

---

# rfc4512

## Directory Information Models

**DIB** Directory Information Base  
**DUA** Directory User Agent  
**DSA** Directory System Agent  
**DSE** DSA-Specific Entry  
**DIT** Directory Information Tree  
**RDN** Relative Distinguished Name  
**AVA** Attribute Value Assertions

Le but d'un annuaire est de maintenir, et de fournir l'accès à des informations sur des objets d'intérêt. Un objet peut être tout ce qui peut être identifiable (peut être nommé). Certains attributs maintiennent des informations sur l'objet qu'ils représentent et sont appelés des attributs utilisateurs. D'autres attributs représentent des informations administratifs et/ou opérationnels et sont appelés des attributs opérationnels.

## Classes d'objet abstraites

Fournit une base de caractéristiques depuis laquelle d'autres classes d'objet peuvent hériter. Une entrée ne peut pas appartenir à une classe abstraite à moins qu'elle appartienne à une classe auxiliaire ou structurelle qui hérite de cette classe abstraite.

Une classe abstraite ne peut pas dériver d'une classe auxiliaire ou structurelle. Toutes les classes d'objet structurelles dérivent directement ou indirectement de la classe objet '**top**'. Une classe auxiliaire ne dérive pas nécessairement de **top**.

## Classes d'objet structurelles

Elles sont utilisées dans la définition de la structure des noms d'objets. Un objet ou alias est caractérisé par précisément une chaîne super-classe d'objet structurelle qui a une simple classe structurelle comme classe objet la plus subordonnée. Elle est appelée la classe objet structurelle de l'objet.

## Classes d'objet auxiliaires

Elles sont utilisées pour augmenter les caractéristiques des entrées. Elles sont communément utilisées pour augmenter le jeu d'attributs requis et permis dans une entrée. Les classes auxiliaires ne peuvent pas être des sous-classes de classes structurelles.

## Types d'attributs

Le type d'un attribut définit si il est multi-valué, la syntaxe et les règles de correspondance utilisées pour construire et comparer les valeurs de cet attribut, et d'autres fonctions. Si aucune correspondance d'égalité n'est spécifiée pour ce type d'attribut :

- 
- l'attribut ne peut pas être utilisé pour le nommage
  - 2 valeurs ne peuvent pas être équivalentes
  - les valeurs individuelles d'un attribut multi-valué ne peuvent pas être ajoutées/enlevées indépendamment
  - AVA (tels que la correspondance dans les filtres de recherche et de comparaisons) en utilisant les valeurs de cet attribut ne peuvent pas être effectuées

Sinon, la règle de correspondance d'égalité spécifiée est utilisée pour évaluer l'AVA concernant le type d'attribut. La règle d'égalité spécifiée doit être transitive et commutative.

Le type d'attribut indique si l'attribut est un attribut utilisateur ou un attribut opérationnel. S'il est opérationnel, l'attribut indique l'usage opérationnel et si l'attribut est modifiable ou non par les utilisateurs.

Un type d'attribut (un sous-type) peut dériver d'autres type d'attributs (un super-type). Les restrictions suivantes s'appliquent au sous-typage :

- un sous-type doit avoir le même usage que son super-type direct la syntaxe du sous-type doit être la même, ou une re-définition de la syntaxe du super-type
- un sous-type doit être collectif si son super-type est collectif.

Chaque type d'attribut est identifié par un Object IDentifier (OID) et optionnellement d'un ou plusieurs noms courts (descripteurs)

## Options d'attributs

Toutes les options ne peuvent pas être associées avec des attributs maintenus dans l'annuaire. le tagging options le permet.

Une description d'attribut peut être le sous-type direct de 0 ou plusieurs autres descriptions d'attributs. ces relations de sous-typage sont utilisées pour former une hiérarchie de description d'attributs et d'attributs.

Les hiérarchies d'attribut permettent un accès à la DIB avec différents degré de granularité, grâce à la valeur des attributs qui sont accédés soit en utilisant leur description d'attribut direct ou une description d'attribut plus générique.

