
rfc3672

Subentries dans LDAP

Dans les annuaires X500, les sous-entrées sont des entrées spéciales utilisées pour maintenir des informations associées avec une sous-arborescence. C'est un type d'entrée spécial subordonné à un point administratif. Il contient des attributs qui sont pertinents pour une sous-arborescence. Les sous-entrées et leur point administratif font partie du même contexte de nommage. Une simple sous-entrée peut servir de nombreux aspects de l'autorité administrative.

Subtree Specification Syntax

Fournis un mécanisme général pour la spécification de jeu d'entrées dans une sous-arborescence du DIT. une sous-arborescence commence à une entrée de base et inclus tous les subordonnés. une subtree specification est toujours utilisée dans un contexte ou un scope qui détermine implicitement les limites de la sous-arborescence. Cette syntaxe correspond au type **SubtreeSpecification (X501)** :

```
SubtreeSpecification ::= SEQUENCE {
    base [0] LocalName DEFAULT { },
    COMPONENTS OF ChopSpecification,
    specificationFilter [4] Refinement OPTIONAL }
```

```
LocalName ::= RDNSequence
```

```
ChopSpecification ::= SEQUENCE {
    specificExclusions [1] SET OF CHOICE {
    chopBefore [0] LocalName,
    chopAfter [1] LocalName } OPTIONAL,
    minimum [2] BaseDistance DEFAULT 0,
    maximum [3] BaseDistance OPTIONAL }
```

```
BaseDistance ::= INTEGER (0 .. MAX)
```

```
Refinement ::= CHOICE {
    item [0] OBJECT-CLASS.&id,
    and [1] SET OF Refinement,
    or [2] SET OF Refinement,
    not [3] Refinement }
```

La syntaxe LDAP est :

(**1.3.6.1.4.1.1466.115.121.1.45 DESC 'SubtreeSpecification'**)

base Le composant **base** nomme l'entrée de base de la sous-arborescence. cette entrée peut être une sequence de RDN relatif à l'entrée root, ou vide si la base est le root lui-même.

Specific Exclusion Liste d'exclusion qui spécifie les entrées et leur subordonnés à exclure de l'arborescence. chaque entrée est une sequence de RDN relatif à l'entrée de base. Chaque exclusion est de forme **chopBefore** ou **chopAfter**. chopBefore exclue l'entrée et ses subordonnés, chopAfter exclue seulement les subordonnés.

Minimum est Maximum Permettent l'exclusion d'entrées basé sur leur profondeur dans le DIT

Specification Filter Expression booléenne sur l'assertion des valeurs de l'attribut objectClass de l'entrée de base et ses subordonnés. un élément d'assertion est évalué à TRUE pour une entrée si l'attribut objectClass de l'entrée contient l'OID nommé.

Types d'attributs de rôle administratif

L'attribut opérationnel **administrativeRole** est spécifié comme suit :

```
( 2.5.18.5 NAME 'administrativeRole'  
EQUALITY objectIdentifierMatch  
USAGE directoryOperation  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

Les valeurs possible pour cette attribut sont définis dans X.501 :

```
2.5.23.1 autonomousArea  
2.5.23.2 accessControlSpecificArea  
2.5.23.3 accessControlInnerArea  
2.5.23.4 subschemaAdminSpecificArea  
2.5.23.5 collectiveAttributeSpecificArea  
2.5.23.6 collectiveAttributeInnerArea
```

L'attribut opérationnel **administrativeRole** est aussi utilisé pour réguler les sous-entrées qui ont le droit d'être subordonné à une entrée administrative. Une sous-entrée qui n'est pas d'une classe permise ne peut pas être subordonnée à cette entrée administrative.

Type d'attribut de spécification de sous-arborescence.

```
( 2.5.18.6 NAME 'subtreeSpecification'  
SINGLE-VALUE  
USAGE directoryOperation  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.45 )
```

Cet attribut est présent dans toutes les sous-entrées. nomme les collections d'entrées dans le DIT pour un ou plusieurs aspects de l'autorité administrative.

Classe d'objet **subentry**

```
( 2.5.17.0 NAME 'subentry'  
SUP top STRUCTURAL  
MUST ( cn $ subtreeSpecification ) )
```

Contrôle des sous-entrées

Le contrôle des **subentry** peut être envoyé avec un `searchRequest` pour contrôler la visibilité des entrées qui sont dans le scope. Le contrôle est un contrôle LDAP dont **controlType** est **1.3.6.1.4.1.4203.1.10.1**, et **controlValue** contient un booleen encodé BER indiquant la visibilité. Noter que la visibilité à true vaut { **01 01 FF** } et FALSE vaut { **01 01 00** }.

Considérations de sécurité

Les **subentry** maintiennent des informations administratives ou autre informations sensibles et devraient être protégés des accès non-autorisés.

Descripteurs

NAME Type OID

--- ----

```
accessControlInnerArea R 2.5.23.3  
accessControlSpecificArea R 2.5.23.2  
administrativeRole A 2.5.18.5
```

autonomousArea R 2.5.23.1
collectiveAttributeInnerArea R 2.5.23.6
collectiveAttributeSpecificArea R 2.5.23.5
subentry O 2.5.17.0
subschemaAdminSpecificArea R 2.5.23.4
subtreeSpecification A 2.5.18.6

A : Attribute

O : objectClass

R : Administrative Role

Subtree spécification ABNF :

```
SubtreeSpecification = "{" [ sp ss-base ]  
  [ sep sp ss-specificExclusions ]  
  [ sep sp ss-minimum ]  
  [ sep sp ss-maximum ]  
  [ sep sp ss-specificationFilter ]  
  sp "}"
```

```
ss-base = id-base msp LocalName  
ss-specificExclusions = id-specificExclusions msp  
SpecificExclusions  
ss-minimum = id-minimum msp BaseDistance  
ss-maximum = id-maximum msp BaseDistance  
ss-specificationFilter = id-specificationFilter msp Refinement
```

```
id-base = %x62.61.73.65 ; "base"  
id-specificExclusions = %x73.70.65.63.69.66.69.63.45.78.63.6C.75.73  
%x69.6F.6E.73 ; "specificExclusions"  
id-minimum = %x6D.69.6E.69.6D.75.6D ; "minimum"  
id-maximum = %x6D.61.78.69.6D.75.6D ; "maximum"  
id-specificationFilter = %x73.70.65.63.69.66.69.63.61.74.69.6F.6E.46  
%x69.6C.74.65.72 ; "specificationFilter"
```

```
SpecificExclusions = "{" [ sp SpecificExclusion  
  ( "," sp SpecificExclusion ) ] sp "}"  
SpecificExclusion = chopBefore / chopAfter  
chopBefore = id-chopBefore " : " LocalName  
chopAfter = id-chopAfter " : " LocalName  
id-chopBefore = %x63.68.6F.70.42.65.66.6F.72.65 ; "chopBefore"  
id-chopAfter = %x63.68.6F.70.41.66.74.65.72 ; "chopAfter"  
Refinement = item / and / or / not  
item = id-item " : " OBJECT-IDENTIFIER  
and = id-and " : " Refinements  
or = id-or " : " Refinements  
not = id-not " : " Refinement  
Refinements = "{" [ sp Refinement  
  ( "," sp Refinement ) ] sp "}"  
id-item = %x69.74.65.6D ; "item"  
id-and = %x61.6E.64 ; "and"  
id-or = %x6F.72 ; "or"  
id-not = %x6E.6F.74 ; "not"
```

```
BaseDistance = INTEGER-0-MAX
```