> est définie dans la [RFC4512].

Exemple :

Ceci est une valeur de Directory String contenant #!%#@.

Les serveurs et les clients DOIVENT être prêts à recevoir des codets UCS arbitraires, y compris des codets en dehors de la gamme ASCII imprimable et des codets non actuellement alloués à un caractère.

Les définitions de type d'attribut qui utilisent la syntaxe de Directory String ne devraient pas restreindre le format des valeurs de Directory String, par exemple, en exigeant que la chaîne de caractères se conforme à des schémas spécifiques décrits par l'ABNF. Une nouvelle syntaxe devrait être définie dans de tels cas.

La définition LDAP pour la syntaxe de Directory String est :

( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String' )

Cette syntaxe correspond au type ASN.1 paramétré DirectoryString tiré de [X.520].

Le type ASN.1 DirectoryString permet un choix entre les types ASN.1 TeletexString, PrintableString, ou UniversalString tirés de [ASN.1]. Noter cependant, que la solution choisie n'est pas indiquée dans le codage spécifique de LDAP d'une valeur de Directory String.

Les mises en œuvre qui convertissent les valeurs de Directory String du codage spécifique de LDAP en codage BER utilisé par X.500 doivent choisir une solution qui permette les caractères particuliers dans la chaîne, et doivent convertir les caractères du codage UTF-8 en codage de la solution choisie. Lors de la conversion des valeurs de Directory String du codage BER en codage

spécifique de LDAP, les caractères doivent être convertis du codage de caractère de la solution choisie en codage UTF-8. Ces conversions DEVRAIT être effectuées d'une manière cohérente avec l'étape Transcode dans les algorithmes de préparation de chaîne [RFC4518] pour LDAP.

### 3.3.7 Description de la règle de contenu de DIT

Une valeur de la syntaxe de la description de règle de contenu de DIT est la définition d'une règle de contenu d'arbre d'informations de répertoire (DIT, *Directory Information Tree*). Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <DITContentRuleDescription> dans la [RFC4512].

Exemple :

```
( 2.5.6.4 DESC 'content rule for organization' NOT ( x121Address $ telexNumber ) )
```

La définition LDAP pour la syntaxe de la description de règle de contenu de DIT est :

```
( 1.3.6.1.4.1.1466.115.121.1.16 DESC 'DIT Content Rule Description' )
```

Cette syntaxe correspond au type ASN.1 DITContentRuleDescription tiré de [X.501].

### 3.3.8 Description de la règle de structure de DIT

Une valeur de la syntaxe de la description de règle de structure de DIT est la définition d'une règle de structure de DIT. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <DITStructureRuleDescription> dans la [RFC4512].

Exemple :

```
( 2 DESC 'organization structure rule' FORM 2.5.15.3 )
```

La définition LDAP pour la syntaxe de la description de règle de structure de DIT est :

```
( 1.3.6.1.4.1.1466.115.121.1.17 DESC 'DIT Structure Rule Description' )
```

Cette syntaxe correspond au type ASN.1 DITStructureRuleDescription tiré de [X.501].

### 3.3.9 DN

Une valeur de la syntaxe DN est le nom distinctif (DN, *distinguished name*) (prétendu) d'une entrée [RFC4512]. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <distinguishedName> tirée de la représentation de chaîne des noms distinctifs [RFC4514].

Exemples (d'après la [RFC4514]) :

```
UID=jsmith,DC=example,DC=net
OU=Sales+CN=J. Smith,DC=example,DC=net
CN=John Smith\, III,DC=example,DC=net
CN=Before\0dAfter,DC=example,DC=net
1.3.6.1.4.1.1466.0=#04024869,DC=example,DC=com
CN=Lu\C4\8Di\C4\87
```

La définition LDAP pour la syntaxe de DN est :

```
( 1.3.6.1.4.1.1466.115.121.1.12 DESC 'DN' )
```

La syntaxe de DN correspond au type ASN.1 de DistinguishedName tiré de [X.501]. Noter qu'un nom distinctif codé en BER (comme dans X.500) recodé dans le codage spécifique de LDAP n'est pas nécessairement réversible dans le codage DER d'origine car le type de chaîne choisie dans un des composants DirectoryString du nom distinctif n'est pas indiqué dans le codage spécifique de LDAP du nom distinctif (voir au paragraphe 3.3.6).

### 3.3.10 Guide amélioré

Une valeur de la syntaxe de Enhanced Guide (*guide amélioré*) suggère des critères, qui consistent en combinaisons de types d'attribut et d'opérateurs de filtre, à utiliser en construisant des filtres pour rechercher des entrées de classes d'objet particulières. La syntaxe de Enhanced Guide améliore la syntaxe de Guideen permettant de spécifier la profondeur de recherche recommandée.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```

EnhancedGuide      = object-class SHARP WSP criteria WSP SHARP WSP subset
  object-class     = WSP oid WSP
  subset          = "baseobject" / "oneLevel" / "wholeSubtree"
  criteria        = and-term *( BAR and-term )
  and-term        = term *( AMPERSAND term )
  term            = EXCLAIM term / attributetype DOLLAR match-type / LPAREN criteria RPAREN / true / false
  match-type      = "EQ" / "SUBSTR" / "GE" / "LE" / "APPROX"
  true            = "?true"
  false           = "?false"
  BAR             = %x7C ; barre verticale ("|")
  AMPERSAND       = %x26 ; éperluette ("&")
  EXCLAIM         = %x21 ; point d'exclamation ("!")

```

Les règles <SHARP>, <WSP>, <oid>, <LPAREN>, <RPAREN>, <attributetype>, et <DOLLAR> sont définies dans la [RFC4512].

La définition LDAP pour la syntaxe de Enhanced Guide est :

```
( 1.3.6.1.4.1.1466.115.121.1.21 DESC 'Enhanced Guide' )
```

Exemple :

```
person#(sn$EQ)#oneLevel
```

La syntaxe de Enhanced Guide correspond au type ASN.1 de EnhancedGuide tiré de [X.520]. Le type EnhancedGuide fait référence au type ASN.1 Criteria, tiré aussi de [X.520]. La règle <true>, ci-dessus, représente une expression "et" vide dans une valeur de type Criteria. La règle <false>, ci-dessus, représente une expression "ou" vide dans une valeur du type Criteria.

### 3.3.11 Numéro téléphonique de télécopie

Une valeur de la syntaxe Facsimile Telephone Number est un numéro d'abonné d'un appareil de télécopie sur le réseau téléphonique public commuté. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```

fax-number         = telephone-number *( DOLLAR fax-parameter )
telephone-number   = PrintableString
fax-parameter      = "twoDimensional" / "fineResolution" / "unlimitedLength" / "b4Length" / "a3Width" /
                    "b4Width" / "uncompressed"

```

Le <telephone-number> est une chaîne de caractères imprimables qui se conforme au format accepté mondialement pour représenter les numéros de téléphone internationaux [E.123]. La règle <PrintableString> est définie au paragraphe 3.2. La règle <DOLLAR> est définie dans la [RFC4512].

La définition LDAP pour la syntaxe de Facsimile Telephone Number est :

```
( 1.3.6.1.4.1.1466.115.121.1.22 DESC 'Facsimile Telephone Number' )
```

La syntaxe de Facsimile Telephone Number correspond au type ASN.1 de FacsimileTelephoneNumber tiré de [X.520].

### 3.3.12 Fax

Une valeur de la syntaxe Fax est une image qui est produite en utilisant le processus de télécopie de groupe 3 [FAX] pour dupliquer un objet, comme un mémo. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la chaîne d'octets pour une image de télécopie Groupe 3 comme définie dans [FAX].

La définition LDAP pour la syntaxe de Fax est :

( 1.3.6.1.4.1.1466.115.121.1.23 DESC 'Fax' )

Le type ASN.1 correspondant à la syntaxe Fax est définie comme suit, en supposant des EXPLICIT TAGS :

```
Fax ::= CHOICE {
  g3-facsimile [3] G3FacsimileBodyPart
}
```

Le type ASN.1 G3FacsimileBodyPart est défini dans [X.420].

### 3.3.13 Generalized Time

Une valeur de la syntaxe de Generalized Time est une chaîne de caractères représentant la date et l'heure. Le codage spécifique de LDAP d'une valeur de cette syntaxe est une restriction du format défini dans [ISO8601], et est décrit par l'ABNF suivant:

```
GeneralizedTime    = century year month day hour [ minute [ second / leap-second ] ] [ fraction ] g-time-zone

century            = 2(%x30-39) ; "00" à "99"
year               = 2(%x30-39) ; "00" à "99"
month              = ( %x30 %x31-39 ) ; "01" (janvier) à "09" / ( %x31 %x30-32 ) ; "10" à "12"
day                = ( %x30 %x31-39 ) ; "01" à "09" / ( %x31-32 %x30-39 ) ; "10" à "29" / ( %x33 %x30-31 ) ; "30" à "31"
hour               = ( %x30-31 %x30-39 ) / ( %x32 %x30-33 ) ; "00" to "23"
minute             = %x30-35 %x30-39 ; "00" à "59"
second             = ( %x30-35 %x30-39 ) ; "00" à "59"
leap-second        = ( %x36 %x30 ) ; "60"
fraction           = ( DOT / COMMA ) 1*(%x30-39)
g-time-zone        = %x5A ; "Z" / g-differential
g-differential      = ( MINUS / PLUS ) hour [ minute ]
MINUS               = %x2D ; signe moins ("-")
```

Les règles <DOT>, <COMMA>, et <PLUS> sont définies dans la [RFC4512].

L'ABNF ci-dessus permet des chaînes de caractères qui ne représentent pas des dates valides (dans le calendrier grégorien) et/ou des heures valides (par exemple, 31 février 1994). De telles chaînes de caractères DEVRAIENT être considérées comme invalides pour cette syntaxe.

La valeur d'heure représente le temps universel coordonné (équivalent à l'heure moyenne de Greenwich) si la forme "Z" de <g-time-zone> est utilisée ; autrement, la valeur représente une heure locale dans la zone horaire indiquée par <g-differential>. Dans ce dernier cas, le temps universel coordonné peut être calculé en soustrayant la différence avec l'heure locale. La forme "Z" de <g-time-zone> DEVRAIT être utilisée de préférence à <g-differential>.

Si <minute> est omis, <fraction> représente alors une fraction d'heure ; autrement, si <second> et <leap-second> sont omis, <fraction> représente alors une fraction d'une minute ; autrement, <fraction> représente une fraction de seconde.

Exemples :

```
199412161032Z
199412160532-0500
```

Les deux exemples de valeurs représentent la même heure en temps universel coordonné : 10 h 32 le 16 décembre 1994.

