Groupe de travail sur les réseaux
Requête pour Commentaires : 2252
Catégorie : Standard

M. Wahl
Critical Angle Inc.
A. Coulbeck
Isode Inc.
T. Howes
Netscape Communications Corp.
S. Kille
Isode Limited
Décembre 1997

Traduction : Yves lescop
Lycée la croix-rouge - Brest

# Protocole allégé d'accès à un annuaire (LDAP) : Définitions de Syntaxe d'Attribut

## 1. Statut de ce document

Ce document spécifie un protocole standard d'Internet pour la communauté Internet, et ne sera éprouvé qu'après plusieurs discussions et suggestions. Merci de vous référer à l'édition courante du " Internet Official Protocol Standards " (STD1) pour l'état de standardisation et le statut de ce protocole. La distribution de ce document est illimitée.

## Copyright



## Note d'IESG

Ce document décrit un protocole d'accès à un annuaire qui fournit tant l'accès en lecture que l'accès pour mise à jour. L'accès de mise à jour exige une authentification sécurisée, mais ce document n'exige la mise en place d'aucun mécanisme d'authentification adéquat.

Selon RFC 2026, section 4.4.1, cette spécification est approuvée par IESG comme norme proposée en dépit de cette limitation, pour les raisons suivantes :

a. pour encourager la mise en place et le test d'interopérabilité de ces protocoles (avec ou sans l'accès de mise à jour) avant qu'ils soient déployés, et

b. pour encourager le déploiement et l'utilisation de ces protocoles dans des applications à lecture seule. (par exemple applications où LDAPv3 est utilisé comme langage d'interrogation pour les annuaires qui sont mis à jour par un mécanisme sécurisé autre que LDAP), et

c. pour éviter de retarder l'avancement et le déploiement d'autres protocoles standard d'Internet qui exigent la possibilité de questionner, mais pas de mettre à jour, des serveurs d'annuaire LDAPv3.

Les lecteurs sont avertis par la présente que jusqu'à ce que des mécanismes obligatoires d'authentification soient normalisés, les clients et les serveurs écrits selon cette spécification qui se servent de la fonctionnalité de mise à jour sont IMPROBABLEMENT INTEROPERABLE ou PEUVENT INTEROPERER SEULEMENT SI L'AUTHENTIFICATION EST RÉDUITE À UN NIVEAU INADMISSIBLEMENT FAIBLE.

Les implanteurs sont découragés par la présente de déployer des clients ou des serveurs LDAPv3 qui mettent en œuvre la fonctionnalité de mise à jour, jusqu'à ce qu'une norme proposée pour l'authentification obligatoire dans LDAPv3 ait été approuvée et éditée comme RFC.

## 2. Résumé

Le protocole allégé d'accès aux annuaires (LDAP) [1] exige que les teneurs des champs "AttributeValue" dans des éléments du protocole soient des chaînes de caractères d'octets. Ce document définit un ensemble de syntaxes pour LDAPv3, et les règles par lesquelles les valeurs des attributs de ces syntaxes sont représentées comme chaînes de caractères pour la transmission dans le protocole LDAP. Les syntaxes définies dans ce document sont référencées par celui-ci et par d'autres documents qui définissent des types d'attribut. Ce document définit également l'ensemble de types d'attribut que les serveurs LDAP devraient supporter.

## 3. Vue d'ensemble

Ce document définit le cadre pour le développement des schémas pour les annuaires accessibles par l'intermédiaire du protocole LDAP.

Le schéma est le recueil des définitions de type d'attribut, de définitions de classe d'objet et de toute autre information qu'un serveur utilise pour déterminer comment apparier une affirmation de valeur de filtre ou d'attribut (dans une opération de compaison) contre les attributs d'une entrée, et s'il est autorisé d'effectuer les opérations d'ajout ou de modification.

La section 4 énonce les exigences générales et les notations pour les types d'attribut, les classes d'objet, la syntaxe et les définitions des règles d'appariement.

La section 5 énumère les attributs, la section 6 les syntaxes et la section 7 les classes d'objet.

Des documents supplémentaires définissent les schémas pour représenter les objets réels comme entrées d'annuaire.

# 4. Problèmes Généraux

Ce document décrit les encodages utilisés dans un protocole d'Internet.

Les mots clés "DOIT", "NE DOIT PAS", "REQUIS", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans ce document doivent être interprétés comme décrit dans RFC 2119 [4].

Le type d'attribut et les définitions de classe d'objet sont écrites dans une représentation de chaîne de caractères des types de données "AttributeTypeDescription" et "ObjectClassDescription" définis dans X.501(93) [3]. Il est fortement conseillé aux développeurs de lire en premier la description de la manière dont le schéma est représenté dans X.500 avant de lire le reste de ce document.

## 4.1 Aspects Encodants Communs

Dans le but de définir les règles d'encodage des syntaxes d'attribut, les définitions suivantes de BNF seront utilisées. Elles sont basées sur les modèles BNF de RFC 822 [13].

```
a     = "a" / "b" / "c" / "d" / "e" / "f" / "g" / "h" / "i" /
        "j" / "k" / "l" / "m" / "n" / "o" / "p" / "q" / "r" /
        "s" / "t" / "u" / "v" / "w" / "x" / "y" / "z" / "A" /
        "B" / "C" / "D" / "E" / "F" / "G" / "H" / "I" / "J" /
        "K" / "L" / "M" / "N" / "O" / "P" / "Q" / "R" / "S" /
        "T" / "U" / "V" / "W" / "X" / "Y" / "Z"

d               = "0" / "1" / "2" / "3" / "4" /
                  "5" / "6" / "7" / "8" / "9"

hex-digit       =  d / "a" / "b" / "c" / "d" / "e" / "f" /
                        "A" / "B" / "C" / "D" / "E" / "F"

k               = a / d / "-" / ";"

p               = a / d / "'" / "(" / ")" / "+" / "," /
                  "-" / "." / "/" / ":" / "?" / " "

letterstring    = 1*a

numericstring   = 1*d

anhstring       = 1*k

keystring       = a [ anhstring ]

printablestring = 1*p

space           = 1*" "

whsp            = [ space ]

utf8            = <toute suite d'octets formée à partir d'une transformation
```

```
                    UTF-8 [9] d'un caractère venant de ISO10646 [10]>

dstring         = 1*utf8

qdstring        = whsp "'" dstring "'" whsp

qdstringlist    = [ qdstring *( qdstring ) ]

qdstrings       = qdstring / ( whsp "(" qdstringlist ")" whsp )
```

Dans le BNF suivant pour la représentation en chaîne de caractères des identificateurs d'OBJET, "descr" est la représentation syntaxique d'un descripteur d'objet, qui se compose de lettres et de chiffres, commençant par une lettre. Un IDENTIFICATEUR d'OBJET dans le format "numericoid" ne devrait pas avoir de zéros en tête (par exemple "0.9.3" est autorisé mais "0.09.3" ne devrait pas être produit).