## Alias

Un alias, ou un nom alias, pour un objet est un nom alternatif qui est fournis par l'utilisation d'entrées alias. Chaque entrée alias contient dans l'attribut **aliasedObjectName**, le nom d'un objet.

## Attributs opérationnels

Il en existe 3 sortes :

**Directory operational attributes** Est utilisé pour représenter une information opérationnelle et/ou administrative dans le Directory Information Model. Cela inclut les attributs opérationnels maintenus par le serveur (ex : createTimestamp), et les attributs opérationnels qui maintiennent des valeurs administrées par l'utilisateur (ex : ditContentRules)

**DSA-shared operational attributes** Est utilisé pour représenter des information du DSA Information Model qui est partagé entre les DSA.

**DSA-specific operational attributes** Est utilisé pour représenter des information sur le DSA Information Model qui est spécifique au DSA. ( bien que parfois peut être dérivé des informations partagées entre les DSA)

---

Les attributs opérationnel ne sont normalement pas visible. Ils ne sont pas retournés dans les résultats de recherche à moins qu'ils aient été explicitement demandés Tous les attributs opérationnels ne sont pas modifiables.

## Schéma

Le schéma de l'annuaire est un jeu de définitions et de contraintes concernant la structure du DIT, les entrées possibles, etc.

### Définition de schéma (ABNF)

```
oidlen = numericoid [ LCURLY len RCURLY ]
len = number

oids = oid / ( LPAREN WSP oidlist WSP RPAREN )
oidlist = oid *( WSP DOLLAR WSP oid )

extensions = *( SP xstring SP qdstrings )
xstring = "X" HYPHEN 1*( ALPHA / HYPHEN / USCORE )

qdescrs = qdescr / ( LPAREN WSP qdescrlist WSP RPAREN )
qdescrlist = [ qdescr *( SP qdescr ) ]
qdescr = SQUOTE descr SQUOTE

qdstrings = qdstring / ( LPAREN WSP qdstringlist WSP RPAREN )
qdstringlist = [ qdstring *( SP qdstring ) ]
qdstring = SQUOTE dstring SQUOTE
dstring = 1*( QS / QQ / QUTF8 ) ; escaped UTF-8 string

QQ = ESC %x32 %x37 ; "\27"
QS = ESC %x35 ( %x43 / %x63 ) ; "\5C" / "\5c"

; Any UTF-8 encoded Unicode character
; except %x27 ("\27") and %x5C ("\5C")
QUTF8 = QUTF1 / UTFMB

; Any ASCII character except %x27 ("\27") and %x5C ("\5C")
QUTF1 = %x00-26 / %x28-5B / %x5D-7F
```

Le champ **NAME** fournis un jeu de noms court qui sont utilisés comme alias pour l'OID

Le champs **DESC** est optionnel et apporte un description

Le champ **OBSOLETE**, si présent, indique que l'élément n'est pas actif

## Définition de classe d'objet

```
ObjectClassDescription = LPAREN WSP
numericoid ; object identifier
[ SP "NAME" SP qdescrs ] ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ] ; not active
[ SP "SUP" SP oids ] ; superior object classes
[ SP kind ] ; kind of class
[ SP "MUST" SP oids ] ; attribute types
[ SP "MAY" SP oids ] ; attribute types
extensions WSP RPAREN
```

---

kind = "ABSTRACT" / "STRUCTURAL" / "AUXILIARY"

**<numericoid>** est l'OID assigné à cette classe d'objet  
**NAME <qdescrs>** Sont des noms courts identifiant cette classe d'objet  
**DESC <qdstring>** est une courte description  
**OBSOLETE** indique que cette classe d'objet n'est pas active  
**SUP <oids>** spécifie les super-classes directe de cette classe d'objet.  
**kind** Spécifie le type de classe (ABSTRACT, STRUCTURAL ou AUXILIARY)  
**MUST et MAY** spécifient les jeux d'attributs requis et permis  
**<extensions>** décrit les extensions