La définition LDAP pour la syntaxe de Generalized Time syntax est :

( 1.3.6.1.4.1.1466.115.121.1.24 DESC 'Generalized Time' )

Cette syntaxe correspond au type ASN.1 GeneralizedTime tiré de [ASN.1], avec la contrainte que l'heure locale sans différence à GMT NE DEVRA PAS être utilisée.

### 3.3.14 Guide

Une valeur de la syntaxe Guide suggère des critères, qui consistent en combinaisons de types d'attribut et opérateurs de filtre, à utiliser en construisant des filtres de recherche des entrées de classes d'objet particulières. La syntaxe de Guide est obsolète et ne devrait plus être utilisée pour définir de nouveaux types d'attribut.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

$$\text{Guide} = [ \text{object-class SHARP} ] \text{criteria}$$

Les règles <object-class> et <criteria> sont définies au paragraphe 3.3.10. La règle <SHARP> est définie dans [RFC4512].

La définition LDAP pour la syntaxe de Guide est :

$$( 1.3.6.1.4.1.1466.115.121.1.25 \text{DESC 'Guide'} )$$

La syntaxe Guide correspond au type ASN.1 Guide tiré de [X.520].

### 3.3.15 Chaîne IA5

Une valeur de la syntaxe IA5 String est une chaîne de zéro, un, ou plusieurs caractères de l'alphabet international n° 5 (IA5) [T.50], la version internationale de l'ensemble de caractères ASCII. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la chaîne non convertie de caractères, qui se conforme à la règle <IA5String> au paragraphe 3.2.

La définition LDAP pour la syntaxe de IA5 String est :

$$( 1.3.6.1.4.1.1466.115.121.1.26 \text{DESC 'IA5 String'} )$$

Cette syntaxe correspond au type ASN.1 IA5String d'après [ASN.1].

### 3.3.16 Entier

Une valeur de la syntaxe Integer est un nombre entier de grandeur non limitée. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la représentation de chaîne de caractères de nombre décimaux facultativement signés du nombre (par exemple, le nombre 1321 est représenté par la chaîne de caractères "1321"). Le codage est défini par l'ABNF suivant :

$$\text{Integer} = ( \text{HYPHEN LDIGIT *DIGIT} ) / \text{number}$$

Les règles <HYPHEN>, <LDIGIT>, <DIGIT>, et <number> sont définies dans la [RFC4512].

La définition LDAP pour la syntaxe de Integer est :

$$( 1.3.6.1.4.1.1466.115.121.1.27 \text{DESC 'INTEGER'} )$$

Cette syntaxe correspond au type ASN.1 INTEGER d'après [ASN.1].

### 3.3.17 JPEG

Une valeur de la syntaxe JPEG est une image dans le format d'échange de fichier JPEG (JFIF, *JPEG File Interchange Format*), comme décrit dans [JPEG]. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la séquence d'octets du codage JFIF de l'image.

La définition LDAP pour la syntaxe de JPEG est :

$$( 1.3.6.1.4.1.1466.115.121.1.28 \text{DESC 'JPEG'} )$$

La syntaxe de JPEG correspond au type ASN.1 suivant :

```
JPEG ::= OCTET STRING (CONSTRAINED BY
{ -- les octets contenus sont une image dans le format d'échange de fichier JPEG -- })
```

### 3.3.18 Description de syntaxe LDAP

Une valeur de la syntaxe de LDAP Syntax Description est la description d'une syntaxe LDAP. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <SyntaxDescription> dans la [RFC4512].

La définition LDAP pour la syntaxe de LDAP Syntax Description est :

```
( 1.3.6.1.4.1.1466.115.121.1.54 DESC 'LDAP Syntax Description' )
```

La définition LDAP ci-dessus pour la syntaxe de LDAP Syntax Description est elle-même une valeur légale de la syntaxe de LDAP Syntax Description.

Le type ASN.1 correspondant à la syntaxe de LDAP Syntax Description est définie comme suit, en supposant des EXPLICIT TAGS :

```
LDAPSyntaxDescription ::= SEQUENCE {
    identifieur      OBJECT IDENTIFIER,
    description      DirectoryString { ub-schema } FACULTATIF }
```

Le type ASN.1 paramétré de DirectoryString est défini dans [X.520].

La valeur de sous-schéma (un entier) est définie par la mise en œuvre. Une définition non normative apparaît dans [X.520].

### 3.3.19 Description de règle de correspondance

Une valeur de la syntaxe de Matching Rule Description est la définition d'une règle de correspondance. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <MatchingRuleDescription> dans la [RFC4512].

Exemple :

```
( 2.5.13.2 NAME 'caseIgnoreMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

La définition LDAP pour la syntaxe de Matching Rule Description est :

```
( 1.3.6.1.4.1.1466.115.121.1.30 DESC 'Matching Rule Description' )
```

Cette syntaxe correspond au type ASN.1 MatchingRuleDescription tiré de [X.501].

### 3.3.20 Description d'utilisation de règle de correspondance

Une valeur de la syntaxe de Matching Rule Use Description indique les types d'attribut auxquels une règle de correspondance peut être appliquée dans un filtre de recherche extensibleMatch [RFC4511]. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <MatchingRuleUseDescription> dans la [RFC4512].

Exemple :

```
( 2.5.13.16 APPLIES ( givenName $ surname ) )
```

La définition LDAP pour la syntaxe de Matching Rule Use Description est :

```
( 1.3.6.1.4.1.1466.115.121.1.31 DESC 'Matching Rule Use Description' )
```

Cette syntaxe correspond au type ASN.1 de MatchingRuleUseDescription tiré de [X.501].

### 3.3.21 Name et Optional UID

Une valeur de la syntaxe Name et Optional UID est le nom distinctif [RFC4512] d'une entité facultativement accompagné par un identifiant unique qui sert à différencier l'entité des autres qui ont un nom distinctif identique.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

NameAndOptionalUID = distinguishedName [ SHARP BitString ]

La règle <BitString> est définie au paragraphe 3.3.2. La règle <distinguishedName> est définie dans la [RFC4514]. La règle <SHARP> est définie dans la [RFC4512].

Noter que bien que le caractère '#' puisse survenir dans la représentation de chaîne d'un nom distinctif, aucun échappement supplémentaire de ce caractère n'est effectué lorsqu'un <distinguishedName> est codé dans un <NameAndOptionalUID>.

Exemple :

1.3.6.1.4.1.1466.0=#04024869,O=Test,C=GB#'0101'B

La définition LDAP pour la syntaxe de Name et Optional UID est :

( 1.3.6.1.4.1.1466.115.121.1.34 DESC 'Name And Optional UID' )

Cette syntaxe correspond au type ASN.1 NameAndOptionalUID d'après [X.520].

### 3.3.22 Description de forme de nom

Une valeur de la syntaxe de Name Form Description est la définition d'une forme de nom, qui régule la façon dont les entrées peuvent être nommées. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <NameFormDescription> dans la [RFC4512].

Exemple :

( 2.5.15.3 NAME 'orgNameForm' OC organization DOIT o )

La définition LDAP pour la syntaxe de Name Form Description est :

( 1.3.6.1.4.1.1466.115.121.1.35 DESC 'Name Form Description' )

Cette syntaxe correspond au type ASN.1 de NameFormDescription d'après [X.501].

### 3.3.23 Chaîne numérique

Une valeur de la syntaxe de Numeric String est la séquence d'un ou plusieurs chiffres et espaces. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la chaîne non convertie de caractères, qui se conforme à l'ABNF suivant :

NumericString = 1\*(DIGIT / SPACE)

Les règles <DIGIT> et <SPACE> sont définies dans la [RFC4512].

Exemple :

15 079 672 281

La définition LDAP pour la syntaxe de Numeric String est :

( 1.3.6.1.4.1.1466.115.121.1.36 DESC 'Numeric String' )

Cette syntaxe correspond au type ASN.1 de NumericString ASN.1 d'après [ASN.1].

### 3.3.24 Description de classe d'objets

Une valeur de la syntaxe de Object Class Description est la définition d'une classe d'objets. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <ObjectClassDescription> dans la [RFC4512].

Exemple :

( 2.5.6.2 NAME 'country' SUP top STRUCTURAL MUST c MAY ( searchGuide \$ description ) )

La définition LDAP pour la syntaxe de Object Class Description est :

( 1.3.6.1.4.1.1466.115.121.1.37 DESC 'Object Class Description' )

Cette syntaxe correspond au type ASN.1 de ObjectClassDescription d'après [X.501].

### 3.3.25 Chaîne d'octets

Une valeur de la syntaxe de Object String est la séquence de zéro, un, ou plusieurs octets arbitraires. Le codage spécifique de LDAP d'une valeur de cette syntaxe est la séquence d'octets non convertie, qui se conforme à l'ABNF suivant :

OctetString = \*OCTET

La règle <OCTET> est définie dans la [RFC4512]. Les valeurs de cette syntaxe ne sont généralement pas lisibles par l'homme.

La définition LDAP pour la syntaxe de Octet String est :

( 1.3.6.1.4.1.1466.115.121.1.40 DESC 'Octet String' )

Cette syntaxe correspond au type ASN.1 de OCTET STRING d'après [ASN.1].

### 3.3.26 OID

Une valeur de la syntaxe OID est un identifiant d'objet : une séquence de deux entiers non négatifs ou plus qui identifient de façon unique un objet ou élément de spécification. Beaucoup d'identifiants d'objet utilisés dans LDAP ont aussi des noms enregistrés auprès de IANA [RFC4520].

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par la règle <oid> dans la [RFC4512].

Exemples :

1.2.3.4  
cn

La définition LDAP pour la syntaxe de OID est :

( 1.3.6.1.4.1.1466.115.121.1.38 DESC 'OID' )

Cette syntaxe correspond au type ASN.1 OBJECT IDENTIFIER tiré de [ASN.1].

### 3.3.27 Other Mailbox

Une valeur de la syntaxe Other Mailbox identifie une boîte de messagerie électronique, dans un système de messagerie nommé particulier. Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

OtherMailbox = mailbox-type DOLLAR mailbox  
mailbox-type = PrintableString  
mailbox = IA5String

La règle <mailbox-type> représente le type de système de messagerie dans lequel réside la boîte de messagerie (par exemple, "MCIMail"), et <mailbox> est la boîte de messagerie réelle dans le système de messagerie décrit par <mailbox-type>. Les règles <PrintableString> et <IA5String> sont définies au paragraphe 3.2. La règle <DOLLAR> est définie dans la [RFC4512].

La définition LDAP pour la syntaxe de Other Mailbox est :

( 1.3.6.1.4.1.1466.115.121.1.39 DESC 'Other Mailbox' )

Le type ASN.1 correspondant à la syntaxe Other Mailbox est défini comme suit, en supposant des EXPLICIT TAGS :

OtherMailbox ::= SEQUENCE {

```

mailboxType PrintableString,
mailbox IA5String
}

```

### 3.3.28 Adresse postale

Une valeur de la syntaxe Postal Address est une séquence de chaînes d'un ou plusieurs caractères UCS arbitraires, qui forment une adresse dans un système de messagerie physique.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```

PostalAddress = line *( DOLLAR line )
line          = 1*line-char
line-char     = %x00-23
               / (%x5C "24") ; "$" codé en pourcentage
               / %x25-5B
               / (%x5C "5C") ; "\" codé en pourcentage
               / %x5D-7F
               / UTFMB

```

Chaque chaîne de caractères (c'est-à-dire, <line>) d'une valeur d'adresse postale est codée comme une chaîne UTF-8 [RFC3629], excepté les caractères "\" et "\$", qui, s'il surviennent dans la chaîne, sont esquivés par un caractère "\" suivi par les deux chiffres du code hexadécimal pour le caractère. Les règles <DOLLAR> et <UTFMB> sont définies dans la [RFC4512].