Quand on encode des éléments "oid" en valeur, l'option encodante "descr" DEVRAIT être utilisée de préférence au "numericoid". Un descripteur d'objet est un alias plus lisible pour un numéro IDENTIFICATEUR d'OBJET, et ceux-ci (à l'endroit assigné et connu par l'implémentation) DEVRAIENT être utilisés de préférence aux "oids" numériques jusqu'au plus grand degré possible. Les exemples des descripteurs d'objet dans LDAP sont du type attribut, classe d'objet et noms de règles d'appariement.

```
 oid             = descr / numericoid

 descr           = keystring

 numericoid      = numericstring *( "." numericstring )

 woid            = whsp oid whsp

 ; jeu d' "oids" d'une autre forme
 oids            = woid / ( "(" oidlist ")" )

 oidlist         = woid *( "$" woid )

 ; descripteurs d'objet utilisés comme noms d'élément du schéma
 qdescrs         = qdescr / ( whsp "(" qdescrlist ")" whsp )

 qdescrlist      = [ qdescr *( qdescr ) ]

 qdescr          = whsp "'" descr "'" whsp
```

## 4.2. Types d'Attribut

Les types d'attribut sont décrits par des valeurs échantillon pour l'attribut du sous-schéma "attributeTypes", qui est écrit dans la syntaxe "AttributeTypeDescription". Tandis que des retours à la ligne ont été effectués pour la lisibilité, les valeurs transférées dans le protocole ne contiendraient pas d'interlignes.

"AttributeTypeDescription" est encodé selon le BNF suivant, et les productions pour l'oid, "qdescrs" et "qdstring" sont données dans la section 4.1. Les développeurs devrait noter que les

futures versions de ce document peuvent avoir augmenté ce BNF pour inclure des expressions supplémentaires. Les expressions qui commencent par les caractères "X-" sont réservées pour des expériences privées, et DOIVENT être suivies par un <qdstrings>.

```
AttributeTypeDescription = "(" whsp
      numericoid whsp              ; Identificateur de type d'Attribut
    [ "NAME" qdescrs ]             ; nom utilisé dans AttributeType
    [ "DESC" qdstring ]            ; description
    [ "OBSOLETE" whsp ]
    [ "SUP" woid ]                 ; dérivé de cet autre AttributeType
    [ "EQUALITY" woid              ; Nom de règle d'appariement
    [ "ORDERING" woid              ; Nom de règle d'appariement
    [ "SUBSTR" woid ]              ; Nom de règle d'appariement
    [ "SYNTAX" whsp noidlen whsp ] ; voir section 4.3
    [ "SINGLE-VALUE" whsp ]        ; valeurs multiples par défaut
    [ "COLLECTIVE" whsp ]          ; non collectif par défaut
    [ "NO-USER-MODIFICATION" whsp ]; modifiable par l'utilisateur par défaut
    [ "USAGE" whsp AttributeUsage ]; userApplications par défaut
    whsp ")"

AttributeUsage =
    "userApplications"     /
    "directoryOperation"   /
    "distributedOperation" / ; partagé DSA
    "dSAOperation"           ; spécifique DSA, la valeur dépend du serveur
```

Il n’est pas exigé des serveurs qu’ils fournissent la même chose ou un texte quelconque dans la partie description des valeurs de sous-schéma qu'ils maintiennent. Les serveurs DEVRAIENT fournir au moins un des champs "SUP" et "SYNTAX" pour chaque "AttributeTypeDescription".

Les serveurs DOIVENT implémenter tous les types d'attribut référencés dans les sections 5.1, 5.2 et 5.3.

Les serveurs PEUVENT identifier des noms supplémentaires et des attributs non-cités dans ce document, et s'ils font ainsi, DOIVENT éditer les définitions des types dans l'attribut "attributeTypes" de leurs entrées de sous-schéma.

Les réalisateurs de schéma NE DOIVENT PAS créer les définitions d'attribut dont les noms sont en conflit avec des attributs définis pour être utilisés avec LDAP dans les normes RFC existantes.

Un "AttributeDescription" peut être utilisé comme valeur dans une partie "NAME" d'un "AttributeTypeDescription". Notez que ceux-ci sont insensibles à la casse.

Notez que "AttributeTypeDescription" n'énumère pas les règles d'appariement qui peuvent être utilisées avec ce type d'attribut dans le filtre de recherche "extensibleMatch". Ceci est fait en utilisant l'attribut "matchingRuleUse" décrit dans la section 4.5.

Ce document affine la description du schéma de X.501 en exigeant que le champ syntaxe dans un "AttributeTypeDescription" soit une représentation en chaîne de caractères d'un IDENTIFICATEUR d'OBJET pour la définition de syntaxe en chaîne de caractères LDAP, et une

indication facultative de la longueur maximum d'une valeur de cet attribut (défini dans la section 4.3.2).

## 4.3. Syntaxes

Cette section définit les exigences générales pour la syntaxe des encodages de valeur d'attribut LDAP. On s'attend à ce que tous les documents définissant la syntaxe des encodages d'attribut pour l'usage avec LDAP répondent à ces exigences.

Les règles encodantes définies pour une syntaxe d'attribut donnée doivent produire des chaînes de caractères d'octet. Jusqu'au plus grand degré possible, les chaînes de caractères encodées devraient être utilisables sous leur forme encodée native pour l'affichage. En particulier, les règles encodantes pour des syntaxes d'attribut définissant des valeurs non-binaires devraient produire des chaînes de caractères qui peuvent être affichées avec peu ou pas de traduction par des clients mettant en application LDAP. Il y a quelques cas cependant (par exemple audio), où il n'est pas pratique de produire une représentation imprimable, et les clients NE DOIVENT PAS supposer qu'une syntaxe non reconnue est une représentation en chaîne de caractères.

Dans les encodages où une chaîne de caractères arbitraire, pas un nom différencié, est utilisée en tant qu'élément d'une plus grande production, et d'autres en tant qu'élément d'un nom différencié, un mécanisme de cotation antislash est employé pour échapper au caractère symbolique de séparation (comme "'", "$" ou "#") s'il devait apparaître dans cette chaîne de caractères. L'antislash est suivi d'une paire de chiffres hexadécimaux représentant le prochain caractère. Un antislash lui-même dans la chaîne de caractères qui fait partie d'une plus grande syntaxe est toujours transmis en tant que '\5C 'ou '\5c '. Un exemple est donné dans la section 6.27.

Des syntaxes sont également définies pour les règles d'appariements dont la syntaxe de valeur d'affirmation est différente de la syntaxe de valeur d'attribut.

### *4.3.1 Transfert binaire des valeurs*

Ce format encodant est utilisé si le codage binaire est demandé par le client pour un attribut, ou si la syntaxe du nom d'attribut est "1.3.6.1.4.1.1466.115.121.1.5". Le contenu du champ LDAP "AttributeValue" ou "AssertionValue" est un exemple encodé BER de la valeur d'attribut ou une valeur d'assertion d'une règle d'appariement de type ASN.1 comme défini pour une utilisation avec le X.500. (le premier octet à l'intérieur de l'emballage de CHAÎNE DE CARACTÈRES est un octet d'étiquette. Cependant, la CHAÎNE DE CARACTÈRES est encore encodée dans sa forme primitive).

Tous les serveurs DOIVENT mettre en application cette forme pour les deux valeurs d'attribut se produisant dans des réponses de recherche, et les valeurs d'attribut d'analyse dans des demandes d'ajout, de comparaison et de modification, si le type d'attribut est identifié et si le nom de syntaxe d'attribut est celui de binaire. Les clients qui demandent que tous les attributs soient retournés des entrées DOIVENT être disposés à recevoir des valeurs en binaire (par exemple "userCertificate;binary"), et NE DEVRAIENT PAS simplement afficher des valeurs binaires ou non reconnues aux utilisateurs.

