

LDAP : Une présentation rapide

Table des matières

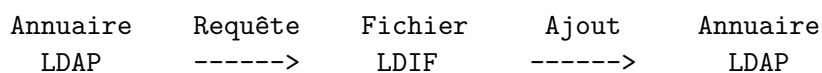
1	Présentation de LDAP	1
2	Caractéristiques de LDAP	1
3	Utilisation de LDAP	2
4	Structure de LDAP	2
5	L'attribut particulier <code>dn</code>	2
6	L'attribut particulier <code>objectclass</code>	2
7	Les attributs classiques	3
8	Le langage de requête	3
9	Création d'attributs et de classes	4
9.1	Définition d'un attribut	4
9.2	Définition d'une classe	4

1 Présentation de LDAP

- LDAP : *Lightweight Directory Access Protocol*.
 - C'est un protocole d'accès à un annuaire.
 - Il n'y a donc pas de contrainte d'implantation.
 - Un annuaire propriétaire peut (et doit) fournir une interface LDAP (comme ODBC pour les S.G.B.D.R.).
 - Il existe des annuaires LDAP natifs (openLDAP).

2 Caractéristiques de LDAP

- Caractéristiques :
 - Accès rapide mais mise à jour plus lente.
 - Structure arborescente.
 - Langage de recherche.
 - l'échange de données se fait par LDIF.



- Il existe des opérations de duplication et de synchronisation.
 - il est facile de maintenir des copies.

3 Utilisation de LDAP

- Quelques exemples :
 - Annuaire d'entreprise,
 - Annuaire *Active Directory* (*Windows 2000*).
 - Alternative à NIS et NIS+.
 - Représentation des paramètres du Système d'Information (J2EE).
 - Annuaire de l'université (<http://annuaire.univmed.fr>).

4 Structure de LDAP

- Un annuaire LDAP est un arbre (DIT pour *Directory Information Tree*).
- Chaque noeud correspond à une série d'affectations d'attributs.
 - On associe une ou plusieurs valeurs à chaque attribut.
 - Les valeurs sont des chaînes de caractères ou des données binaires.
 - Exemples :

```
dn: dc=my-domain,dc=com
objectclass: dcObject
objectclass: organization
o: Ma première organisation
dc: my-domain
```

```
dn: cn=massat,dc=my-domain,dc=com
objectclass: person
cn: massat
sn: Jean-Luc Massat
description: Enseignant au DIL
```

5 L'attribut particulier dn

- « dn » pour *distinguished name*.
- Le DN est l'identifiant d'une entrée LDAP. C'est un chemin dans l'arborescence de l'annuaire.

```
dc=youpi,dc=com
| dc=magasin,dc=youpi,dc=com
| | user=jean,dc=magasin,dc=youpi,dc=com
| | user=bob,dc=magasin,dc=youpi,dc=com
| | ...
| dc=usine,dc=youpi,dc=com
| | user=jean,dc=usine,dc=youpi,dc=com
| | user=marcel,dc=usine,dc=youpi,dc=com
| | ..
```

6 L'attribut particulier objectclass

- L'attribut « objectclass » désigne la ou les classes associées au noeud.
- Une *classe* définit les attributs obligatoires et optionnels d'un noeud.

- Les attributs et les classes sont définis dans des *schémas* LDAP.
- Dans un serveur LDAP il existe un jeu de schémas de base qui définissent des classes et des attributs.

```
dn: dc=my-domain,dc=com
objectclass: dcObject
objectclass: organization
...
```

```
dn: cn=massat,dc=my-domain,dc=com
objectclass: person
..
```

7 Les attributs classiques

- « **cn** » Le *common name* ou nom commun.
- « **gn** » *given name* c'est à dire le prénom.
- « **sn** » *surname*.
- « **l** » Le *locality name*.
- « **st** » *state or province name*.
- « **ou** » *organisational unit*.
- « **dc** » *domain component*.
- « **o** » *organization name*.

8 Le langage de requête

- Les tests élémentaires de la valeur d'un attribut :

```
attribut = valeur    égalité
attribut ~= valeur   approximation
attribut != valeur   différence
attribut >= valeur   supérieur
attribut <= valeur   inférieur
```

- La valeur peut être une chaîne combinée au caractère joker * et au symbole de neutralisation \c. Exemples :

```
(cn = user1*)   cn débute par « user1 »
(cn = *)        cn a une valeur
```

- Les conditions élémentaires peuvent être composées :

```
(| cond1 cond2)   condition 1 ou condition 2
(& cond1 cond2)   condition 1 et condition 2
(! cond1)         négation de la condition 1
```

- Un exemple plus réaliste :

```
(!(sn = A*) (& (cn = user1)(dc = dil*)))
```

9 Création d'attributs et de classes

Si les classes et les attributs prédéfinis ne suffisent pas, il est possible de :

- définir de nouveaux attributs,
 - à partir de rien,
 - à partir d'autres attributs.
- définir de nouvelles classes
 - à partir de rien,
 - à partir de classes existantes.

9.1 Définition d'un attribut

Définition d'un attribut dans un schéma LDAP :

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

attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
  DESC 'RFC2256: last (family) name(s) for which the entity is known by'
  SUP name )

attributetype ( 2.5.4.6 NAME ( 'c' 'countryName' )
  DESC 'RFC2256: ISO-3166 country 2-letter code'
  SUP name SINGLE-VALUE )

attributetype ( 2.5.4.10 NAME ( 'o' 'organizationName' )
  DESC 'RFC2256: organization this object belongs to'
  SUP name )
```

9.2 Définition d'une classe

Définition d'une classe dans un schéma LDAP :

```
objectclass ( 2.5.6.6 NAME 'person'
  DESC 'RFC2256: a person'
  SUP top STRUCTURAL
  MUST ( sn $ cn )
  MAY ( userPassword $ telephoneNumber $
    seeAlso $ description ) )

objectclass ( 2.5.6.2 NAME 'country'
  DESC 'RFC2256: a country'
  SUP top STRUCTURAL
  MUST c
  MAY ( searchGuide $ description ) )

objectclass ( 1.3.6.1.4.1.1466.344 NAME 'dcObject'
  DESC 'RFC2247: domain component object'
  SUP top AUXILIARY MUST dc )
```

Un exemple plus réaliste :

```
objectclass ( 2.5.6.4 NAME 'organization'
  DESC 'RFC2256: an organization'
  SUP top STRUCTURAL
  MUST o
  MAY ( userPassword $ searchGuide $ seeAlso $
    businessCategory $ x121Address $
    registeredAddress $ destinationIndicator $
    preferredDeliveryMethod $ telexNumber $
    teletexTerminalIdentifier $
    telephoneNumber $ internationaliSDNNumber $
    facsimileTelephoneNumber $ street $ postOfficeBox $
    postalCode $
    postalAddress $ physicalDeliveryOfficeName $ st $ l $
    description ) )
```