De nombreux serveurs limitent l'adresse postale à un maximum de six lignes de pas plus de trente caractères chacune.

Exemple :

```

1234 Main St.$Anytown, CA 12345$USA
\241,000,000 Sweepstakes$PO Box 1000000$Anytown, CA 12345$USA

```

La définition LDAP pour la syntaxe de Postal Address est :

```
( 1.3.6.1.4.1.1466.115.121.1.41 DESC 'Postal Address' )
```

Cette syntaxe correspond au type ASN.1 de PostalAddress tiré de [X.520] ; c'est-à-dire

```

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF
DirectoryString { ub-postal-string }

```

Les valeurs de ub-postal-line et ub-postal-string (toutes deux entières) sont définies par la mise en œuvre. Des définitions non normatives apparaissent dans [X.520].

### 3.3.29 Chaîne imprimable

Une valeur de la syntaxe Printable String est une chaîne d'un ou plusieurs caractères latins alphabétiques, numériques, et de ponctuation choisis comme spécifié par la règle <PrintableCharacter> au paragraphe 3.2.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est la chaîne de caractères non convertie, qui se conforme à la règle <PrintableString> au paragraphe 3.2.

Exemple :

```
Ceci est une PrintableString.
```

La définition LDAP pour la syntaxe de PrintableString est :

```
( 1.3.6.1.4.1.1466.115.121.1.44 DESC 'Printable String' )
```

Cette syntaxe correspond au type ASN.1 de PrintableString tiré de [ASN.1].

### 3.3.30 Assertion de sous-chaîne

Une valeur de la syntaxe Substring Assertion est une séquence de zéro, une, ou plusieurs sous-chaînes de caractères utilisées comme argument pour la correspondance extensible de sous-chaîne des valeurs d'attribut de chaîne de caractères ; c'est-à-dire, comme matchValue d'une MatchingRuleAssertion [RFC4511]. Chaque sous-chaîne est une chaîne d'un ou plusieurs caractères arbitraires tirés de l'ensemble de caractères universel (UCS) [UCS]. Une sous-chaîne de longueur zéro n'est pas permise.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```

SubstringAssertion = [ initial ] any [ final ]
initial = substring
any = ASTERISK *(substring ASTERISK)
final = substring
ASTERISK = %x2A ; asterisk ("*")
substring = 1*substring-character
substring-character = %x00-29
                    / (%x5C "2A") ; "*" codé en pourcentage
                    / %x2B-5B
                    / (%x5C "5C") ; "\" codé en pourcentage
                    / %x5D-7F
                    / UTFMB

```

Chaque <substring> d'une valeur de Substring Assertion est codée comme une chaîne UTF-8 [RFC3629], excepté que les caractères "\" et "\*", s'ils surviennent dans la sous-chaîne, sont esquivés par un caractère "\" suivi par le code à deux chiffres hexadécimaux pour le caractère.

La syntaxe de Substring Assertion n'est utilisée que comme la syntaxe d'assertion des valeurs dans la correspondance extensible. Elle n'est pas utilisée comme une syntaxe d'attribut, ou dans le SubstringFilter [RFC4511].

La définition LDAP pour la syntaxe de Substring Assertion est :

```
( 1.3.6.1.4.1.1466.115.121.1.58 DESC 'Substring Assertion' )
```

Cette syntaxe correspond au type ASN.1 de SubstringAssertion tiré de [X.520].

### 3.3.31 Numéro de téléphone

Une valeur de la syntaxe de Telephone Number est une chaîne de caractères imprimables qui se conforme au format accepté au niveau mondial pour représenter les numéros de téléphone international [E.123].

Le codage spécifique de LDAP d'une valeur de cette syntaxe est la chaîne de caractères non convertie, qui se conforme à la règle <PrintableString> du paragraphe 3.2.

Exemples :

```

+1 512 315 0280
+1-512-315-0280
+61 3 9896 7830

```

La définition LDAP pour la syntaxe de Telephone Number est :

```
( 1.3.6.1.4.1.1466.115.121.1.50 DESC 'Telephone Number' )
```

La syntaxe de Telephone Number correspond au type ASN.1 suivant tiré de [X.520] :

```
PrintableString (SIZE(1..ub-telephone-number))
```

La valeur de ub-telephone-number (un entier) est définie par la mise en œuvre. Une définition non normative est donnée dans [X.520].

### 3.3.32 Identifiant de terminal Télétex

Une valeur de cette syntaxe spécifie l'identifiant et (facultativement) les paramètres d'un terminal télétex.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
teletex-id = ttx-term *(DOLLAR ttx-param)
ttx-term  = PrintableString      ; identifiant de terminal
ttx-param = ttx-key COLON ttx-value ; paramètre
ttx-key   = "graphic" / "control" / "misc" / "page" / "private"
ttx-value = *ttx-value-octet
ttx-value-octet = %x00-23
               / (%x5C "24") ; "$" codé en pourcentage
               / %x25-5B
               / (%x5C "5C") ; "\" codé en pourcentage
               / %x5D-FF
```

Les règles <PrintableString> et <COLON> sont définies au paragraphe 3.2. La règle <DOLLAR> est définie dans la [RFC4512].

La définition LDAP pour la syntaxe de Teletex Terminal Identifier est :

```
( 1.3.6.1.4.1.1466.115.121.1.51 DESC 'Teletex Terminal Identifier' )
```

Cette syntaxe correspond au type ASN.1 de TeletexTerminalIdentifier tiré de [X.520].

### 3.3.33 Numéro Télex

Une valeur de la syntaxe Telex Number spécifie le numéro de télex, le code de pays, et le code de réponse d'un terminal télex.

Le codage spécifique de LDAP d'une valeur de cette syntaxe est défini par l'ABNF suivant :

```
telex-number = actual-number DOLLAR country-code DOLLAR answerback
actual-number = PrintableString
country-code  = PrintableString
answerback   = PrintableString
```

La règle <PrintableString> est définie au paragraphe 3.2. La règle <DOLLAR> est définie dans la [RFC4512].

La définition LDAP pour la syntaxe de Telex Number est :

```
( 1.3.6.1.4.1.1466.115.121.1.52 DESC 'Telex Number' )
```

Cette syntaxe correspond au type ASN.1 de TelexNumber tiré de [X.520].

### 3.3.34 UTC Time

Une valeur de la syntaxe de UTC Time est une chaîne de caractères représentant une date et une heure à une précision d'une minute ou une seconde. L'année est donnée par un nombre à deux chiffres. Le codage spécifique de LDAP d'une valeur de cette syntaxe suit le format défini dans [ASN.1] pour le type UTCTime et est décrit par l'ABNF suivant :

```
UTCTime      = an mois jour heure minute [ seconde ] [ u-time-zone ]
u-time-zone  = %x5A ; "Z" / u-differential
u-differential = ( MINUS / PLUS ) heure minute
```

Les règles <an>, <mois>, <jour>, <heure>, <minute>, <seconde>, et <MINUS> sont définies au paragraphe 3.3.13. La règle <PLUS> est définie dans la [RFC4512].

L'ABNF ci-dessus permet des chaînes de caractères qui ne représentent pas des dates valides (dans le calendrier grégorien) et/ou des heures valides. De telles chaînes de caractères DEVRAIENT être considérées comme invalides pour cette syntaxe.

La valeur de l'heure représente l'heure universelle coordonnée si la forme "Z" de <u-time-zone> est utilisée ; autrement, la valeur représente une heure locale. Dans ce cas, si <u-differential> est fourni, l'heure universelle coordonnée peut être calculée

en retranchant le différentiel de l'heure locale. La <u-time-zone> DEVRAIT être présente dans les valeurs d'heure, et la forme "Z" de <u-time-zone> DEVRAIT être utilisée de préférence à <u-differential>.

La définition LDAP pour la syntaxe de UTC Time est :

( 1.3.6.1.4.1.1466.115.121.1.53 DESC 'UTC Time' )

Note : Cette syntaxe est déconseillée en faveur de la syntaxe Generalized Time.

La syntaxe UTC Time correspond au type ASN.1 UTCTime tiré de [ASN.1].

## 4 Règles de correspondance

Les règles de correspondance sont utilisées par les mises en œuvre de répertoires pour comparer des valeurs d'attribut à des valeurs d'assertion lorsque sont effectuées les opérations Search et Compare [RFC4511]. Elles sont aussi utilisées lors de la comparaison d'un nom distinctif prétendu [RFC4512] avec le nom d'une entrée. Lors de la modification d'entrées, les règles de correspondance sont utilisées pour identifier les valeurs à supprimer et empêcher qu'un attribut contienne deux valeurs égales.

Les règles de correspondance qui sont exigées pour le fonctionnement d'un répertoire, ou qui sont d'usage courant, sont spécifiées dans cette section.

### 4.1 Considérations générales

Une règle de correspondance est appliquée à des valeurs d'attribut à travers une AttributeValueAssertion ou une MatchingRuleAssertion [RFC4511]. Les conditions selon lesquelles une AttributeValueAssertion ou MatchingRuleAssertion s'évalue comme Undefined (*indéfini*) sont spécifiées dans un autre document [RFC4511]. Si une assertion n'est pas Undefined, le résultat de l'assertion est alors le résultat de l'application de la règle de correspondance choisie. Une règle de correspondance évalue à TRUE (*vrai*), et dans certains cas à indéfini, comme spécifié dans la description de la règle de correspondance ; autrement, elle évalue à FALSE (*faux*).

Chaque assertion contient une valeur d'assertion. La définition de chaque règle de correspondance spécifie la syntaxe de la valeur d'assertion. La syntaxe de la valeur d'assertion est normalement, mais pas nécessairement, la même que la syntaxe des valeurs d'attribut auxquelles la règle de correspondance peut être appliquée. Noter qu'une AssertionValue dans un SubstringFilter [RFC4511] se conforme à la syntaxe d'assertion de la règle de correspondance d'égalité pour le type d'attribut plutôt qu'à la syntaxe d'assertion de la règle de correspondance de sous-chaînes pour le type d'attribut. En principe, le SubstringFilter tout entier est converti en une valeur d'assertion de la règle de correspondance de sous-chaînes avant d'appliquer la règle.