### *4.3.2. Syntaxe d'Identificateurs d'Objet*

Des syntaxes pour l'usage avec LDAP sont nommées par les identificateurs d'OBJET, qui sont des chaînes de caractères décimales pointées. Celles-ci ne sont pas destinées à être affichées aux utilisateurs.

```
noidlen = numericoid [ "{" len "}" ]

len     = numericstring
```

Le tableau suivant présente certaines des syntaxes qui ont été définies pour LDAP jusqu'ici. La colonne LH suggère si une valeur de cette syntaxe est probablement une chaîne de caractères lisible par l'homme. Les clients et les serveurs n'ont pas besoin de mettre en application toutes les syntaxes énumérées ici, et PEUVENT mettre en application d'autres syntaxes.

D'autres documents peuvent définir des syntaxes supplémentaires. Cependant, la définition des syntaxes arbitraires supplémentaires est fortement désapprouvée puisqu'elle gênera l'interopérabilité : les réalisations de client et de serveur d'aujourd'hui n'ont généralement pas la capacité d'identifier dynamiquement de nouvelles syntaxes. Dans la plupart des cas les attributs seront définis avec la syntaxe pour des chaînes de caractères de l'annuaire.

| Valeur étant représentée | LH | IDENTIFICATEUR D'OBJET |
|---|---|---|
| ACI Item | N | 1.3.6.1.4.1.1466.115.121.1.1 |
| Access Point | O | 1.3.6.1.4.1.1466.115.121.1.2 |
| Attribute Type Description | O | 1.3.6.1.4.1.1466.115.121.1.3 |
| Audio | N | 1.3.6.1.4.1.1466.115.121.1.4 |
| Binary | N | 1.3.6.1.4.1.1466.115.121.1.5 |
| Bit String | O | 1.3.6.1.4.1.1466.115.121.1.6 |
| Boolean | O | 1.3.6.1.4.1.1466.115.121.1.7 |
| Certificate | N | 1.3.6.1.4.1.1466.115.121.1.8 |
| Certificate List | N | 1.3.6.1.4.1.1466.115.121.1.9 |
| Certificate Pair | N | 1.3.6.1.4.1.1466.115.121.1.10 |
| Country String | O | 1.3.6.1.4.1.1466.115.121.1.11 |
| DN | O | 1.3.6.1.4.1.1466.115.121.1.12 |
| Data Quality Syntax | O | 1.3.6.1.4.1.1466.115.121.1.13 |
| Delivery Method | O | 1.3.6.1.4.1.1466.115.121.1.14 |
| Directory String | O | 1.3.6.1.4.1.1466.115.121.1.15 |
| DIT Content Rule Description | O | 1.3.6.1.4.1.1466.115.121.1.16 |
| DIT Structure Rule Description | O | 1.3.6.1.4.1.1466.115.121.1.17 |
| DL Submit Permission | O | 1.3.6.1.4.1.1466.115.121.1.18 |
| DSA Quality Syntax | O | 1.3.6.1.4.1.1466.115.121.1.19 |
| DSE Type | O | 1.3.6.1.4.1.1466.115.121.1.20 |
| Enhanced Guide | O | 1.3.6.1.4.1.1466.115.121.1.21 |
| Facsimile Telephone Number | O | 1.3.6.1.4.1.1466.115.121.1.22 |
| Fax | N | 1.3.6.1.4.1.1466.115.121.1.23 |
| Generalized Time | O | 1.3.6.1.4.1.1466.115.121.1.24 |
| Guide | O | 1.3.6.1.4.1.1466.115.121.1.25 |
| IA5 String | O | 1.3.6.1.4.1.1466.115.121.1.26 |
| INTEGER | O | 1.3.6.1.4.1.1466.115.121.1.27 |
| JPEG | N | 1.3.6.1.4.1.1466.115.121.1.28 |
| LDAP Syntax Description | O | 1.3.6.1.4.1.1466.115.121.1.54 |

```
LDAP Schema Definition          O  1.3.6.1.4.1.1466.115.121.1.56
LDAP Schema Description         O  1.3.6.1.4.1.1466.115.121.1.57
Master And Shadow Access Points O  1.3.6.1.4.1.1466.115.121.1.29
Matching Rule Description       O  1.3.6.1.4.1.1466.115.121.1.30
Matching Rule Use Description   O  1.3.6.1.4.1.1466.115.121.1.31
Mail Preference                 O  1.3.6.1.4.1.1466.115.121.1.32
MHS OR Address                  O  1.3.6.1.4.1.1466.115.121.1.33
Modify Rights                   O  1.3.6.1.4.1.1466.115.121.1.55
Name And Optional UID           O  1.3.6.1.4.1.1466.115.121.1.34
Name Form Description           O  1.3.6.1.4.1.1466.115.121.1.35
Numeric String                  O  1.3.6.1.4.1.1466.115.121.1.36
Object Class Description        O  1.3.6.1.4.1.1466.115.121.1.37
Octet String                    O  1.3.6.1.4.1.1466.115.121.1.40
OID                             O  1.3.6.1.4.1.1466.115.121.1.38
Other Mailbox                   O  1.3.6.1.4.1.1466.115.121.1.39
Postal Address                  O  1.3.6.1.4.1.1466.115.121.1.41
Protocol Information            O  1.3.6.1.4.1.1466.115.121.1.42
Presentation Address            O  1.3.6.1.4.1.1466.115.121.1.43
Printable String                O  1.3.6.1.4.1.1466.115.121.1.44
Substring Assertion             O  1.3.6.1.4.1.1466.115.121.1.58
Subtree Specification           O  1.3.6.1.4.1.1466.115.121.1.45
Supplier Information            O  1.3.6.1.4.1.1466.115.121.1.46
Supplier Or Consumer            O  1.3.6.1.4.1.1466.115.121.1.47
Supplier And Consumer           O  1.3.6.1.4.1.1466.115.121.1.48
Supported Algorithm             N  1.3.6.1.4.1.1466.115.121.1.49
Telephone Number                O  1.3.6.1.4.1.1466.115.121.1.50
Teletex Terminal Identifier     O  1.3.6.1.4.1.1466.115.121.1.51
Telex Number                    O  1.3.6.1.4.1.1466.115.121.1.52
UTC Time                        O  1.3.6.1.4.1.1466.115.121.1.53
```

Une limite supérieure minimum suggérée sur le nombre de caractères en valeur avec une syntaxe basée sur les chaînes de caractères, ou le nombre d'octets en valeur pour toutes autres syntaxes, peut être indiquée en ajoutant ce compte de limite à l'intérieur d'accolades qui suivent le nom de syntaxe IDENTIFICATEUR d'OBJET dans une description de type d'attribut. Cette limite n'est pas une partie du nom de syntaxe elle-même. Par exemple, "1.3.6.4.1.1466.0{64}" suggère que les implémentations du serveur devraient permettre à une chaîne de caractères d'être longue de 64 caractères, bien qu'elles puissent permettre de plus longues chaînes de caractères. Notez qu'un caractère simple de la syntaxe de chaîne de caractères d'annuaire peut être encodé sur plus d'un octet puisque UTF-8 est un codage de longueur variable.