## Définition des attributs

```
AttributeTypeDescription = LPAREN WSP
numericoid ; object identifier
[ SP "NAME" SP qdescrs ] ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ] ; not active
[ SP "SUP" SP oid ] ; supertype
[ SP "EQUALITY" SP oid ] ; equality matching rule
[ SP "ORDERING" SP oid ] ; ordering matching rule
[ SP "SUBSTR" SP oid ] ; substrings matching rule
[ SP "SYNTAX" SP noidlen ] ; value syntax
[ SP "SINGLE-VALUE" ] ; single-value
[ SP "COLLECTIVE" ] ; collective
[ SP "NO-USER-MODIFICATION" ] ; not user modifiable
[ SP "USAGE" SP usage ] ; usage
extensions WSP RPAREN ; extensions
```

```
usage = "userApplications" / ; user
"directoryOperation" / ; directory operational
"distributedOperation" / ; DSA-shared operational
"dSAOperation" ; DSA-specific operational
```

**<numericoid>** est l'OID assigné à ce type d'attribut  
**NAME <qdescrs>** sont des noms courts identifiant ce type d'attribut  
**DESC <qdstring>** est une courte description  
**OBSOLETE** Indique que cet attribut n'est pas actif  
**SUP oid** Spécifie le supertype direct de ce type  
**EQUALITY, ORDERING, et SUBSTR** Fournissent l'OID de l'égalité, l'ordonnancement et les règle de correspondance de chaîne, respectivement.  
**SYNTAX** Identifie La syntaxe de la valeur par OID et peut suggérer une limite minimum  
**SINGLE-VALUE** Indique que les attributs de ce type sont restreint à une seule valeur  
**COLLECTIVE** Indique que cet attribut est collectif  
**NO-USER-MODIFICATION** Indique que ce type d'attribut n'est pas modifiable par l'utilisateur  
**USAGE** Indique l'application de ce type d'attribut  
**<extensions>** décrit les extensions  
Chaque description de type d'attribut doit avoir au moins **SUP** ou **SYNTAX**.

---

Si **SUP** est fournis, **SYNTAX**, **EQUALITY**, **ORDERING** et **SUBSTRING** sont pris du super-type s'ils ne sont pas spécifié **COLLECTIVE** nécessite un usage userApplications  
**NO-USER-MODIFICATION** nécessite un usage opérationnel  
**AttributeTypeDescription** ne liste pas les règles de correspondance utilisées avec ce type d'attribut dans un filtre de recherche **extensibleMatch**. C'est fait avec l'attribut **matchingRuleUse**.

## Règles de correspondance

Elles sont utilisée pour les opérations AVA tels que les opérations de comparaison ou les filtres de recherche, déterminant quelles valeurs individuelles sont ajoutée ou supprimée durant les opération de modification, et en comparant des DN.

```
MatchingRuleDescription = LPAREN WSP
numericoid ; object identifier
[ SP "NAME" SP qdescrs ] ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ] ; not active
SP "SYNTAX" SP numericoid ; assertion syntax
extensions WSP RPAREN ; extensions
```

**<numericoid>** est l'OID de la règle de correspondance  
**NAME <qdescrs>** sont des noms cours décrivant la règle  
**DESC <qdstring>** est une courte description  
**OBSOLETE** Indique que cette règle n'est pas active  
**SYNTAX** Identifie la syntaxe d'assertion par OID  
**<extensions>** Décris les extensions

## Utilisation des règles de correspondance

Une règle de correspondance liste les types d'attributs qui sont utilisables avec un filtre de recherche **extensibleMatch**

```
MatchingRuleUseDescription = LPAREN WSP
numericoid ; object identifier
[ SP "NAME" SP qdescrs ] ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ] ; not active
SP "APPLIES" SP oids ; attribute types
extensions WSP RPAREN ; extensions
```

**<numericoid>** est l'OID de la règle associée avec cette utilisation de règle  
**NAME <qdescrs>** Sont des noms cours identifiant l'utilisation de règle  
**DESC <qdstring>** est une courte description  
**OBSOLETE** indique que cette utilisation de règle de correspondance est désactivée  
**APPLIES** Fournis une liste de type d'attributs auquel la règle s'applique  
**<extensions>** décrit les extensions