La définition de chaque règle de correspondance indique les syntaxes d'attribut auxquelles la règle peut s'appliquer, en spécifiant les conditions que le type ASN.1 correspondant de la syntaxe de candidat attribut doit satisfaire. Ces conditions sont aussi satisfaites si le type ASN.1 correspondant est un dérivé étiqueté ou soumis à des contraintes du type ASN.1 explicitement mentionné dans la description de la règle (c'est-à-dire que les étiquettes et contraintes ASN.1 sont ignorées lors de la vérification d'applicabilité), ou s'il est une notation de référence de remplacement pour le type explicitement mentionné. Chaque description de règle fait la liste complète, en tant qu'exemples de syntaxes d'attribut applicables, des syntaxes définies dans le présent document auxquelles la règle de correspondance s'applique. Une règle de correspondance peut être applicable à des syntaxes supplémentaires définies dans d'autres documents si ces syntaxes satisfont aux conditions sur les types ASN.1 correspondants.

La description de chaque règle de correspondance indique si la règle convient pour une utilisation comme règle de correspondance d'égalité (EQUALITY), règle de correspondance d'ordonnement (ORDERING), ou règle de correspondance de sous-chaîne (SUBSTR) dans une définition de type d'attribut [RFC4512].

Chaque règle de correspondance est identifiée de façon univoque par un identifiant d'objet. La définition d'une règle de correspondance ne devrait pas être changée ultérieurement. Si un changement est souhaitable, une nouvelle règle de correspondance devrait plutôt être alors définie avec un identifiant d'objet différent.

Les serveurs PEUVENT mettre en œuvre les règles de correspondance wordMatch et keywordMatch, mais ils DEVRAIENT mettre en œuvre les autres règles de correspondance du paragraphe 4.2. Les serveurs PEUVENT mettre en œuvre des règles de correspondance supplémentaires.

Les serveurs qui mettent en œuvre le filtre extensibleMatch DEVRAIENT permettre l'utilisation des règles de correspondance dont la liste figure au paragraphe 4.2 dans le filtre extensibleMatch et DEVRAIENT permettre l'utilisation des règles de correspondance avec tous les d'attributs connus du serveur, lorsque la syntaxe d'assertion de la règle de correspondance est la même que la syntaxe de valeur de l'attribut.

Les serveurs DOIVENT publier, dans l'attribut matchingRules, les définitions des règles de correspondance référencées par les valeurs des attributs attributeTypes et matchingRuleUse dans la même entrée de sous-schéma. D'autres règles de correspondance non référencées PEUVENT être publiées dans l'attribut matchingRules.

Si le serveur prend en charge le filtre extensibleMatch, le serveur PEUT alors utiliser l'attribut matchingRuleUse pour indiquer l'applicabilité (dans un filtre extensibleMatch) des règles de correspondance choisies des types d'attribut désignés.

## 4.2 Définitions des règles de correspondance

Les chaînes de caractères désignées dans les assertions et les valeurs d'attribut sont préparées conformément aux algorithmes de préparation de chaîne [RFC4518] pour LDAP lors de l'évaluation des règles de correspondance suivantes :

numericStringMatch,  
 numericStringSubstringsMatch,  
 caseExactMatch,  
 caseExactOrderingMatch,  
 caseExactSubstringsMatch,  
 caseExactIA5Match,  
 caseIgnoreIA5Match,  
 caseIgnoreIA5SubstringsMatch,  
 caseIgnoreListMatch,  
 caseIgnoreListSubstringsMatch,  
 caseIgnoreMatch,  
 caseIgnoreOrderingMatch,  
 caseIgnoreSubstringsMatch,  
 directoryStringFirstComponentMatch,  
 telephoneNumberMatch,  
 telephoneNumberSubstringsMatch et  
 wordMatch.

Les étapes Transcode, Normalize, Prohibit, et Check bidi sont les mêmes pour chacune des règles de correspondance. Cependant, les étapes Map et Insignificant Character Handling (*traitement des caractères insignifiants*) dépendent de la règle spécifique, comme précisé dans la description de ces règles de correspondance dans les paragraphes suivants.

### 4.2.1 bitStringMatch

La règle bitStringMatch compare une valeur d'assertion de la syntaxe de Bit String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Bit String) dont le type ASN.1 correspondant est BIT STRING.

Si le type ASN.1 correspondant de la syntaxe d'attribut n'a pas une liste de bits désignée [ASN.1] (ce qui est le cas pour la syntaxe Bit String), la règle évalue alors à TRUE si et seulement si la valeur d'attribut a le même nombre de bits que la valeur d'assertion et que les bits correspondent bit par bit.

Si le type ASN.1 correspondant a une liste de bits désignée, bitStringMatch fonctionne alors comme ci-dessus, excepté que les bits zéro de queue dans les valeurs d'attribut et d'assertion sont traités comme absents.

La définition LDAP pour la syntaxe de la règle bitStringMatch est :

( 2.5.13.16 NAME 'bitStringMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.6 )

La règle bitStringMatch est une règle de correspondance d'égalité.

#### 4.2.2 booleanMatch

La règle booleanMatch compare une valeur d'assertion de la syntaxe Boolean à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Boolean) dont le type ASN.1 correspondant est BOOLEAN.

La règle évalue à TRUE si et seulement si la valeur d'attribut et la valeur d'assertion sont toutes deux TRUE ou toutes deux FALSE.

La définition LDAP pour la syntaxe de la règle booleanMatch est :

( 2.5.13.13 NAME 'booleanMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 )

La règle booleanMatch est une règle de correspondance d'égalité.

#### 4.2.3 caseExactIA5Match

La règle caseExactIA5Match compare une valeur d'assertion de la syntaxe IA5 String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe IA5 String) dont le type ASN.1 correspondant est IA5String.

La règle évalue à TRUE si et seulement si la chaîne de caractère de valeur d'attribut préparée et la chaîne de caractère de valeur d'assertion préparée ont le même nombre de caractères et que les caractères correspondants ont la même combinaison binaire.

En préparant la comparaison de la valeur d'attribut et de la valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de caseExactIA5Match est :

( 1.3.6.1.4.1.1466.109.114.1 NAME 'caseExactIA5Match' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

La règle caseExactIA5Match rule est une règle de correspondance d'égalité.

#### 4.2.4 caseExactMatch

La règle caseExactMatch compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou des types de chaîne de remplacement de DirectoryString, tels que PrintableString (les autres alternatives ne correspondent à aucune syntaxe définie dans le présent document).

La règle évalue à TRUE si et seulement si la chaîne préparée de caractères de valeur d'attribut et la chaîne préparée de caractères de valeur d'assertion ont le même nombre de caractères et si les caractères correspondants ont la même combinaison binaire.

En préparant la comparaison de la valeur d'attribut et de la valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de caseExactMatch est :

( 2.5.13.5 NAME 'caseExactMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

La règle caseExactMatch est une règle de correspondance d'égalité.

#### 4.2.5 caseExactOrderingMatch

La règle caseExactOrderingMatch compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou un de ses types de chaîne de remplacement.

La règle évalue à TRUE si et seulement si, dans l'ordre de collationnement des combinaisons binaires, la chaîne préparée de caractères de valeur d'attribut apparaît plus tôt que la chaîne préparée de caractères de valeur d'assertion; c'est-à-dire, la valeur d'attribut est "moins que" la valeur d'assertion.

Pour préparer la comparaison de valeur d'attribut et de valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractères insignifiants.

La définition LDAP pour la syntaxe de la règle caseExactOrderingMatch est :

( 2.5.13.6 NAME 'caseExactOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

La règle caseExactOrderingMatch est une règle de correspondance d'ordonnement.

#### 4.2.6 caseExactSubstringsMatch

La règle caseExactSubstringsMatch compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou un de ses types de chaîne de remplacement.

La règle évalue à TRUE si et seulement si (1) les sous-chaînes préparées de la valeur d'assertion correspondent à des portions disjointes de la chaîne préparée de caractères de valeur d'attribut dans l'ordre des sous-chaînes dans la valeur d'assertion, (2) une sous-chaîne <initial>, si elle est présente, correspond au commencement de la chaîne préparée de caractères de valeur d'attribut, et (3) une sous-chaîne <final>, si elle est présente, correspond à la fin de la chaîne préparée de caractères de valeur d'attribut. Une sous-chaîne préparée correspond à une portion de la chaîne préparée de caractères de valeur d'attribut si les caractères correspondants ont la même séquence binaire.

En préparant la comparaison des sous-chaînes de valeur d'attribut et de valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle caseExactSubstringsMatch est :

( 2.5.13.7 NAME 'caseExactSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle caseExactSubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.7 caseIgnoreIA5Match

La règle caseIgnoreIA5Match compare une valeur d'assertion de la syntaxe IA5 String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe IA5 String) dont le type ASN.1 correspondant est IA5String.

La règle évalue à TRUE si et seulement si la chaîne préparée de caractères de valeur d'attribut et la chaîne préparée de caractères de valeur d'assertion ont le même nombre de caractères et si les caractères correspondants ont la même séquence binaire.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation de Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractères insignifiants.

La définition LDAP pour la syntaxe de la règle caseIgnoreIA5Match est :

( 1.3.6.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

La règle caseIgnoreIA5Match est une règle de correspondance d'égalité.

#### 4.2.8 caseIgnoreIA5SubstringsMatch

La règle caseIgnoreIA5SubstringsMatch compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe IA5 String) dont le type ASN.1 correspondant est IA5String.

La règle évalue à TRUE si et seulement si (1) les sous-chaînes préparées de la valeur d'assertion correspondent à des portions disjointes de la chaîne préparée de caractères de valeur d'attribut dans l'ordre des sous-chaînes dans la valeur d'assertion, (2) une sous-chaîne <initial>, si elle est présente, correspond au commencement de la chaîne préparée de caractères de valeur d'attribut, et (3) une sous-chaîne <final>, si elle est présente, correspond à la fin de la chaîne préparée de caractères de valeur d'attribut. Une sous-chaîne préparée correspond à une portion de la chaîne préparée de caractères de valeur d'attribut si les caractères correspondants ont la même séquence binaire.

En préparant la comparaison des sous-chaînes de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation de Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

( 1.3.6.1.4.1.1466.109.114.3 NAME 'caseIgnoreIA5SubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle caseIgnoreIA5SubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.9 caseIgnoreListMatch

La règle caseIgnoreListMatch compare une valeur d'assertion qui est une séquence de chaînes à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Postal Address) dont le type ASN.1 correspondant est une SEQUENCE OF du type ASN.1 DirectoryString.