### *4.3.3. Description de Syntaxe*

Le BNF suivant peut être employé pour associer une description courte à une syntaxe IDENTIFICATEUR d'OBJET . Les développeurs devrait noter que les futures versions de ce document peuvent augmenter cette définition pour inclure des termes supplémentaires. Les termes dont l'identificateur commence par "X-" sont réservés pour des expériences privées, et DOIVENT être suivis par des <qdstrings>.

```
SyntaxDescription = "(" whsp
    numericoid whsp
    [ "DESC" qdstring ]
```

```
            whsp ")"
```

## 4.4. Classes d'Objet

Le format pour la représentation des classes d'objet est défini dans X.501 [3]. En général chaque entrée contiendra une classe abstraite ("top" ou "alias"), au moins une classe d'objet structurale, et zéro ou plus de classes d'objet auxiliaires. Qu'une classe d'objet soit abstraite, structurale ou auxiliaire elle est définie quand l'identificateur de classe d'objet est assigné. Une définition de classe d'objet ne devrait pas être changée sans qu'un nouvel identificateur lui soit assigné.

Les descriptions de classe d'objet sont écrites selon le BNF suivant. Les développeurs devrait noter que les futures versions de ce document peuvent augmenter cette définition pour inclure des termes supplémentaires. Les termes dont l'identificateur commence par "X-" sont réservés pour des expériences privées, et DOIVENT être suivis d'un codage <qdstrings>.

```
ObjectClassDescription = "(" whsp
    numericoid whsp      ; ObjectClass identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" whsp ]
    [ "SUP" oids ]       ; Superior ObjectClasses
    [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) whsp ]
                         ; default structural
    [ "MUST" oids ]      ; AttributeTypes
    [ "MAY" oids ]       ; AttributeTypes
whsp ")"
```

Ceux-ci sont décrits comme valeurs d'échantillon pour l'attribut de sous-schéma "objectClasses" pour un serveur qui met en application le schéma de LDAP. Tandis que des retours à la ligne ont été effectués pour la lisibilité, les valeurs transférées dans le protocole ne contiendraient pas d'interlignes.

Les serveurs DEVRAIENT mettre en application toutes les classes d'objet référencées dans la section 7, excepté "extensibleObject", qui est facultatif. Les serveurs PEUVENT mettre en application les classes supplémentaires d'objet non citées dans ce document, et s'ils le font ainsi, DOIVENT éditer les définitions des classes dans l'attribut "objectClasses" de leurs entrées de sous-schéma.

Les réalisateurs de schéma NE DOIVENT PAS créer des définitions de classe d'objet dont les noms sont en conflit avec des attributs définis pour l'usage avec LDAP dans la norme RFC existante.

## 4.5. Règles d'appariement

Des règles d'appariement sont employées par les serveurs pour comparer des valeurs d'attribut à des valeurs d'affirmation en exécutant des opérations de recherche et de comparaison. Elles sont également employées pour identifier la valeur à ajouter ou à effacer quand on modifie des entrées, et sont utilisées en comparant un prétendu nom différencié au nom d'une entrée.

La plupart des attributs indiqués dans ce document auront une règle d'appariement d'égalité définie.

Les descriptions de règle d'appariement sont écrites selon le BNF suivant. Les développeurs devrait noter que les futures versions de ce document ont pu avoir augmenté ce BNF pour inclure des termes supplémentaires. Les termes dont l'identificateur commence par "X-" sont réservées pour des expériences privées, et DOIVENT être suivies d'un codage <qdstrings>.

```
MatchingRuleDescription = "(" whsp
    numericoid whsp  ; MatchingRule identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" whsp ]
    "SYNTAX" numericoid
whsp ")"
```

Les valeurs du "matchingRuleUse" énumèrent les attributs qui conviennent pour l'usage avec une règle d'appariement extensible.

```
MatchingRuleUseDescription = "(" whsp
    numericoid whsp  ; MatchingRule identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" ]
   "APPLIES" oids    ; AttributeType identifiers
whsp ")"
```

Les serveurs qui supportent les règles d'appariement et "extensibleMatch" DEVRAIENT mettre en application toutes les règles d'appariement de la section 8.

Les serveurs PEUVENT mettre en application des règles d'appariement supplémentaires non citées dans ce document, et s'ils font ainsi, DOIVENT éditer les définitions des règles d'appariement dans l'attribut "matchingRules" de leurs entrées de sous-schéma. Si le serveur supporte "extensibleMatch", alors le serveur DOIT éditer la relation entre les règles d'appariement et les attributs dans l'attribut "matchingRuleUse".

Par exemple, un serveur qui met en application une règle d'appariement de définition privée pour rechercher des associations “ tonalité semblable ” sur des attributs évalués Chaîne de caractères de l'annuaire inclurait ce qui suit dans l'entrée de sous-schéma (1.2.3.4.5 est un exemple, l'OID d'une règle d'appariement réelle seraient différents) :

```
matchingRule: ( 1.2.3.4.5 NAME 'soundAlikeMatch'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Si cette règle d'appariement pouvait être utilisée avec les attributs 2.5.4.41 et 2.5.4.15, ce qui suit serait également présent :

```
matchingRuleUse: ( 1.2.3.4.5 APPLIES (2.5.4.41 $ 2.5.4.15) )
```

Un client pourrait alors se servir de cette règle d'appariement en envoyant une opération de

recherche dans laquelle le filtre est du choix "extensibleMatch", le champ "matchingRule" est "soundAlikeMatch", et le champ type est "2.5.4.41" ou "2.5.4.15".

# 5. Types d'Attribut

Toutes les implémentations de serveur LDAP DOIVENT identifier les types d'attribut définis dans cette section.

Les serveurs DEVRAIENT également identifier tous les attributs de la section 5 de [12].

## 5.1. Attributs Opérationnels Standard

Les serveurs DOIVENT conserver les valeurs de ces attributs selon les définitions en X.501(93).

### *5.1.1. "createTimestamp", création d'un timbre de temps*

Cet attribut DEVRAIT apparaître dans les entrées qui ont été créées en utilisant l'opération d'ajout.

```
( 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 )
```

### *5.1.2. "modifyTimestamp", modification d'un timbre de temps*

Cet attribut DEVRAIT apparaître dans les entrées qui ont été modifiées en utilisant l'opération de modification.

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

### *5.1.3. "creatorsName", nom du créateur*

Cet attribut DEVRAIT apparaître dans les entrées qui ont été créées en utilisant l'opération d'ajout.

```
( 2.5.18.3 NAME 'creatorsName' EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE NO-USER-MODIFICATION USAGE directoryOperation )
```

### *5.1.4. "modifiersName", nom du modificateur*

Cet attribut DEVRAIT apparaître dans les entrées qui ont été modifiées en utilisant

l'opération de modification.

```
( 2.5.18.4 NAME 'modifiersName' EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE NO-USER-MODIFICATION USAGE directoryOperation )
```

### *5.1.5. "subschemaSubentry", sous-entrée de sous-schéma*

La valeur de cet attribut est le nom d'une entrée de sous-schéma (ou de sous-entrée si le serveur est basé sur X.500(93)) dans laquelle le serveur rend disponibles des attributs indiquant le schéma.