## Syntaxes LDAP

---

Les syntaxes LDAP de valeurs d'attribut et d'assertion sont décrites en ASN.1 et optionnellement, ont un encodage de chaîne (LDAP-specific encoding), généralement en UTF-8.

```
SyntaxDescription = LPAREN WSP
  numericoid ; object identifier
  [ SP "DESC" SP qdstring ] ; description
  extensions WSP RPAREN ; extensions
```

**<numericoid>** est l'OID assignée à cette syntaxe LDAP

**DESC <qdstring>** est une courte description

**<extensions>** Décris les extensions

## Règles de contenu DIT

Une **DIT content rule** est une règle régissant le contenu des entrées d'une classe d'objet structurelle particulière. Elle définissent quelles classes d'objets auxiliaires sont requis, permis ou non permises dans l'entrée.

```
DITContentRuleDescription = LPAREN WSP
  numericoid ; object identifier
  [ SP "NAME" SP qdesrcs ] ; short names (descriptors)
  [ SP "DESC" SP qdstring ] ; description
  [ SP "OBSOLETE" ] ; not active
  [ SP "AUX" SP oids ] ; auxiliary object classes
  [ SP "MUST" SP oids ] ; attribute types
  [ SP "MAY" SP oids ] ; attribute types
  [ SP "NOT" SP oids ] ; attribute types
  extensions WSP RPAREN ; extensions
```

**<numericoid>** est l'OID de la classe structurelle associée avec ces DIT content rule

**NAME <qdesrcs>** Sont des noms courts identifiant ce DIT content rule

**DESC <qdstring>** est une courte description

**OBSOLETE** Indique que ce DIT content rule est inactif

**AUX** Spécifie une liste de classes auxiliaires que les entrée sujettes à ce DIT content rules peuvent appartenir.

**MUST, MAY, et NOT** Spécifient les listes de type d'attributs qui sont requis, permis ou non-permis

**<extensions>** décrit les extensions

## Règles de structure DIT

Une règle de structure DIT est une règle régissant la structure du DIT en spécifiant un supérieur autorisé à subordonner la relation d'entrée. une règle de structure concerne une forme de nom, et donc une classe structurelle, de structurer les règles supérieurs. Cela permet au entrées de la classe structurelle identifiées par la forme de nom d'exister dans le DIT comme subordonnée aux entrée régies par la règle structurelle supérieur.

```
DITStructureRuleDescription = LPAREN WSP
  ruleid ; rule identifier
  [ SP "NAME" SP qdesrcs ] ; short names (descriptors)
  [ SP "DESC" SP qdstring ] ; description
  [ SP "OBSOLETE" ] ; not active
  SP "FORM" SP oid ; NameForm
```

```
[ SP "SUP" ruleids ] ; superior rules
extensions WSP RPAREN ; extensions

ruleids = ruleid / ( LPAREN WSP ruleidlist WSP RPAREN )
ruleidlist = ruleid *( SP ruleid )
ruleid = number
```

**<ruleid>** est identifiant de règle de ce DIT structure rule  
**NAME <qdescrs>** sont des noms identifiant ce DIT structure rule  
**DESC <qdstring>** est une courte description  
**OBSOLETE** indique que ce DIT structure rule use n'est pas actif  
**FORM** spécifie le name form associé à ce DIT structure rule  
**SUP** identifie les règles supérieurs (par rule id)  
**<extensions>** Décris les extensions

## Formes de nom

un **name form** spécifie un **RDN** permis pour les entrées d'une classe d'objet structurale particulière. il identifie une classe nommée et un ou plusieurs types d'attributs pour le nommage (ex : le RDN). ce sont des pièces primitives de spécification utilisées dans la définition de règles de structures de DIT. Chaque **name form** indique la classe structurale à nommer, un jeu de types d'attributs requis, et un autre pour les types permis.

```
NameFormDescription = LPAREN WSP
numericoid ; object identifier
[ SP "NAME" SP qdescrs ] ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ] ; not active
SP "OC" SP oid ; structural object class
SP "MUST" SP oids ; attribute types
[ SP "MAY" SP oids ] ; attribute types
extensions WSP RPAREN ; extensions
```