La règle évalue à TRUE si et seulement si la valeur d'attribut et la valeur d'assertion ont le même nombre de chaînes et si les chaînes correspondantes (par position) correspondent conformément à la règle de correspondance caseIgnoreMatch.

Dans [X.520], la syntaxe d'assertion pour cette règle de correspondance est définie comme étant :

SEQUENCE OF DirectoryString {ub-match}

C'est-à-dire qu'elle est différente du type correspondant pour la syntaxe Postal Address. Le choix de la syntaxe Postal Address pour la syntaxe d'assertion de caseIgnoreListMatch dans LDAP ne devrait pas être compris comme une limitation à la règle de correspondance à appliquer seulement aux attributs qui ont la syntaxe Postal Address.

La définition LDAP pour la syntaxe de caseIgnoreListMatch est :

( 2.5.13.11 NAME 'caseIgnoreListMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )

La règle caseIgnoreListMatch est une règle de correspondance d'égalité.

#### 4.2.10 caseIgnoreListSubstringsMatch

La règle caseIgnoreListSubstringsMatch compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Postal Address) dont le type ASN.1 correspondant est une SEQUENCE OF du type ASN.1 DirectoryString.

La règle évalue à TRUE si et seulement si la valeur d'assertion correspond, selon la règle caseIgnoreSubstringsMatch, à la chaîne de caractères formée par l'enchaînement des chaînes de la valeur d'attribut, excepté que aucune sous-chaîne <initial>, <any>, ou <final> de la valeur d'assertion n'est prise en considération pour la confrontation à une sous-chaîne de la chaîne concaténée qui s'étend sur plus d'une des chaînes d'origine de la valeur d'attribut.

Noter que, en termes de codage spécifique de LDAP de la syntaxe Postal Address, la chaîne concaténée omet le séparateur de ligne <DOLLAR> et l'échappement (*codage en pourcentage*) des caractères "\" et "\$".

La définition LDAP pour la syntaxe de la règle caseIgnoreListSubstringsMatch est :

( 2.5.13.12 NAME 'caseIgnoreListSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle caseIgnoreListSubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.11 caseIgnoreMatch

La règle caseIgnoreMatch compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou un de ses types de chaîne de remplacement.

La règle évalue à TRUE si et seulement si la chaîne préparée de caractères de valeur d'attribut et la chaîne préparée de caractères de valeur d'assertion ont le même nombre de caractères et si les caractères correspondants ont la même séquence binaire.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle caseIgnoreMatch est :

( 2.5.13.2 NAME 'caseIgnoreMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

La règle caseIgnoreMatch est une règle de correspondance d'égalité.

#### 4.2.12 caseIgnoreOrderingMatch

La règle caseIgnoreOrderingMatch compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou un de ses types de chaîne de remplacement.

La règle évalue à TRUE si et seulement si, dans l'ordre de collationnement de la séquence binaire, la chaîne préparée de caractères de valeur d'attribut apparaît plus tôt que la chaîne préparée de caractères de valeur d'assertion; c'est-à-dire, la valeur d'attribut est "inférieure" à la valeur d'assertion.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle caseIgnoreOrderingMatch est :

( 2.5.13.3 NAME 'caseIgnoreOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

La règle caseIgnoreOrderingMatch est une règle de correspondance d'ordonnement.

#### 4.2.13 caseIgnoreSubstringsMatch

La règle caseIgnoreSubstringsMatch compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String, Printable String, Country String, ou Telephone Number) dont le type ASN.1 correspondant est DirectoryString ou un de ses types de chaîne de remplacement.

La règle évalue à TRUE si et seulement si (1) les sous-chaînes préparées de la valeur d'assertion correspondent à des portions disjointes de la chaîne préparée de caractères de valeur d'attribut dans l'ordre des sous-chaînes dans la valeur d'assertion, (2) une sous-chaîne <initial>, si elle est présente, correspond au commencement de la chaîne préparée de caractères de valeur d'attribut, et (3) une sous-chaîne <final>, si elle est présente, correspond à la fin de la chaîne préparée de caractères de valeur d'attribut. Une sous-chaîne préparée correspond à une portion de la chaîne préparée de caractères de valeur d'attribut si les caractères correspondants ont la même séquence binaire.

En préparant la comparaison des sous-chaînes de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation Map, et seul le traitement d'espace insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle caseIgnoreSubstringsMatch est :

( 2.5.13.4 NAME 'caseIgnoreSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle caseIgnoreSubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.14 **directoryStringFirstComponentMatch**

La règle `directoryStringFirstComponentMatch` compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe dont le type ASN.1 correspondant est une SEQUENCE avec un premier composant obligatoire du type ASN.1 DirectoryString.

Noter que la syntaxe d'assertion de cette règle de correspondance diffère de la syntaxe d'attribut des attributs pour lesquels c'est la règle de correspondance d'égalité.

La règle évalue à TRUE si et seulement si la valeur d'assertion correspond au premier composant de la valeur d'attribut qui utilise les règles de `caseIgnoreMatch`.

La définition LDAP pour la syntaxe de la règle de correspondance `directoryStringFirstComponentMatch` est :

( 2.5.13.31 NAME 'directoryStringFirstComponentMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

La règle `directoryStringFirstComponentMatch` est une règle de correspondance d'égalité. Lorsqu'on utilise `directoryStringFirstComponentMatch` pour comparer deux des valeurs d'attribut (d'une syntaxe applicable), une valeur d'assertion doit d'abord être dérivée d'une des valeurs d'attribut. Une valeur d'assertion peut être dérivée d'une valeur d'attribut en prenant le premier composant de cette valeur d'attribut.

#### 4.2.15 **distinguishedNameMatch**

La règle `distinguishedNameMatch` compare une valeur d'assertion de la syntaxe DN à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe DN) dont le type ASN.1 correspondant est DistinguishedName.

La règle évalue à TRUE si et seulement si la valeur d'attribut et la valeur d'assertion ont le même nombre de noms distinctifs relatifs et si les noms distinctifs relatifs correspondants (par position) sont les mêmes. Un nom distinctif relatif (RDN, *relative distinguished name*) de la valeur d'assertion est le même qu'un RDN de la valeur d'attribut si et seulement si ils ont le même nombre d'assertions de valeur d'attribut et si chaque assertion de valeur d'attribut (AVA, *attribute value assertion*) du premier RDN est le même que l'AVA du second RDN avec le même type d'attribut. L'ordre des AVA n'est pas significatif. Noter aussi qu'un type d'attribut particulier peut apparaître dans au plus une AVA dans un RDN. Deux AVA avec le même type d'attribut sont les mêmes si leurs valeurs sont égales selon la règle de correspondance d'égalité du type d'attribut. Si une ou plusieurs des comparaisons d'AVA sont évaluées comme Indéfini et que les comparaisons d'AVA restantes retournent TRUE, la règle `distinguishedNameMatch` évalue à Indéfini.

La définition LDAP pour la syntaxe de la règle `distinguishedNameMatch` est :

( 2.5.13.1 NAME 'distinguishedNameMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

La règle `distinguishedNameMatch` est une règle de correspondance d'égalité.

#### 4.2.16 **generalizedTimeMatch**

La règle `generalizedTimeMatch` compare une valeur d'assertion de la syntaxe Generalized Time à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Generalized Time) dont le type ASN.1 correspondant est GeneralizedTime.

La règle évalue à TRUE si et seulement si la valeur d'attribut représente la même heure coordonnée universelle que la valeur d'assertion. Si une heure est spécifiée sans les minutes ou les secondes, le nombre de minutes ou (respectivement) de secondes est supposé être de zéro.

La définition LDAP pour la syntaxe de la règle `generalizedTimeMatch` est :

( 2.5.13.27 NAME 'generalizedTimeMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

La règle `generalizedTimeMatch` est une règle de correspondance d'égalité.

#### 4.2.17 **generalizedTimeOrderingMatch**

La règle `generalizedTimeOrderingMatch` compare l'ordre horaire d'une valeur d'assertion de la syntaxe `Generalized Time` à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe `Generalized Time`) dont le type ASN.1 correspondant est `GeneralizedTime`.

La règle évalue à `TRUE` si et seulement si la valeur d'attribut représente une heure coordonnée universelle qui est plus tôt que l'heure coordonnée universelle représentée par la valeur d'assertion.

La définition LDAP pour la syntaxe de la règle `generalizedTimeOrderingMatch` est :

( 2.5.13.28 NAME 'generalizedTimeOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

La règle `generalizedTimeOrderingMatch` est une règle de correspondance d'ordonnement.

#### 4.2.18 **integerFirstComponentMatch**

La règle `integerFirstComponentMatch` compare une valeur d'assertion de la syntaxe `Integer` à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe `DIT de description de règle de structure`) dont le type ASN.1 correspondant est une `SEQUENCE` avec un premier composant obligatoire du type ASN.1 `INTEGER`.

Noter que la syntaxe d'assertion de cette règle de correspondance diffère de la syntaxe d'attribut des attributs pour lesquels c'est la règle de correspondance d'égalité.

La règle évalue à `TRUE` si et seulement si la valeur d'assertion et le premier composant de la valeur d'attribut ont la même valeur d'entier.

La définition LDAP pour la syntaxe de la règle de correspondance `integerFirstComponentMatch` est :

( 2.5.13.29 NAME 'integerFirstComponentMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

La règle `integerFirstComponentMatch` est une règle de correspondance d'égalité. Lorsqu'on utilise `integerFirstComponentMatch` pour comparer deux des valeurs d'attribut (sur une syntaxe applicable), une valeur d'assertion doit d'abord être déduite d'une des valeurs d'attribut. Une valeur d'assertion peut être déduite d'une valeur d'attribut en prenant le premier composant de cette valeur d'attribut.

#### 4.2.1 **integerMatch**

La règle `integerMatch` compare une valeur d'assertion de la syntaxe `Integer` à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe `Integer`) dont le type ASN.1 correspondant est `INTEGER`.

La règle évalue à `TRUE` si et seulement si la valeur d'attribut et la valeur d'assertion ont la même valeur entière.

La définition LDAP pour la syntaxe de la règle de correspondance `integerMatch` est :

( 2.5.13.14 NAME 'integerMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

La règle `integerMatch` est une règle de correspondance d'égalité.

#### 4.2.20 **integerOrderingMatch**

La règle `integerOrderingMatch` compare une valeur d'assertion de la syntaxe `Integer` à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe `Integer`) dont le type ASN.1 correspondant est `INTEGER`.

La règle évalue à `TRUE` si et seulement si la valeur entière de la valeur d'attribut est inférieure à la valeur entière de la valeur d'assertion.