```
( 2.5.18.10 NAME 'subschemaSubentry'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 NO-USER-MODIFICATION
  SINGLE-VALUE USAGE directoryOperation )
```

### *5.1.6. "attributeTypes", types d'attribut*

Cet attribut est typiquement placé dans l'entrée de sous-schéma.

```
( 2.5.21.5 NAME 'attributeTypes'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.3 USAGE directoryOperation )
```

### *5.1.7. "objectClasses", classes d'objet*

Cet attribut est typiquement placé dans l'entrée de sous-schéma.

```
( 2.5.21.6 NAME 'objectClasses'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.37 USAGE directoryOperation )
```

### *5.1.8. "matchingRules", règles d'appariement*

Cet attribut est typiquement placé dans l'entrée de sous-schéma.

```
( 2.5.21.4 NAME 'matchingRules'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.30 USAGE directoryOperation )
```

### *5.1.9. "matchingRuleUse", utilisation de règle d'appariement*

Cet attribut est typiquement placé dans l'entrée de sous-schéma.

```
( 2.5.21.8 NAME 'matchingRuleUse'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.31 USAGE directoryOperation )
```

## 5.2. Attributs Opérationnels de LDAP

Ces attributs sont seulement présents dans la racine DSE (voir [1] et [3]).

Les serveurs DOIVENT identifier ces noms d'attribut, mais on n'exige pas qu'un serveur fournissent des valeurs pour ces attributs, quand l'attribut correspond à un dispositif que le serveur n'implémente pas.

### *5.2.1. "namingContexts", contextes de nommage*

Les valeurs de cet attribut correspondent aux contextes de nommage que ce serveur maîtrise ou recopie. Si le serveur ne maîtrise aucune information (par exemple c'est une passerelle LDAP à un annuaire X.500 public) cet attribut sera absent. Si le serveur pense qu'il contient l'annuaire entier, l'attribut aura une valeur unique, et cette valeur sera la chaîne de caractères vide (indiquant le DN nul de la racine). Cet attribut permettra à un client de choisir les objets de base appropriés pour la recherche quand il a contacté un serveur.

```
( 1.3.6.1.4.1.1466.101.120.5 NAME 'namingContexts'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE dSAOperation )
```

### *5.2.2. "altServer", serveur alternatif*

Les valeurs de cet attribut sont des URL d'autres serveurs qui peuvent être contactés quand ce serveur devient indisponible. Si le serveur ne sait pas du tout quels autres serveurs pourraient être utilisés cet attribut sera absent. Les clients peuvent mémoriser cette information au cas où leur serveur LDAP préféré deviendrait plus tard indisponible.

```
( 1.3.6.1.4.1.1466.101.120.6 NAME 'altServer'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE dSAOperation )
```

### *5.2.3. "supportedExtension", extensions supportées*

Les valeurs de cet attribut sont des identificateurs d'OBJET identifiant les opérations étendues supportées par le serveur.

Si le serveur ne supporte aucune extension cet attribut sera absent.

```
( 1.3.6.1.4.1.1466.101.120.7 NAME 'supportedExtension'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 USAGE dSAOperation )
```

### *5.2.4. "supportedControl", contrôles supportés*

Les valeurs de cet attribut sont les identificateurs d'OBJET identifiant les commandes que le serveur supporte. Si le serveur ne supporte aucune commande, cet attribut sera absent.

```
( 1.3.6.1.4.1.1466.101.120.13 NAME 'supportedControl'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 USAGE dSAOperation )
```

#### *5.2.5. "supportedSASLMechanisms", mécanismes SASL supportés*

Les valeurs de cet attribut sont les noms des mécanismes de SASL supportés par le serveur. Si le serveur ne supporte aucun mécanisme cet attribut sera absent.

```
( 1.3.6.1.4.1.1466.101.120.14 NAME 'supportedSASLMechanisms'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE dSAOperation )
```

#### *5.2.6. "supportedLDAPVersion", versions LDAP supportées*

Les valeurs de cet attribut sont les versions du protocole LDAP implantées sur le serveur.

```
( 1.3.6.1.4.1.1466.101.120.15 NAME 'supportedLDAPVersion'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 USAGE dSAOperation )
```

### 5.3. Attribut de Sous-schéma de LDAP

Cet attribut est typiquement placé dans l'entrée de sous-schéma.

#### *5.3.1. "ldapSyntaxes", syntaxes LDAP*

Les serveurs PEUVENT employer cet attribut pour énumérer les syntaxes qui sont mises en application. Chaque valeur correspond à une syntaxe.

```
( 1.3.6.1.4.1.1466.101.120.16 NAME 'ldapSyntaxes'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.54 USAGE directoryOperation )
```

### 5.4. Attributs du sous-schéma X.500

Ces attributs sont situés dans l'entrée de sous-schéma. Tous les serveurs DEVRAIENT identifier leur nom, bien qu'en général seulement les serveurs X.500 mettent en application leur fonctionnalité.

#### *5.4.1. "dITStructureRules"*

```
( 2.5.21.1 NAME 'dITStructureRules' EQUALITY integerFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.17 USAGE directoryOperation )
```

#### *5.4.2. "nameForms"*

```
( 2.5.21.7 NAME 'nameForms'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.35 USAGE directoryOperation )
```

#### *5.4.3. "ditContentRules"*

```
( 2.5.21.2 NAME 'dITContentRules'
  EQUALITY objectIdentifierFirstComponentMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.16 USAGE directoryOperation )
```

# 6. Syntaxes

Les serveurs DEVRAIENT identifier toutes les syntaxes décrites dans cette section.

## 6.1. Description de Type d'Attribut

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

Les valeurs dans cette syntaxe sont encodées selon le BNF indiqué au début de la section 4.2. Par exemple,

```
( 2.5.4.0 NAME 'objectClass'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

## 6.2. Binaire

```
( 1.3.6.1.4.1.1466.115.121.1.5 DESC 'Binary' )
```

Les valeurs dans cette syntaxe sont encodées comme décrit dans la section 4.3.1.

## 6.3. Chaîne binaire

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

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

```
bitstring = "'" *binary-digit "'B"
binary-digit = "0" / "1"
```

Exemple : '0101111101'B

## 6.4. Booléen

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

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

```
boolean = "TRUE" / "FALSE"
```

Les valeurs booléennes ont un codage à "TRUE" si elles sont logiquement vraies, et ont un codage à "FALSE" autrement.

## 6.5. Certificat

```
( 1.3.6.1.4.1.1466.115.121.1.8 DESC 'Certificate' )
```

En raison des changements de X.509(1988), de X.509(1993) et de changements supplémentaires à la définition ASN.1 pour le support des extensions de certificat, aucune représentation de chaîne de caractères n'est définie, et les valeurs dans cette syntaxe DOIVENT seulement être transférées en utilisant le codage binaire, en demandant ou retournant des attributs avec une description "userCertificate;binary" ou "caCertificate;binary". L'utilisation de la notation BNF du RFC 1778 pour "user certificate" n'est pas recommandée.

### 6.6. Liste de Certificat

```
( 1.3.6.1.4.1.1466.115.121.1.9 DESC 'Certificate List' )
```

En raison de l'incompatibilité des définitions des listes de révocation du X.509(1988) et du X.509(1993), des valeurs dans cette syntaxe DOIVENT seulement être transférées en utilisant un codage binaire, en demandant ou retournant des attributs avec une description "certificateRevocationList;binary" ou "authorityRevocationList;binary". L'utilisation de la notation BNF du RFC 1778 pour "Authority Revocation List" n'est pas recommandée.