**<numericoid>** est l'OID du name form  
**NAME <qdescrs>** sont des noms identifiant ce name form  
**DESC <qdstring>** est une courte description  
**OBSOLETE** indique que ce name form est inactif  
**OC** identifie la classe structurale à laquelle cette règle s'applique  
**MUST et MAY** Spécifient les jeux d'attributs nommés requis et permis pour ce name form  
**<extensions>** Décris les extensions

## Sous-entrées de sous-schéma

Elles sont utilisée pour administrer les informations sur le schéma de l'annuaire. Une sous-entrée de sous-schéma contient toutes les définitions de schéma utilisées par les entrées dans une partie de l'arborescence. Les serveurs peuvent autoriser la modification du sous-schéma.

## Découverte du sous-schéma

---

Pour découvrir le DN du sous-schéma qui maintient le sous-schéma contrôlant une entrée particulière, un client lit l'attribut opérationnel **subschemaSubentry** de cet entrée. Pour lire cet attribut, le client doit fournir une opération de recherche où le **baseObject** est le DN de l'entrée de sous-schéma. Le **scope** est **baseObject**, le filtre est "(objectClass=subschema)" et le champs attributs liste les attributs désirés.

## DSA Information Model

Le protocole LDAP assume qu'il y'a un ou plusieurs serveurs qui fournissent conjointement accès au DIT. Le serveur maintenant l'information originale est appelée le master, ceux qui maintiennent une copie, des serveurs shadowing ou caching.

**context prefix** La séquence de RDN partant de la racine du DIT jusqu'au vertex initial d'un contexte de nommage, correspondant au DN de ce vertex

**naming context** Une sous-arborescence d'entrées maintenues dans un DSA master

Un contexte de nommages est la plus grande collection d'entrées, commençant à une entrée master, et incluant tous ses subordonnés et leur subordonnés, jusqu'aux entrées qui sont master sur d'autres serveurs. Le Root d'un DIT est un DSE et ne fait pas partit d'un contexte de nommage (ou d'une sous-arborescence). Chaque serveur a des valeurs d'attribut différents dans le root DSE.

## Server-Specific Data Requirements

Un serveur LDAP devrait fournir des informations sur lui-même et d'autres informations qui sont spécifiques à d'autres serveurs. C'est représenté sous la forme d'un groupe d'attributs dans le root DSE, qui sont nommé avec un DN sans RDN.

## Cache et Shadowing

Certains serveurs maintiennent des copies des entrées, qui peuvent être utilisée pour répondre à des requêtes ou des comparaisons, mais vont retourner des referrals ou contacter d'autres serveur si des opérations de modification sont demandées. Ces serveurs doivent s'assurer qu'ils ne violent pas de contrôle d'accès.

## Conduite des serveurs

- Les serveurs doivent reconnaître tous les noms des attributs et classes définis dans ce document, mais n'ont pas besoin de supporter les fonctionnalité associés (sauf mentionné).
- Les serveurs doivent s'assurer que les entrées sont conformes au règles de schéma et autres contraintes
- Les serveurs peuvent supporter : DIT Content Rules, DIT Structure Rules, Name Forms, alias, la classe object extensibleObject, les sous-entrées.
- Les serveurs peuvent implémenter des éléments de schéma additionnel.

## Considération IANA

Les descripteurs suivant réfèrent à cette RFC :