La définition LDAP pour la syntaxe de la règle de correspondance `integerOrderingMatch` est :

( 2.5.13.15 NAME 'integerOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

La règle `integerOrderingMatch` est une règle de correspondance d'ordonnement.

#### 4.2.21 `keywordMatch`

La règle `keywordMatch` compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String) dont le type ASN.1 correspondant est DirectoryString.

La règle évalue à TRUE si et seulement si la chaîne de caractères de valeur d'assertion correspond à chaque mot clé dans la valeur d'attribut. L'identification des mots clés dans la valeur d'attribut et l'exactitude de la correspondance sont toutes deux spécifiques de la mise en œuvre.

La définition LDAP pour la syntaxe de la règle `keywordMatch` est :

( 2.5.13.33 NAME 'keywordMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

#### 4.2.22 `numericStringMatch`

La règle `numericStringMatch` compare une valeur d'assertion de la syntaxe Numeric String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Numeric String) dont le type ASN.1 correspondant est NumericString.

La règle évalue à TRUE si et seulement si la chaîne préparée de caractères de valeur d'attribut et la chaîne préparée de caractères de valeur d'assertion ont le même nombre de caractères et si les caractères correspondants ont la même séquence binaire. En préparant la comparaison de la valeur d'attribut et de la valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation de Map, et seul le traitement `numericString` de caractère insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle de correspondance `numericStringMatch` est :

( 2.5.13.8 NAME 'numericStringMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )

La règle `numericStringMatch` est une règle de correspondance d'égalité.

#### 4.2.23 `numericStringOrderingMatch`

La règle `numericStringOrderingMatch` compare une valeur d'assertion de la syntaxe Numeric String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Numeric String) dont le type ASN.1 correspondant est NumericString.

La règle évalue à TRUE si et seulement si, dans l'ordre de collationnement de la combinaison binaire, la chaîne préparée de caractères de valeur d'attribut apparaît plus tôt que la chaîne préparée de caractères de valeur d'assertion ; c'est-à-dire, la valeur d'attribut est "inférieure" à la valeur d'assertion.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation de Map, et seul le traitement `numericString` de caractère insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La règle est identique à la règle `caseIgnoreOrderingMatch` excepté que tous les caractères espace sont sautés durant la comparaison (la casse n'est pas pertinente car les caractères sont numériques).

La définition LDAP pour la syntaxe de la règle de correspondance `numericStringOrderingMatch` est :

( 2.5.13.9 NAME 'numericStringOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )

La règle `numericStringOrderingMatch` est une règle de correspondance d'ordonnement.

#### 4.2.24 `numericStringSubstringsMatch`

La règle `numericStringSubstringsMatch` compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Numeric String) dont le type ASN.1 correspondant est NumericString.

La règle évalue à TRUE si et seulement si (1) les sous-chaînes préparées de la valeur d'assertion correspondent à des portions disjointes de la chaîne préparée de caractères de valeur d'attribut dans l'ordre des sous chaînes dans la valeur d'assertion, (2) une sous-chaîne <initial>, si elle est présente, correspond au commencement de la chaîne préparée de caractères de valeur d'attribut, et (3) une sous-chaîne <final>, si elle est présente, correspond à la fin de la chaîne préparée de caractères de valeur d'attribut. Une sous-chaîne préparée correspond à une portion de la chaîne préparée de caractères de valeur d'attribut si les caractères correspondants ont la même combinaison binaire.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères ne sont pas changés de casse dans l'étape de préparation de Map, et seul le traitement numericString de caractère insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle de correspondance numericStringSubstringsMatch est :

( 2.5.13.10 NAME 'numericStringSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle numericStringSubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.25 objectIdentifierFirstComponentMatch

La règle objectIdentifierFirstComponentMatch compare une valeur d'assertion de la syntaxe OID à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe de description de type d'attribut, de description DIT de règle de contenu, de description de syntaxe LDAP, de description de règle de correspondance, de description d'utilisation de règle de correspondance, de description de forme de nom, ou de description de classe d'objet) dont le type ASN.1 correspondant est une SEQUENCE avec un premier composant obligatoire du type ASN.1 OBJECT IDENTIFIER.

Noter que la syntaxe d'assertion de cette règle de correspondance diffère de la syntaxe d'attribut des attributs pour lesquels c'est la règle de correspondance d'égalité.

La règle évalue à TRUE si et seulement si la valeur d'assertion correspond au premier composant de la valeur d'attribut en utilisant les règles de objectIdentifierMatch.

La définition LDAP pour la syntaxe de la règle de correspondance objectIdentifierFirstComponentMatch est :

( 2.5.13.30 NAME 'objectIdentifierFirstComponentMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

La règle objectIdentifierFirstComponentMatch est une règle de correspondance d'égalité. Lorsqu'on utilise objectIdentifierFirstComponentMatch pour comparer deux des valeurs d'attribut (d'une syntaxe applicable), une valeur d'assertion doit d'abord être déduite d'une des valeurs d'attribut. Une valeur d'assertion peut être déduite d'une valeur d'attribut en prenant le premier composant de cette valeur d'attribut.

#### 4.2.26 objectIdentifierMatch

La règle objectIdentifierMatch compare une valeur d'assertion de la syntaxe OID à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe OID) dont le type ASN.1 correspondant est OBJECT IDENTIFIER.

La règle évalue à TRUE si et seulement si la valeur d'assertion et la valeur d'attribut représentent le même identifiant d'objet; c'est-à-dire, la même séquence d'entiers, qu'ils soient représentés explicitement sous la forme <numericoid> de <oid> ou implicitement sous la forme <descr> (voir la [RFC4512]).

Si un client LDAP fournit une valeur d'assertion sous la forme <descr> et si le descripteur choisi n'est pas reconnu par le serveur, la règle objectIdentifierMatch évalue alors à Indéfini.

La définition LDAP pour la syntaxe de la règle de correspondance objectIdentifierMatch est :

( 2.5.13.0 NAME 'objectIdentifierMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

La règle objectIdentifierMatch est une règle de correspondance d'égalité.

#### 4.2.27 **octetStringMatch**

La règle `octetStringMatch` compare une valeur d'assertion de la syntaxe Octet String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Octet String ou la syntaxe JPEG) dont le type ASN.1 correspondant est OCTET STRING.

La règle évalue à TRUE si et seulement si la valeur d'attribut et la valeur d'assertion ont la même longueur et si les octets correspondants (par position) sont les mêmes.

La définition LDAP pour la syntaxe de la règle de correspondance `octetStringMatch` est :

( 2.5.13.17 NAME 'octetStringMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )

La règle `octetStringMatch` est une règle de correspondance d'égalité.

#### 4.2.28 **octetStringOrderingMatch**

La règle `octetStringOrderingMatch` compare une valeur d'assertion de la syntaxe Octet String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Octet String ou la syntaxe JPEG) dont le type ASN.1 correspondant est OCTET STRING.

La règle évalue à TRUE si et seulement si la valeur d'attribut apparaît plus tôt dans l'ordre de collationnement que la valeur d'assertion. La règle compare les chaînes d'octet du premier au dernier octet, et du bit de plus fort poids au bit de plus faible poids dans l'octet. La première occurrence d'un bit différent détermine l'ordre des chaînes. Un bit zéro précède un bit un. Si les chaînes contiennent un nombre d'octets différent mais que la chaîne la plus longue est identique à la chaîne la plus courte jusqu'à la longueur de la plus courte chaîne, la plus courte chaîne précède la plus longue chaîne.

La définition LDAP pour la syntaxe de la règle de correspondance `octetStringOrderingMatch` est :

( 2.5.13.18 NAME 'octetStringOrderingMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )

La règle `octetStringOrderingMatch` est une règle de correspondance d'ordonnement.

#### 4.2.29 **telephoneNumberMatch**

La règle `telephoneNumberMatch` compare une valeur d'assertion de la syntaxe Telephone Number à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Telephone Number) dont le type ASN.1 correspondant est une PrintableString représentant un numéro de téléphone.

La règle évalue à TRUE si et seulement si la chaîne préparée de caractères de valeur d'attribut et la chaîne préparée de caractères de valeur d'assertion ont le même nombre de caractères et si les caractères correspondants ont la même combinaison binaire.

En préparant la comparaison de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation de Map, et seul le traitement `telephoneNumber` de caractère insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle de correspondance `telephoneNumberMatch` est :

( 2.5.13.20 NAME 'telephoneNumberMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )

La règle `telephoneNumberMatch` est une règle de correspondance d'égalité.

#### 4.2.30 **telephoneNumberSubstringsMatch**

La règle `telephoneNumberSubstringsMatch` compare une valeur d'assertion de la syntaxe Substring Assertion à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Telephone Number) dont le type ASN.1 correspondant est une PrintableString représentant un numéro de téléphone.

La règle évalue à TRUE si et seulement si (1) les sous-chaînes préparées de valeur d'assertion correspondent à des portions disjointes de la chaîne préparée de caractères de valeur d'attribut dans l'ordre des sous-chaînes dans la valeur d'assertion, (2) une sous-chaîne <initial>, si elle est présente, correspond au commencement de la chaîne préparée de caractères de valeur d'attribut, et (3) une sous-chaîne <final>, si elle est présente, correspond à la fin de la chaîne préparée de caractères de valeur

d'attribut. Une sous-chaîne préparée correspond à une portion de la chaîne préparée de caractères de valeur d'attribut si les caractères correspondants ont la même combinaison binaire.

En préparant la comparaison des sous-chaînes de valeur d'attribut et de valeur d'assertion, les caractères sont changés de casse dans l'étape de préparation de Map, et seul le traitement telephoneNumber de caractère insignifiant est appliqué dans l'étape de traitement de caractère insignifiant.

La définition LDAP pour la syntaxe de la règle de correspondance telephoneNumberSubstringsMatch est :

( 2.5.13.21 NAME 'telephoneNumberSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

La règle telephoneNumberSubstringsMatch est une règle de correspondance de sous-chaîne.

#### 4.2.31 uniqueMemberMatch

La règle uniqueMemberMatch compare une valeur d'assertion de la syntaxe Name And Optional UID à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Name And Optional UID) dont le type ASN.1 correspondant est NameAndOptionalUID.

La règle évalue à TRUE si et seulement si les composants <distinguishedName> de la valeur d'assertion et de la valeur d'attribut selon la règle distinguishedNameMatch et si, (1) le composant <BitString> est absent à la fois de la valeur d'attribut et de la valeur d'assertion, ou (2) si le composant <BitString> est présent à la fois dans la valeur d'attribut et dans la valeur d'assertion et si le composant <BitString> de la valeur d'assertion correspond au composant <BitString> de la valeur d'attribut selon la règle bitStringMatch.