### 6.7. Paire de Certificat

```
( 1.3.6.1.4.1.1466.115.121.1.10 DESC 'Certificate Pair' )
```

Puisque le certificat est transporté en binaire, les valeurs dans cette syntaxe DOIVENT seulement être transférées en utilisant un codage binaire, en demandant ou retournant une description d'attribut "crossCertificatePair;binary". L'utilisation de la notation BNF du RFC 1778 pour la "Certificate Pair" n'est pas recommandée.

### 6.8. Chaîne de caractères de Pays

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

Une valeur en cette syntaxe est encodée de même qu'une valeur de syntaxe de chaîne de caractères d'annuaire. Notez que cette syntaxe est limitée aux valeurs d'exactement deux caractères imprimables, comme cité dans ISO 3166 [14].

```
CountryString  = p p
```

Exemple : US

### 6.9. DN

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

Les valeurs dans la syntaxe des noms différenciés sont encodées pour avoir la représentation définie dans [5]. Notez que cette représentation n'est pas réversible à un codage ASN.1 utilisé dans le X.500 pour des noms différenciés, comme le CHOIX de n'importe quel élément de

"DirectoryString" dans un RDN n'est plus connu.

Exemples (de [5]) :
CN=Steve Kille,O=Isode Limited,C=GB
OU=Sales+CN=J. Smith,O=Widget Inc.,C=US
CN=L. Eagle,O=Sue\, Grabbit and Runn,C=GB
CN=Before\0DAfter,O=Test,C=GB
1.3.6.1.4.1.1466.0=#04024869,O=Test,C=GB
SN=Lu\C4\8Di\C4\87

## 6.10. Chaîne de caractères d'annuaire

```
( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String' )
```

Une chaîne de caractères dans cette syntaxe est encodée sous la forme UTF-8 d'ISO 10646 (une version élaborée d'Unicode). Les serveurs et les clients DOIVENT être préparés recevoir des encodages de caractères arbitraires Unicode, y compris des caractères non assignés actuellement à un jeu de caractères.

Pour des caractères sous la forme imprimable, la valeur est encodée comme la chaîne de caractères elle-même.

Si elle est de la forme de chaîne Teletex, alors les caractères sont transcrits en caractères équivalents de chaîne de caractères Universelle, et encodés en UTF-8 [9].

Si elle est de la forme "UniversalString" ou "BMPString" [10], UTF-8 est employé pour les encoder.

Note : la forme "DirectoryString" n'est pas indiquée dans le protocole à moins que la valeur d'attribut soit transportée en binaire. Les serveurs qui convertissent en DAP DOIVENT choisir une forme appropriée. Les serveurs NE DOIVENT PAS rejeter des valeurs simplement parce qu'elles contiennent des caractères légaux d'Unicode en dehors de l'intervalle de l'ASCII imprimable.

Exemple :
Ceci est une chaîne de caractères de DirectoryString contenant #!%#@

## 6.11. Description de Règle de contenu de DIT

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

Des valeurs dans cette syntaxe sont encodées selon le BNF suivant. Les développeurs devraient noter que les futures versions de ce document ont pu avoir augmenté ce BNF pour inclure des termes supplémentaires.

```
DITContentRuleDescription = "("
    numericoid   ; Structural ObjectClass identifier
    [ "NAME" qdescrs ]
```

```
    [ "DESC" qdstring ]
    [ "OBSOLETE" ]
    [ "AUX" oids ]    ; Auxiliary ObjectClasses
    [ "MUST" oids ]   ; AttributeType identifiers
    [ "MAY" oids ]    ; AttributeType identifiers
    [ "NOT" oids ]    ; AttributeType identifiers
   ")"
```

## 6.12. Numéro de Téléphone de Fac-similé

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

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

```
fax-number    = printablestring [ "$" faxparameters ]
faxparameters = faxparm / ( faxparm "$" faxparameters )
faxparm = "twoDimensional" / "fineResolution" /
          "unlimitedLength" /
          "b4Length" / "a3Width" / "b4Width" / "uncompressed"
```

Dans ce qui précède, la première chaîne de caractère imprimable est le numéro de téléphone, basé sur E.123 [15], et les jetons "faxparm" représentent des paramètres de fax.

## 6.13. Fax

```
( 1.3.6.1.4.1.1466.115.121.1.23 DESC 'Fax' )
```

Les valeurs dans cette syntaxe sont encodées comme si elles étaient des chaînes de caractères contenant des images de fax du groupe 3 comme défini dans [7].

## 6.14. Temps Généralisé

```
( 1.3.6.1.4.1.1466.115.121.1.24 DESC 'Generalized Time' )
```

Les valeurs dans cette syntaxe sont encodées en tant que chaînes de caractères imprimables, représentées comme indiqué dans X.208. Notez que le fuseau horaire doit être indiqué. On recommande vivement d'utiliser le temps GMT. Par exemple,

199412161032Z

## 6.15. Chaîne de caractères AI5 (Alphabet International n°5)

```
( 1.3.6.1.4.1.1466.115.121.1.26 DESC 'IA5 String' )
```

Le codage d'une valeur dans cette syntaxe est la valeur de la chaîne de caractères elle-même.

## 6.16. Nombre ENTIER

```
( 1.3.6.1.4.1.1466.115.121.1.27 DESC 'INTEGER' )
```

Les valeurs dans cette syntaxe sont encodées comme représentation décimale de leurs valeurs, avec chaque chiffre décimal représenté par son caractère équivalent. Ainsi le nombre 1321 est représenté par la chaîne de caractères "1321".

### 6.17. JPEG

```
( 1.3.6.1.4.1.1466.115.121.1.28 DESC 'JPEG' )
```

Les valeurs dans cette syntaxe sont encodées comme chaînes de caractères contenant des images JPEG dans le format d'échange de fichier de JPEG (JFIF), comme décrit dans [8].

### 6.18. Description de Règle d'appariement

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

Des valeurs du type "matchingRules" sont encodées comme chaînes de caractères selon le BNF indiqué dans la section 4.5.

### 6.19. Description d'Utilisation de Règle d'appariement

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

Les valeurs du type "matchingRuleUse" sont encodées comme chaînes de caractères selon le BNF indiqué dans la section 4.5.

### 6.20. MHS ou Adresse

```
( 1.3.6.1.4.1.1466.115.121.1.33 DESC 'MHS OR Address' )
```

Les valeurs dans cette syntaxe sont encodées comme chaînes de caractères, selon le format défini dans [11].

### 6.21. Nom et option UID

```
( 1.3.6.1.4.1.1466.115.121.1.34 DESC 'Name And Optional UID' )
```

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

NameAndOptionalUID = DistinguishedName [ "#" bitstring ]

Bien que le caractère '#' puisse apparaître dans une représentation en chaîne de caractères d'un nom différencié, aucune citation spéciale supplémentaire n'est faite. Cette syntaxe a été ajoutée ultérieurement à RFC 1778.