NAME Type OID



---

```
alias O 2.5.6.1
aliasedObjectName A 2.5.4.1
altServer A 1.3.6.1.4.1.1466.101.120.6
attributeTypes A 2.5.21.5
createTimestamp A 2.5.18.1
creatorsName A 2.5.18.3
dITContentRules A 2.5.21.2
dITStructureRules A 2.5.21.1
extensibleObject O 1.3.6.1.4.1.1466.101.120.111
ldapSyntaxes A 1.3.6.1.4.1.1466.101.120.16
matchingRuleUse A 2.5.21.8
matchingRules A 2.5.21.4
modifiersName A 2.5.18.4
modifyTimestamp A 2.5.18.2
nameForms A 2.5.21.7
namingContexts A 1.3.6.1.4.1.1466.101.120.5
objectClass A 2.5.4.0
objectClasses A 2.5.21.6
subschema O 2.5.20.1
subschemaSubentry A 2.5.18.10
supportedControl A 1.3.6.1.4.1.1466.101.120.13
supportedExtension A 1.3.6.1.4.1.1466.101.120.7
supportedFeatures A 1.3.6.1.4.1.4203.1.3.5
supportedLDAPVersion A 1.3.6.1.4.1.1466.101.120.15
supportedSASLMechanisms A 1.3.6.1.4.1.1466.101.120.14
top O 2.5.6.0
```

## Description des attributs

**aliasedObjectName** Le DN de l'objet pointé

**objectClass** Chaque entrée dans le DIT a un attribut objectClass

**creatorsName** Créateur de l'entrée dans le DIT

**createTimestamp** Date et heure de la création de l'entrée

**modifiersName** Spécifie le DN du dernier modificateur de l'objet

**modifyTimestamp** Date et heure de la dernière modification de l'objet

**structuralObjectClass** Indique la classe objet structurelle de l'entrée

**governingStructureRule** Indique la règle gouvernante de l'entrée

**subschemaSubentry** Permet aux clients de découvrir les attributs et classes d'objet permis et présents dans l'arborescence

**objectClasses** Maintient les définitions des classes d'objet

**attributeTypes** Maintient les définitions des types d'attributs

**matchingRules** Maintient les définitions des règles de correspondance

**matchingRuleUse** Maintient les définitions des utilisations de règle de correspondance

**ldapSyntaxes** Maintient les définitions des syntaxes LDAP

**dITContentRules** Maintient les définitions des règles de contenu de DIT

**dITStructureRules** Maintient les définitions des règles de structure de DIT

**nameForms** Liste les Names Form qui sont forcés

**altServer** (Root DSE) Serveurs alternatifs

**namingContexts** (Root DSE) Contexte de nommage

**supportedControl** (Root DSE) Contrôles LDAP reconnus

---

**supportedExtension** (Root DSE) Opération étendues reconnus  
**supportedFeatures** (Root DSE) fonctionnalités LDAP reconnus  
**supportedLDAPVersion** (Root DSE) Versions LDAP supportées  
**supportedSASLMechanisms** (Root DSE) Mécanismes SASL reconnus

## Description des classes d'objet

**alias** Un objet alias  
**subschema** Le sous-schéma est maintenu en sous-entrées appartenant à cette classe  
**extensibleObject** Permet à des objets de maintenir des attributs utilisateur

## Définition des attributs

```
attributeType ( 2.5.4.1 NAME 'aliasedObjectName'  
  EQUALITY distinguishedNameMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12  
  SINGLE-VALUE )
```

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

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

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

```
attributeType ( 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 )
```

```
attributeType ( 2.5.4.0 NAME 'objectClass'  
  EQUALITY objectIdentifierMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

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

```
attributeType ( 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 )
```

---

```
attributeType ( 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 )

attributeType ( 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 )

attributeType ( 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 )

attributeType ( 2.5.21.9 NAME 'structuralObjectClass'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
  SINGLE-VALUE NO-USER-MODIFICATION
  USAGE directoryOperation )

attributeType ( 2.5.21.10 NAME 'governingStructureRule'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE NO-USER-MODIFICATION
  USAGE directoryOperation )

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

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

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

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

attributeType ( 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 )

attributeType ( 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 )

attributeType ( 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 )

attributeType ( 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 )

attributeType ( 1.3.6.1.4.1.4203.1.3.5 NAME 'supportedFeatures'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
  USAGE dSAOperation )

attributeType ( 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 )

attributeType ( 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 )
```

## Définition des classes d'objet

```
objectClass ( 2.5.6.1 NAME 'alias'
  SUP top STRUCTURAL
  MUST aliasedObjectName )

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

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