
rfc3296

Named Subordinate References

objectClass

Détail les éléments du schéma et du protocole pour représenter et gérer les références subordonnées nommées dans LDAP. un objet référant maintient une ou plusieurs URI contenus dans des valeurs de type d'attributs de référence et sont utilisé pour générer des références. Un contrôle, **ManageDsaIT**, est définis pour permettre de manipuler ces référants et autres objets spéciaux comme des objets spéciaux.

ObjectClass referral

Cette classe d'objet devrait être utilisé en conjonction avec la classe d'objet extensibleObject pour supporter les attributs de nommage utilisé dans le DN de l'entrée. Les objets référant sont normalement instanciés aux DSE immédiatement subordonnés au entrées dans un contexte de nommage maintenu par le DSA.

(**2.16.840.1.113730.3.2.6** NAME 'referral' DESC 'named subordinate reference object' STRUCTURAL MUST ref)

Type d'attribut ref

Ce type d'attribut a une syntaxe directoryString sensible à la casse et multi-valué. Les valeurs placées dans cet attribut doivent être conforme à la spécification donnée pour l'attribut labeledURI. Si cet attribut réfère à un serveur LDAP, il doit être sous la forme d'URL LDAP. Cette URL LDAP ne devrait pas contenir de scope, filtre, liste d'attribut, ou extension et devrait contenir un DN on-vide.

(**2.16.840.1.113730.3.1.34** NAME 'ref' DESC 'named reference - a labeledURI' EQUALITY caseExactMatch SYNTAX **1.3.6.1.4.1.1466.115.121.1.15** USAGE distributedOperation)

Contrôle ManageDsaIT

Un client peut fournir le contrôle **ManageDsaIT** avec une opération pour indiquer que l'opération est de gérer des objets dans le DIT. Ce contrôle indique que les entrée spécifique sont traitées comme des entrées normales.

Le **controlType** vaut **2.16.840.1.113730.3.4.2** et **controlValue** est absent.

Références des subordonnés nommés

Une référence subordonnée nommée est construite en instanciant un objet referral dans le serveur dé-référençant avec les les valeurs de l'attribut ref. qui pointent vers le subtree correspondant maintenu dans le serveur référencié. En général, le nom de l'objet referral est le même que l'objet référencié et cet objet référencié est un préfixe de contexte.

Si le serveur A maintient **DC=example,dc=net** et le serveur B maintient **dc=sub,dc=example,dc=net**, le serveur A peut contenir un objet referral nommé **dc=sub,dc=example,dc=net** qui contient un attribut ref avec la valeur **ldap://B/dc=sub,dc=example,dc=net**

```
dn: DC=sub,DC=example,DC=net
dc: sub
ref: ldap://B/DC=sub,DC=example,DC=net
objectClass: referral
objectClass: extensibleObject
```

Typiquement, le DN de l'objet referral et le DN de l'objet sur le serveur référencé sont les même. Si l'attribut ref a plusieurs valeurs, tous les DN contenus dans les URL LDAP devraient être équivalent. Les administrateurs devraient éviter de configurer des boucles en utilisant des référants. Les références nommées doivent être traitées comme des entrées normales si la requête inclus le contrôle manageDsaIT.

scénario

Les sections suivante contiennent des spécifications sur l'utilisation des referral dans différent scénarios. Il est à noter que, dans ce document, une opération de recherche est conceptuellement divisée en 2 phases distincts : 1. trouver l'objet de base où la recherche commence, et 2. effectuer la recherche.

Exemple de configuration

Par exemple, supposons que le serveur contacté (hosta) maintient l'entrée "O=MNN,C=WW" et l'entrée "CN=manager,O=MNN,C=WW" et les objets referral suivants :

```
dn: OU=People,O=MNN,C=WW
ou: People
ref: ldap://hostb/OU=People,O=MNN,C=US
ref: ldap://hostc/OU=People,O=MNN,C=US
objectClass: referral
objectClass: extensibleObject
dn: OU=Roles,O=MNN,C=WW
ou: Roles
ref: ldap://hostd/OU=Roles,O=MNN,C=WW
objectClass: referral
objectClass: extensibleObject
```

Le premier referral indique que **hostb** et **hostc** maintiennent "OU=People,O=MNN,C=WW". Le second indique que **hostd** maintient "OU=Roles,O=MNN,C=WW".

Considérations de l'objet cible

Cette section détail la manipulation des referral pour les opérations add, compare, delete, modify et modrdn. Si le client demande une de ces opérations, il y'a 4 cas que le serveur doit gérer.

cas 1 : l'objet cible n'est pas manipulé par le serveur et n'est pas dans ou subordonné à un contexte de nommage ni subordonné à un referral maintenu par le serveur.

Le serveur devrait traiter la requête normalement comme approprié pour une base non-existante qui n'est pas dans un contexte de nommage du serveur. (retourne généralement noSuchObject ou un referral).

cas 2 : L'objet cible est maintenu par le serveur et est un objet referral.