Exemple :

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

## 6.22. Description de Forme du nom

```
( 1.3.6.1.4.1.1466.115.121.1.35 DESC 'Name Form Description' )
```

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant. Les développeurs devrait noter que les futures versions de ce document ont pu avoir augmenté ce BNF pour inclure des termes supplémentaires.

```
NameFormDescription = "(" whsp
    numericoid whsp  ; NameForm identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" whsp ]
    "OC" woid         ; Structural ObjectClass
    "MUST" oids       ; AttributeTypes
    [ "MAY" oids ]    ; AttributeTypes
whsp ")"
```

## 6.23. Chaîne de caractères Numérique

```
( 1.3.6.1.4.1.1466.115.121.1.36 DESC 'Numeric String' )
```

Le codage d'une chaîne de caractères dans cette syntaxe est la valeur de la chaîne de caractères elle-même. Exemple : 1997

## 6.24. Description de Classe d'Objet

```
( 1.3.6.1.4.1.1466.115.121.1.37 DESC 'Object Class Description' )
```

Les valeurs dans cette syntaxe sont encodées selon le BNF de la section 4.4.

## 6.25. OID

```
( 1.3.6.1.4.1.1466.115.121.1.38 DESC 'OID' )
```

Les valeurs dans la syntaxe d'identificateur d'objet sont encodées selon le BNF de la section 4.1 pour l'"oid".

Exemple :
    1.2.3.4
    cn

## 6.26. Autre Boîte aux lettres

```
( 1.3.6.1.4.1.1466.115.121.1.39 DESC 'Other Mailbox' )
```

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

```
  otherMailbox = mailbox-type "$" mailbox
```

```
mailbox-type = printablestring
mailbox = <an encoded IA5 String>
```

Dans ce qui précède, le type boîte aux lettres représente le type de système de courrier dans lequel la boîte aux lettres réside, par exemple "MCIMail" ; et la boîte aux lettres est la boîte aux lettres réelle dans le système de courrier défini par le type boîte aux lettres ("mailbox-type").

### 6.27. Adresse Postale

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

Les valeurs dans cette syntaxe sont encodées selon le BNF suivant :

postal-address = dstring *( "$" dstring )

Dans ce qui précède, chaque composant "dstring" d'une valeur d'adresses postale est encodé comme une valeur de type syntaxe de chaîne de caractères d'annuaire. Les antislashs et les caractères "dollar", s'ils apparaissent dans le composant, sont cités comme décrit dans la section 4.3. Beaucoup de serveurs limitent l'adresse postale à six lignes de trente caractères.

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

### 6.28. Adresse de Présentation

```
( 1.3.6.1.4.1.1466.115.121.1.43 DESC 'Presentation Address' )
```

Les valeurs dans cette syntaxe sont encodées avec la représentation décrite dans RFC 1278 [6].

### 6.29. Chaîne de caractères Imprimable

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

Le codage d'une valeur dans cette syntaxe est la valeur de chaîne de caractères elle-même. "PrintableString" est limité aux caractères de la production p de la section 4.1.

Exemple :
Ceci est une chaîne de caractère imprimable

### 6.30. Numéro de Téléphone

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

Les valeurs dans cette syntaxe sont encodées comme si elles étaient de type chaîne de caractères imprimables. Il est recommandé dans X.520 de mettre les numéros de téléphone sous la forme internationale, comme décrit dans E.123 [15].

Exemple :
+1 512 305 0280

### 6.31. Temps UTC

```
( 1.3.6.1.4.1.1466.115.121.1.53 DESC 'UTC Time' )
```

Les valeurs dans cette syntaxe sont encodées comme si elles étaient des chaînes de caractères imprimables avec dans les chaînes une valeur de temps UTC. Ceci est historique ; les nouvelles définitions d'attribut DEVRAIENT utiliser "GeneralizedTime" à la place.

### 6.32. Description de Syntaxe de LDAP

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

Les valeurs dans cette syntaxe sont encodées selon le BNF de la section 4.3.3.

### 6.33. Description de Règle de Structure de DIT

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

Les valeurs avec cette syntaxe sont encodées selon le BNF suivant :

```
DITStructureRuleDescription = "(" whsp
    ruleidentifier whsp            ; DITStructureRule identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" whsp ]
    "FORM" woid whsp               ; NameForm
    [ "SUP" ruleidentifiers whsp ] ; superior DITStructureRules
")"

ruleidentifier = integer
ruleidentifiers = ruleidentifier |
    "(" whsp ruleidentifierlist whsp ")"
ruleidentifierlist = [ ruleidentifier *( ruleidentifier ) ]
```

## 7. Classes d'Objet

Les serveurs DEVRAIENT identifier tous les noms des classes standard de la section 7 de [12].

### 7.1. Classe d'Objet Extensible

La classe d'objet "extensibleObject", si présente dans une entrée, permet à cette entrée de détenir éventuellement un attribut. La liste d'attribut POSSIBLE de cette classe est implicitement l'ensemble de tous les attributs.

```
  ( 1.3.6.1.4.1.1466.101.120.111 NAME 'extensibleObject' SUP top AUXILIARY )
```

Il est encore exigé aux attributs obligatoires des autres classes d'objet de cette entrée d'être présents.

Notez que tous les serveurs n'auront pas à mettre en application cette classe d'objet, et ceux qui ne le feront pas devront rejeter les demandes d'ajout des entrées qui contiennent cette classe d'objet, ou de modification d'une entrée pour ajouter cette classe d'objet.

### 7.2. sous-schéma

Cette classe d'objet est utilisée dans l'entrée de sous-schéma.

```
( 2.5.20.1 NAME 'subschema' AUXILIARY
  MAY ( dITStructureRules $ nameForms $ ditContentRules $
  objectClasses $ attributeTypes $ matchingRules $
  matchingRuleUse ) )
```

L'attribut opérationnel de "ldapSyntaxes" peut également être présent dans des entrées de sous-schéma.

## 8. Règles d'Appariement

Les serveurs qui mettent en application le filtre "extensibleMatch" DEVRAIENT permettre à toutes les règles d'appariement citées dans cette section d'être utilisées dans "extensibleMatch". En général ces serveurs DEVRAIENT permettre l'utilisation des règles d'appariement avec tous les types d'attribut connus du serveur, quand la syntaxe d'affirmation de la règle d'appariement est identique à la syntaxe de valeur de l'attribut.

Les serveurs PEUVENT mettre en application des règles d'appariement supplémentaires.

### 8.1. Règles d'appariement utilisées dans des filtres d'égalité

Les serveurs DEVRAIENT être capables d'exécuter les règles d'appariement suivantes.

Pour toutes ces règles, la syntaxe d'affirmation est la même que la syntaxe de valeur.

```
( 2.5.13.0 NAME 'objectIdentifierMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

Si le client fournit un filtre utilisant un "objectIdentifierMatch" dont l'oid de "matchValue" est dans la forme "descr", et que l'oid n'est pas reconnu par le serveur, alors le filtre est non défini.