Noter que cette règle de correspondance a été altérée par rapport à sa description dans X.520 [X.520] afin de rendre la règle de correspondance commutative. Les développeurs de serveurs devraient considérer l'utilisation de la sémantique originale de X.520 (où la correspondance était moins exacte) pour une correspondance approximative des attributs avec uniqueMemberMatch comme règle de correspondance d'égalité.

La définition LDAP pour la syntaxe de la règle de correspondance uniqueMemberMatch est :

( 2.5.13.23 NAME 'uniqueMemberMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.34 )

La règle uniqueMemberMatch est une règle de correspondance d'égalité.

#### 4.2.32 wordMatch

La règle wordMatch compare une valeur d'assertion de la syntaxe Directory String à une valeur d'attribut d'une syntaxe (par exemple, la syntaxe Directory String) dont le type ASN.1 correspondant est DirectoryString.

La règle évalue à TRUE si et seulement si le mot de la valeur d'assertion correspond, conformément à la sémantique de caseIgnoreMatch, à tout mot dans la valeur d'attribut. La définition précise d'un mot est spécifique de la mise en œuvre.

La définition LDAP pour la syntaxe de la règle wordMatch est :

( 2.5.13.32 NAME 'wordMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

## 5 Considérations sur la sécurité

En général, les codages spécifiques de LDAP pour les syntaxes définies dans le présent document ne définissent pas de codages canoniques. C'est-à-dire qu'une transformation d'un codage spécifique de LDAP en quelque autre codage (par exemple, BER) et le retour dans le codage spécifique de LDAP ne reproduira pas nécessairement exactement les octets originaux du codage spécifique de LDAP. Donc, un codage spécifique de LDAP ne devrait pas être utilisé lorsqu'un codage canonique est exigé.

De plus, les codages spécifiques de LDAP n'activent pas nécessairement la reconstruction d'un codage de remplacement des valeurs des syntaxes Directory String et DN ; par exemple, une transformation d'un codage des règles de codage distinctives (DER) [BER] en un codage spécifique de LDAP et le retour dans un codage DER peut ne pas reproduire le codage DER

d'origine. Donc, les codages spécifiques de LDAP ne devraient pas être utilisés lorsque la réversibilité en DER est nécessaire, par exemple, pour la vérification des signatures numériques. On devrait utiliser à la place un codage DER ou DER réversible.

Lors de l'interprétation des champs sensibles à la sécurité (en particulier, les champs utilisés pour accorder ou refuser l'accès), les mises en œuvre DOIVENT s'assurer que toute comparaison de règle de correspondance est faite sur la valeur abstraite sous-jacente, sans considération du codage particulier utilisé.

## 6 Remerciements

Le présent document est à l'origine une révision de la RFC 2252 par M. Wahl, A. Coulbeck, T. Howes, et S. Kille. La RFC 2252 a été produite par le groupe de travail ASID de l'IETF.

Le présent document se fonde sur des travaux du groupe de travail LDAPBIS de l'IETF. L'auteur adresse ses remerciements à Kathy Dally pour l'édition des premiers projets du présent document, et Jim Sermersheim et Kurt Zeilenga pour leurs contributions significatives à la présente révision.

## 7 Considérations relatives à l'IANA

L'Autorité d'allocation des numéros Internet (IANA) a mis à jour le registre des descripteurs LDAP [BCP64] comme indiqué par les gabarits suivants :

Sujet : Demande de mise à jour d'enregistrement de descripteur LDAP  
 Descripteur (nom abrégé) : voir le commentaire  
 Identifiant d'objet: voir le commentaire  
 Adresse personnelle & adresse de messagerie électronique à contacter pour des précisions :  
 Steven Legg <steven.legg@eb2bcom.com>  
 Usage : voir le commentaire  
 Spécification : RFC 4517  
 Auteur/Contrôleur des changements : IESG

| <b>NOM</b>                          | <b>Type</b> | <b>OID</b>                 |
|-------------------------------------|-------------|----------------------------|
| bitStringMatch                      | M           | 2.5.13.16                  |
| booleanMatch                        | M           | 2.5.13.13                  |
| caseExactIA5Match                   | M           | 1.3.6.1.4.1.1466.109.114.1 |
| caseExactMatch                      | M           | 2.5.13.5                   |
| caseExactOrderingMatch              | M           | 2.5.13.6                   |
| caseExactSubstringsMatch            | M           | 2.5.13.7                   |
| caseIgnoreIA5Match                  | M           | 1.3.6.1.4.1.1466.109.114.2 |
| caseIgnoreListMatch                 | M           | 2.5.13.11                  |
| caseIgnoreListSubstringsMatch       | M           | 2.5.13.12                  |
| caseIgnoreMatch                     | M           | 2.5.13.2                   |
| caseIgnoreOrderingMatch             | M           | 2.5.13.3                   |
| caseIgnoreSubstringsMatch           | M           | 2.5.13.4                   |
| directoryStringFirstComponentMatch  | M           | 2.5.13.31                  |
| distinguishedNameMatch              | M           | 2.5.13.1                   |
| generalizedTimeMatch                | M           | 2.5.13.27                  |
| generalizedTimeOrderingMatch        | M           | 2.5.13.28                  |
| integerFirstComponentMatch          | M           | 2.5.13.29                  |
| integerMatch                        | M           | 2.5.13.14                  |
| integerOrderingMatch                | M           | 2.5.13.15                  |
| keywordMatch                        | M           | 2.5.13.33                  |
| numericStringMatch                  | M           | 2.5.13.8                   |
| numericStringOrderingMatch          | M           | 2.5.13.9                   |
| numericStringSubstringsMatch        | M           | 2.5.13.10                  |
| objectIdentifierFirstComponentMatch | M           | 2.5.13.30                  |
| octetStringMatch                    | M           | 2.5.13.17                  |
| octetStringOrderingMatch            | M           | 2.5.13.18                  |

|                                |   |           |
|--------------------------------|---|-----------|
| telephoneNumberMatch           | M | 2.5.13.20 |
| telephoneNumberSubstringsMatch | M | 2.5.13.21 |
| uniqueMemberMatch              | M | 2.5.13.23 |
| wordMatch                      | M | 2.5.13.32 |

Le descripteur de l'identifiant d'objet 2.5.13.0 était incorrectement enregistré comme objectIdentifiersMatch ('s' étranger) dans le BCP 64. Il a été changé comme suit, avec une référence à la RFC 4517.

| NOM                   | Type | OID      |
|-----------------------|------|----------|
| objectIdentifierMatch | M    | 2.5.13.0 |

Sujet : Demande d'enregistrement de descripteur LDAP

Descripteur (nom abrégé) : caseIgnoreIA5SubstringsMatch

Identifiant d'objet : 1.3.6.1.4.1.1466.109.114.3

Adresse personnelle et adresse de messagerie de la personne à contacter pour des informations complémentaires :

Steven Legg <steven.legg@eb2bcom.com>

Usage : autres (M)

Spécification : RFC 4517

Auteur/Contrôleur des changements : IESG

## 8 Références

### 8.1 *Références normatives*

[RFC2119] Bradner, S., "Mots clé à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC 2119, mars 1997.

[RFC3629] Yergeau, F., "UTF-8, un format de transformation de ISO 10646", STD 63, RFC 3629, novembre 2003.

[RFC4234] Crocker, D., Ed. et P. Overell, "BNF augmenté pour les spécifications de syntaxe (ABNF)", RFC 4234, oct. 2005.

[RFC4510] Zeilenga, K., Ed., "Protocole léger d'accès aux répertoires (LDAP) : Descriptif des spécifications techniques", RFC 4510, juin 2006.

[RFC4511] Sermersheim, J., Ed., "Protocole léger d'accès aux répertoires (LDAP) Le protocole", RFC 4511, juin 2006.

[RFC4512] Zeilenga, K., "Protocole léger d'accès aux répertoires (LDAP) : Modèle d'informations de répertoires", juin 2006.

[RFC4514] Zeilenga, K., Ed., "Protocole léger d'accès aux répertoires (LDAP) : Représentation de chaîne des noms distinctifs", RFC 4514, juin 2006.

[RFC4518] Zeilenga, K., "Protocole léger d'accès aux répertoires (LDAP) : Préparation de chaîne internationalisée", juin 2006.

[RFC4520] Zeilenga, K., "Autorité d'allocation des numéros de l'Internet (IANA) : Considérations pour le Protocole léger d'accès aux répertoires (LDAP)", BCP 64, RFC 4520, juin 2006.

[E.123] Recommandation UIT-T E.123, 1988 Notation des numéros de téléphone nationaux et internationaux.

[FAX] Recommandation UIT-T T.4, 1993 Terminaux et protocoles pour les services télématiques – Normalisation des télécopieurs du Groupe 3 pour la transmission de documents.

[T.50] Recommandation UIT-T T.50, 1992 Alphabet international de référence (ancien alphabet international n° 5 ou AI5) – Technologies de l'information – Jeux de caractères codés à 7 bits pour l'échange d'informations.

[X.420] Recommandation UIT-T X.420 (1996) | ISO/CEI 10021-7:1997, Technologies de l'information - Systèmes de messagerie: système de messagerie de personne à personne

[X.501] Recommandation UIT-T X.501 (1993) | ISO/CEI 9594-2:1994, Technologies de l'information - Interconnexion des systèmes ouverts – L'annuaire: les modèles

[X.520] Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1994, Technologies de l'information - Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés.

[ASN.1] Recommandation UIT-T X.680 (07/02) | ISO/CEI 8824-1:2002, Technologies de l'information - Notation de syntaxe abstraite numéro un: spécification de la notation de base.

[ISO3166] ISO 3166, "Codes pour la représentation des noms de pays".

[ISO8601] ISO 8601:2004, "Éléments de données et formats d'échange – Échange d'informations-- Représentation des dates et des heures".

[UCS] ISO/CEI 10646- 1: 1993 Ensemble universel des caractères codés sur plusieurs octets (UCS) - Architecture et plan de base multilingue, (avec les amendements).

[JPEG] Format d'échange de fichier JPEG (Version 1.02). Eric Hamilton, C-Cube Microsystems, Milpitas, CA, 1992.

## 8.2. *Références informatives*

[RFC4519] Sciberras, A., Ed., "Protocole léger d'accès aux répertoires (LDAP) : Schéma pour les applications d'utilisateur", juin 2006.

[RFC4523] Zeilenga, K., "Protocole léger d'accès aux répertoires (LDAP) Définitions de schémas pour les certificats X.509", RFC 4523, juin 2006.

[X.500] Recommandation UIT-T X.500 (1993) | ISO/CEI 9594-1:1994, Technologies de l'information - Interconnexion des systèmes ouverts – L'annuaire: aperçu général des concepts, modèles et services.