Le serveur devrait retourner l'URI contenu dans ref, modifié comme suit.

exemple : Si le client requête une opération modify pour l'objet cible "OU=People,O=MNN,C=WW", le serveur va retourner :

```
ModifyResponse (referral) {  
  ldap ://hostb/OU=People,O=MNN,C=WW  
  ldap ://hostc/OU=People,O=MNN,C=WW  
}
```

cas 3 : L'objet cible n'est pas maintenu par le serveur, mais le contexte de nommage le plus proche ne contient pas de referral.

Si le contexte de nommage le plus proche ne contient pas de referral auquel l'objet cible est subordonné, le serveur devrait traiter la requête de manière appropriée (généralement retourne noSuchObject)

cas 4 : L'objet cible n'est pas maintenu par le serveur, mais le contexte de nommage le plus proche contient un referral.

Si le client demande une opération pour laquelle l'objet cible n'est pas maintenu par le serveur et que le contexte de nommage le plus proche contient un referral, le serveur devrait retourner une réponse referral contruite avec la portion URI de la valeur ref.

exemple : Si le client fournit une opération add où l'objet cible a un DN de "CN=Manager,OU=Roles,O=MNN,C=WW", le serveur va retourner :

```
AddResponse (referral) {  
  ldap ://hostd/CN=Manager,OU=Roles,O=MNN,C=WW"  
}
```

Considérations de l'objet de base

Cette section détail la manipulation des referral pour le traitement de l'objet de base dans les opérations de recherche, il y'a 4 cas.

Dans les cas où l'URI retourné est une URL LDAP, le serveur doit fournir un scope explicite avant de le retourner. En plus, la partie DN doit être modifiée de sorte qu'il réfère à la cible appropriée dans le serveur référencé.

Si le dé-référencement d'alias est nécessaire pour trouver l'objet referral, la partie DN de l'URI doit être remplacée avec le DN de base comme modifié par le dé-référencement d'alias de sorte que l'URL réfère au nouvel objet cible.

cas 1 : L'objet de base n'est pas maintenu par le serveur et n'est pas dans ou subordonné à un contexte de nommage maintenu par le serveur.

Le serveur devrait traiter la requête normalement comme approprié pour une base non-existante qui n'est pas dans un contexte de nommage du serveur (généralement retourne un referral supérieur ou noSuchObject).

cas 2 : L'objet de base est maintenu par le serveur et est un objet referral.

Le serveur devrait retourner la valeur URI contenu dans ref, modifié comme indiqué plus haut.

exemple : Si le client requête un recherche subtree avec l'objet de base "OU=Roles,O=MNN,C=WW", le serveur va retourner :

```
SearchResultDone (referral) {  
  ldap ://hostd/OU=Roles,O=MNN,C=WW??sub  
}
```

cas 3 : L'objet de base n'est pas maintenu par le serveur, mais le contexte de nommage le plus proche ne contient pas de referral dont l'objet de base est subordonné. Le serveur devrait retourner noSuchObject

cas 4 : L'objet de base n'est pas maintenu par le serveur, mais le contexte de nommage le plus proche contient un objet referral dont l'objet de base est subordonné. Le serveur devrait retourner une réponse referral qui est construite depuis la portion URI de ref.

exemple : Si le client requête un search sur la base "CN=Manager,OU=Roles,O=MNN,C=WW", le serveur va retourner :

```
SearchResultDone (referral) {
  ldap://hostd/CN=Manager,OU=Roles,O=MNN,C=WW??base"
}
```

Considérations de continuation de recherche

Pour les opérations de recherche, une fois l'objet de base trouvé et qu'il n'est pas un objet referral, la recherche peut progresser.

exemple : Si un client requête un search subtree de "O=MNN,C=WW", en plus de toute entrée dans le scope dont le filtre match, hosta va aussi retourner 2 références de recherche en tant que 2 objets referral dans le scope. Une réponse possible peut être :

```
SearchEntry for O=MNN,C=WW
SearchResultReference {
  ldap://hostb/OU=People,O=MNN,C=WW??sub
  ldap://hostc/OU=People,O=MNN,C=WW??sub
}
SearchEntry for CN=Manager,O=MNN,C=WW
SearchResultReference {
  ldap://hostd/OU=Roles,O=MNN,C=WW??sub
}
SearchResultDone (success)
```

exemple avec one level : Si le client effectue la recherche de scope OneLevel, une réponse possible peut être :

```
SearchResultReference {
  ldap://hostb/OU=People,O=MNN,C=WW??base
  ldap://hostc/OU=People,O=MNN,C=WW??base
}
SearchEntry for CN=Manager,O=MNN,C=WW
SearchResultReference {
  ldap://hostd/OU=Roles,O=MNN,C=WW??base
}
SearchResultDone (success)
```

BindRequest

Les serveurs ne devraient pas retourner de resultCode referral si un nom bind (ou une authentification identity ou une autorisation identity) est ou est subordonné à un objet referral mais peut utiliser les informations connues pour traiter la requête bind. Si le serveur n'utilise pas ces informations, il doit traiter la requête normalement (retourner un invalidCredentials)

Modify DN

Si le newSuperior est un objet referral ou est subordonné à un objet referral, le serveur devrait retourner un affectsMultipleDSAs. si le newRDN existe déjà mais est un objet referral, le serveur devrait retourner affectsMultipleDSAs au lieu de entryAlreadyExists.