```
( 2.5.13.1 NAME 'distinguishedNameMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

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

```
( 2.5.13.8 NAME 'numericStringMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )

( 2.5.13.11 NAME 'caseIgnoreListMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )

( 2.5.13.14 NAME 'integerMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

( 2.5.13.16 NAME 'bitStringMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.6 )

( 2.5.13.20 NAME 'telephoneNumberMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )

( 2.5.13.22 NAME 'presentationAddressMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.43 )

( 2.5.13.23 NAME 'uniqueMemberMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.34 )

( 2.5.13.24 NAME 'protocolInformationMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.42 )

( 2.5.13.27 NAME 'generalizedTimeMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

( 1.3.6.1.4.1.1466.109.114.1 NAME 'caseExactIA5Match'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( 1.3.6.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

En exécutant le "caseIgnoreMatch", le "caseIgnoreListMatch", le "telephoneNumberMatch", le "caseExactIA5Match" et le "caseIgnoreIA5Match", de multiples caractères espace contigus sont traités de même qu'un seul espace, et l'espace de tête ou de queue est ignoré.

Les clients NE DOIVENT PAS supposer que les serveurs sont capables de transcrire les valeurs d'Unicode.

## 8.2. Règles d'appariement utilisées dans des filtres d'inégalité

Les serveurs DEVRAIENT être capables d'exécuter les règles d'appariement suivantes, utilisées dans des filtres "greaterOrEqual" et "lessOrEqual".

```
( 2.5.13.28 NAME 'generalizedTimeOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

( 2.5.13.3 NAME 'caseIgnoreOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

L'agencement du tri pour un "caseIgnoreOrderingMatch" dépend de l'implémentation.

## 8.3. Syntaxe et Règles d'appariement utilisées dans des sous-chaînes filtre

La syntaxe d'affirmation de sous-chaîne est utilisée seulement comme syntaxe des valeurs d'affirmation dans la règle extensible. Elle n'est pas utilisée comme syntaxe des attributs, ou dans le filtre de sous-chaîne.

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

L'affirmation de sous-chaîne est encodée selon le BNF suivant :

```
substring = [initial] any [final]
initial = value
any = "*" *(value "*")
final = value
```

La production de <valeur> est une chaîne de caractères encodée UTF-8. Si les caractères d'antislash ou d'astérisque sont présents dans une production de <valeur>, ils sont mis entre guillemets comme décrit dans la section 4.3.

Les serveurs DEVRAIENT être capables d'exécuter les règles d'appariement suivantes, qui sont utilisées dans des filtres de sous-chaîne.

```
( 2.5.13.4 NAME 'caseIgnoreSubstringsMatch'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

( 2.5.13.21 NAME 'telephoneNumberSubstringsMatch'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )

( 2.5.13.10 NAME 'numericStringSubstringsMatch'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
```

## 8.4. Règles d'appariement pour des attributs de sous-schéma

Les serveurs qui permettent à des entrées de sous-schéma d'être modifiées par les clients DOIVENT soutenir les règles d'appariement suivantes, car elles sont les règles d'appariement d'égalité pour plusieurs des attributs de sous-schéma.

```
( 2.5.13.29 NAME 'integerFirstComponentMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

( 2.5.13.30 NAME 'objectIdentifierFirstComponentMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

Les développeurs devrait noter que la syntaxe d'affirmation de ces règles d'appariement, un NOMBRE ENTIER ou un OID, est différente de la syntaxe de valeur des attributs pour lesquels c'est la règle d'appariement d'égalité.

Si le client fournit un filtre extensible utilisant un "objectIdentifierFirstComponentMatch" dont le "matchValue" est dans la forme "descr", et que l'OID n'est pas reconnu par le serveur, alors le filtre est non défini.

# 9. Considérations Sécuritaires

## 9.1. Divulgation

Les attributs des entrées d'annuaire sont employés pour fournir des informations descriptives au sujet des objets réels qu'ils représentent, qui peuvent être des gens, des organismes ou des dispositifs. La plupart des pays ont des lois sur la vie privée concernant la publication d'informations sur des personnes.

## 9.2. Utilisation des valeurs d'attribut dans des applications de sécurité

Les transformations d'une valeur "AttributeValue" de sa forme X.501 à une représentation de chaîne de caractères LDAP ne sont pas toujours réversibles vers la même forme BER ou DER. Un exemple d'une situation qui exige la forme DER d'un nom différencié est la vérification d'un certificat X.509.

Par exemple, un nom différencié se composant d'un RDN avec un AVA, dans lequel le type est "commonName" et la valeur est du choix "TeletexString" avec les lettres 'SAM' serait représenté dans LDAP comme chaîne de caractères CN=Sam. Un autre nom différencié dans lequel la valeur est toujours 'SAM' mais de choix "PrintableString" aurait la même représentation CN=Sam.

Les applications qui exigent la reconstruction de la forme DER de la valeur NE DEVRAIENT PAS utiliser la représentation en chaîne de caractères des syntaxes d'attribut quand elles convertissent une valeur en format LDAP. Au lieu de cela elles DEVRAIENT utiliser la syntaxe binaire.

# 10. Remerciements

Ce document est basé substantiellement sur RFC 1778, écrit par Tim Howes, Steve Kille, Wengyik Yeong et Colin Robbins.

Plusieurs des encodages de syntaxe d'attribut définis dans ce document et dans les documents associés sont adaptés de ceux utilisés dans le QUIPU et les réalisations IC R3 de X.500. Les contributions des auteurs de ces deux réalisations dans la spécification des syntaxes sont particulièrement reconnues.

# 11. Adresses des auteurs

Mark Wahl
Critical Angle Inc.
4815 West Braker Lane #502-385

Austin, TX 78759
USA

Phone: +1 512 372-3160
EMail: M.Wahl@critical-angle.com

Andy Coulbeck
Isode Inc.
9390 Research Blvd Suite 305
Austin, TX 78759
USA

Phone: +1 512 231-8993
EMail: A.Coulbeck@isode.com

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Rd, MS MV068
Mountain View, CA 94043
USA

Phone: +1 650 937-3419
EMail: howes@netscape.com

Steve Kille
Isode Limited
The Dome, The Square
Richmond
TW9 1DT
UK

Phone: +44-181-332-9091
EMail: S.Kille@isode.com

## 12. Bibliographie

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

[2] The Directory: Selected Attribute Types. ITU-T Recommendation X.520, 1993.

[3] The Directory: Models. ITU-T Recommendation X.501, 1993.

[4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[5] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997.

[6] Kille, S., "A String Representation for Presentation Addresses", RFC 1278, November 1991.

[7] Terminal Equipment and Protocols for Telematic Services - Standardization of Group 3 facsimile apparatus for document transmission. CCITT, Recommendation T.4.

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

[9] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

[10] Universal Multiple-Octet Coded Character Set (UCS) - Architecture and Basic Multilingual Plane, ISO/IEC 10646-1 : 1993 (With amendments).

[11] Hardcastle-Kille, S., "Mapping between X.400(1988) / ISO 10021 and RFC 822", RFC 1327, May 1992.

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

[13] Crocker, D., "Standard of the Format of ARPA-Internet Text Messages", STD 11, RFC 822, August 1982.

[14] ISO 3166, "Codes for the representation of names of countries".

[15] ITU-T Rec. E.123, Notation for national and international telephone numbers, 1988.

## 13. Copyright intégral

Les autorisations limitées accordées ci-dessus sont éternelles et ne pourront être révoquées par la Internet Society, ses successeurs ou ses repreneurs.

Ce document et les informations contenues ici sont fournis de façon " TELS QUELS " et les traducteurs, la Internet Society et la Internet Engineering Task Force déclinent toute garantie, explicites ou implicites, y compris mais pas seulement toute garantie que l'utilisation des informations de ce document ne violera pas des réglementations ou des garanties implicites commerciales ou physiques pour une application particulière.

L'édition des RFC est actuellement réalisée par l'Internet Society.