[BER] Recommandation UIT-T X.690 (07/02) | ISO/CEI 8825-1:2002, Technologies de l'information - Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives

## Appendice A Récapitulation des identifiants d'objet de syntaxe

La liste suivante récapitule les identifiants d'objet alloués aux syntaxes définies dans le présent document.

| Syntaxe                        | OBJECT IDENTIFIER             |
|--------------------------------|-------------------------------|
| Attribute Type Description     | 1.3.6.1.4.1.1466.115.121.1.3  |
| Bit String                     | 1.3.6.1.4.1.1466.115.121.1.6  |
| Boolean                        | 1.3.6.1.4.1.1466.115.121.1.7  |
| Country String                 | 1.3.6.1.4.1.1466.115.121.1.11 |
| Delivery Method                | 1.3.6.1.4.1.1466.115.121.1.14 |
| Directory String               | 1.3.6.1.4.1.1466.115.121.1.15 |
| DIT Content Rule Description   | 1.3.6.1.4.1.1466.115.121.1.16 |
| DIT Structure Rule Description | 1.3.6.1.4.1.1466.115.121.1.17 |
| DN                             | 1.3.6.1.4.1.1466.115.121.1.12 |
| Enhanced Guide                 | 1.3.6.1.4.1.1466.115.121.1.21 |
| Facsimile Telephone Number     | 1.3.6.1.4.1.1466.115.121.1.22 |
| Fax                            | 1.3.6.1.4.1.1466.115.121.1.23 |
| Generalized Time               | 1.3.6.1.4.1.1466.115.121.1.24 |
| Guide                          | 1.3.6.1.4.1.1466.115.121.1.25 |
| IA5 String                     | 1.3.6.1.4.1.1466.115.121.1.26 |
| Integer                        | 1.3.6.1.4.1.1466.115.121.1.27 |
| JPEG                           | 1.3.6.1.4.1.1466.115.121.1.28 |
| LDAP Syntax Description        | 1.3.6.1.4.1.1466.115.121.1.54 |
| Matching Rule Description      | 1.3.6.1.4.1.1466.115.121.1.30 |
| Matching Rule Use Description  | 1.3.6.1.4.1.1466.115.121.1.31 |
| Name And Optional UID          | 1.3.6.1.4.1.1466.115.121.1.34 |
| Name Form Description          | 1.3.6.1.4.1.1466.115.121.1.35 |
| Numeric String                 | 1.3.6.1.4.1.1466.115.121.1.36 |
| Object Class Description       | 1.3.6.1.4.1.1466.115.121.1.37 |
| Octet String                   | 1.3.6.1.4.1.1466.115.121.1.40 |

|                             |                               |
|-----------------------------|-------------------------------|
| OID                         | 1.3.6.1.4.1.1466.115.121.1.38 |
| Other Mailbox               | 1.3.6.1.4.1.1466.115.121.1.39 |
| Postal Address              | 1.3.6.1.4.1.1466.115.121.1.41 |
| Printable String            | 1.3.6.1.4.1.1466.115.121.1.44 |
| Substring Assertion         | 1.3.6.1.4.1.1466.115.121.1.58 |
| Telephone Number            | 1.3.6.1.4.1.1466.115.121.1.50 |
| Teletex Terminal Identifier | 1.3.6.1.4.1.1466.115.121.1.51 |
| Telex Number                | 1.3.6.1.4.1.1466.115.121.1.52 |
| UTC Time                    | 1.3.6.1.4.1.1466.115.121.1.53 |

## Appendice B Changements par rapport à la RFC 2252

La présente annexe fait la liste des différences significatives entre la présente spécification et la RFC 2252.

La présente annexe n'est fournie qu'à titre d'information. Ce n'est pas une partie normative de la présente spécification.

1. La note de l'IESG a été retirée.
2. La majeure partie des Sections 4, 5 et 7 a été retirée de la RFC 4512 et révisée. Les changements des parties de ces sections déplacées de la RFC 4512 sont détaillés dans la RFC 4512.
3. Les descriptions en BNF des formats de syntaxe ont été remplacées par des spécifications en ABNF [RFC4234].
4. Les déclarations ambiguës du paragraphe 4.3 de la RFC 2252, au sujet de l'utilisation d'un mécanisme de citation de barre oblique inverse pour coder les symboles de séparateur ont été retirées. Le mécanisme d'échappement est maintenant explicitement représenté en ABNF pour les syntaxes où cette disposition s'applique.
5. La description de chacune des syntaxes LDAP a été étendue de sorte qu'elles soient moins dépendantes de la connaissance de X.500 pour leur interprétation.
6. Les relations des syntaxes LDAP avec les définitions de type ASN.1 correspondant ont été rendues explicites.
7. L'ensemble des caractères permis dans une <PrintableString> (anciennement <printablestring>) a été corrigé pour l'aligner avec le type ASN.1 PrintableString dans [ASN.1]. Précisément, le caractère guillemets doubles a été retiré et les caractères de guillemet simple et de signe égal ont été ajoutés.
8. Les valeurs des syntaxes Directory String, Printable String et Telephone Number sont maintenant exigées avec au moins un caractère.
9. Les règles <DITContentRuleDescription>, <NameFormDescription> et <DITStructureRuleDescription> ont été déplacées dans la RFC 4512.
10. Le type ASN.1 correspondant pour la syntaxe Other Mailbox a été incorporé depuis la RFC 1274.
11. Un type ASN.1 correspondant a été inventé pour la syntaxe LDAP Syntax Description.
12. La syntaxe Binary a été retirée parce qu'elle n'était pas spécifiée de façon adéquate ; il existe des mises en œuvre avec différentes interprétations incompatibles, et elle était confondue avec le codage de transfert ;binary.
13. Toutes les discussions d'options de transfert, y compris l'option ";binary", ont été retirées. Tous les impératifs concernant le transfert binaire des valeurs ont été retirées.
14. Les syntaxes Delivery Method, Enhanced Guide, Guide, Octet String, Teletex Terminal Identifier et Telex Number ont été incorporées de la RFC 2256.
15. La règle <criteria> pour les syntaxes Enhanced Guide et Guide ont été étendues pour traiter les expressions vides "et" et "ou".
16. Un codage pour la règle <ttx-value> dans la syntaxe Teletex Terminal Identifier a été défini.

17. Les syntaxes en rapport avec PKI (Certificate, Certificate List et Certificate Pair) ont été retirées. Elles sont réintroduites dans la RFC 4523 (comme l'est la syntaxe Supported Algorithm tirée de la RFC 2256).
18. La syntaxe MHS OR Address a été retirée car sa spécification (dans la RFC 2156) n'a pas la maturité d'un projet de norme.
19. La syntaxe DL Submit Permission a été retirée car elle dépend de la syntaxe MHS OR Address.
20. La syntaxe Presentation Address a été retirée car sa spécification (dans la RFC 1278) n'a pas la maturité d'un projet de norme.
21. Les syntaxes ACI Item, Access Point, Audio, Data Quality, DSA Quality, DSE Type, LDAP Schema Description, Master And Shadow Access Points, Modify Rights, Protocol Information, Subtree Specification, Supplier Information, Supplier Or Consumer et Supplier And Consumer ont été retirées. Ces syntaxes sont référencées dans la RFC 2252, mais pas définies.
22. La syntaxe LDAP Schema Definition (définie dans la RFC 2927) et la syntaxe Mail Preference ont été retirées car elles sont en dehors du domaine d'application de la spécification centrale.
23. La description de chacune des règles de correspondance a été étendue de sorte qu'elles soient moins dépendantes de la connaissance de X.500 pour leur interprétation.
24. La règle de correspondance caseIgnoreIA5SubstringsMatch tirée de la RFC 2798 a été ajoutée.
25. Les règles de correspondance caseIgnoreListSubstringsMatch, caseIgnoreOrderingMatch et caseIgnoreSubstringsMatch ont été ajoutées à la liste des règles de correspondance pour lesquelles s'appliquent les dispositions pour le traitement des caractères d'espaces blanc en tête, en queue et et adjoints multiples (maintenant dans la préparation de chaîne). Ceci est cohérent avec les définitions de ces règles de correspondance dans X.500. La règle caseIgnoreIA5SubstringsMatch a aussi été ajoutée à la liste.
26. La spécification de la règle de correspondance octetStringMatch tirée de la RFC 2256 a été ajoutée au présent document.
27. La règle de correspondance presentationAddressMatch a été retirée car elle dépend d'une syntaxe d'assertion (Presentation Address) qui n'est pas mûre pour un projet de norme.
28. La règle de correspondance protocolInformationMatch a été retirée car elle dépend d'une syntaxe d'assertion indéfinie (Protocol Information).
29. La référence définitive pour l'ASN.1 a été changée de X.208 à X.680 car X.680 est la version d'ASN.1 à laquelle se réfère X.500.
30. La spécification de la règle de correspondance caseIgnoreListSubstringsMatch tirée de la RFC 2798 & X.520 a été ajoutée.
31. Les algorithmes de préparation de chaîne ont été appliqués aux règles de correspondance de chaînes de caractères.
32. Les spécifications des règles de correspondance booleanMatch, caseExactMatch, caseExactOrderingMatch, caseExactSubstringsMatch, directoryStringFirstComponentMatch, integerOrderingMatch, keywordMatch, numericStringOrderingMatch, octetStringOrderingMatch et wordMatch tirées de la RFC 3698 & X.520 ont été ajoutées.

### Adresse de l'auteur

Steven Legg  
eB2Bcom  
Suite3, Woodhouse Corporate Centre  
935 Station Street  
Box Hill North, Victoria 3129  
AUSTRALIA

Téléphone : +61 3 9896 7830  
Télécopie : +61 3 9896 7801  
mél : [steven.legg@eb2bcom.com](mailto:steven.legg@eb2bcom.com)

### **Déclaration de copyright**

Copyright (C) The Internet Society (2006).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://www.rfc-editor.org), et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et LE CONTRIBUTEUR, L'ORGANISATION QU'IL OU ELLE REPRÉSENTE OU QUI LE/LA FINANCE (S'IL EN EST), LA INTERNET SOCIETY ET LA INTERNET ENGINEERING TASK FORCE DÉCLINENT TOUTES GARANTIES, EXPRIMÉES OU IMPLICITES, Y COMPRIS MAIS NON LIMITÉES À TOUTE GARANTIE QUE L'UTILISATION DES INFORMATIONS CI-ENCLOSES NE VIOLENT AUCUN DROIT OU AUCUNE GARANTIE IMPLICITE DE COMMERCIALISATION OU D'APTITUDE À UN OBJET PARTICULIER.

### **Propriété intellectuelle**

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

**Remerciement**Le financement de la fonction d'édition des RFC est fourni par la Administrative Support Activity (IASA) de l'IETF.